Thesis_ZTA

by Vincencius Galvin Hermanto

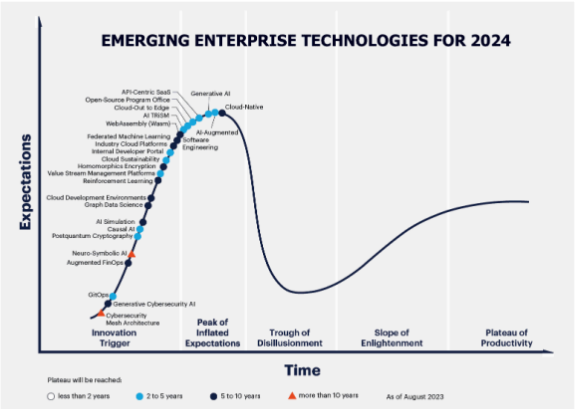
Submission date: 20-Jul-2025 01:14PM (UTC+0700)
Submission ID: 2717549473
File name: TURNITIN-Skripsi.pdf (3.67M)
Word count: 11126
Character count: 72863

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Dengan berkembangnya teknologi digital di era sekarang, semakin banyak perusahaan yang ikut serta melakukan proses digitalisasi terhadap usahanya. Hal ini juga menjadi memicu terbentuknya teknologi baru untuk menompang proses bisnis tersebut, salah satunya dalam bidang *cloud computing* yang ditawarkan oleh berbagai *cloud service provider* seperti *Google Cloud Platform*, *Amazon Web Services*, *Micrsoft Azure*, dan masih banyak lagi. Konsep *cloud computing* berasal dari konsep arsitektur yang terdistribusi (*Distributed software architecture*) yang memiliki tujuan untuk memudahkan pengguna untuk memiliki *resource* yang tepat dan mudah di akses, khususnya dalam proses pengembangan perangkat lunak yang hadir dalam bentuk *cloud resources[1]*. *Cloud resources* merupakan sumber daya TI (*IT Resources*) yang dapat diakses melalui koneksi internet dengan cara berbayar melalui *cloud service provider* setiap kali dipergunakan [2].*Cloud computing* menjadi pilihan yang tepat untuk pelaku usaha dalam hal mengembangkan digitalisasi dengan menawarkan berbagai bentuk *cloud models* seperti IaaS (*Infrastructure as a Service*), PaaS (*Platform as a Service*), CaaS (*Container as a Service*) maupun SaaS (*Software as a Service*)[1]. Hal ini dikarenakan sebagian dari kemampuan *cloud service* yang dapat mempermudah proses pengumpulan, pemrosesan, dan penyimpanan data secara daring melalui platform yang dimiliki oleh *cloud service provider[2]*.

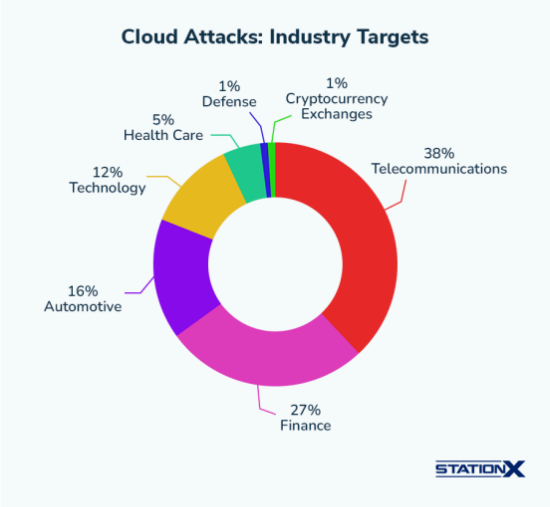


Gambar 1.1 : Prediksi Tren Teknologi yang digunakan Perushaan Tahun 2024

Sumber : Gartner IT *Syposium*, 2023, grafik oleh *US Cloud*

Dengan berkembang dan maraknya penggunaan *cloud computing* di era sekarang, tentu memiliki konsekuensi tersendiri terhadap serangan siber yang semakin marak. Hal tersebut juga disebabkan karena infrastruktur *cloud* mempergunakan protokol internet standar, juga menggunakan konsep *virtualization* [3]. *Virtualization* atau virtualisasi pada komputer merupakan proses penyediaan atau pembuatan perangkat keras, perangkat lunak, sistem operasi, sistem penyimpanan, atau sistem jaringan secara *virtual* dengan mengkedepankan skalabilitas dan kecepatan [4]. Keamanan merupakan bagian yang sangat penting dalam membangun *cloud architecture* yang baik, hal ini meliputi pengunaan *cloud resource* yang benar beserta konfigurasi *resource* tersebut agar sesuai dengan *security compliance* yang ada[1]. Terdapat beberapa usaha untuk meningkatkan postur keamanan yang dimiliki oleh *cloud service provider*, salah satunya adalah "*Autonomus Cloud Intrusion Response System* (ACIRS)" dan sebelumnya "*Network Intrusion Detection and Countermeasure Selection System* (NICE)", yang dimana teknologi *ACIRS* lebih baik dalam memitigasi resiko dalam

pengunaan *virtual network* yang dibuat dalam *cloud environment [1]*. Perkembangan teknologi *machine learning* juga menjadi batu loncatan dalam memperkuat keamanan yang ada, terutama penggunaannya dalam *cloud architecture*. Penggunaan *machine learning* dalam memperkuat keamanan *cloud resources*, berada pada kemampuannya untuk mendeteksi dan memberi peringatan kepada *system administrator* pada saat terjadinya serangan terhadap *cloud environment* yang ada, *machine learning* juga dapat dipergunakan untuk melakukan pengecekan dan penilaian terhadap *cloud infrastructure* yang ada [3].



Gambar 1.2 : Segmentasi Serangan Siber Melalui *Cloud Environment* Sektor Industrial

Sumber : *StationX*, 2024

Pilar keamanan merupakan salah satu hal yang menjadi prioritas dalam membuat arsitektur yang standar. Walaupun sudah ada standarisasi dan fitur- fitur yang menjadi penompang suatu sistem untuk memiliki keamanan yang baik, serangan terhadap sistem tetap dapat terjadi. Salah satu cara untuk meningkatkan

keamanan suatu sistem adalah dengan menerapkan *zero trust concept*. *Zero trust* merupakan paradigma dalam keamanan siber yang terfokus pada perlindungan terhadap *resources* yang ada dalam suatu sistem, konsep ini juga percaya akan akses yang diberikan kepada seseorang terhadap suatu sistem dapat berubah sesuai dengan kebutuhan yang ada dan hanya dibatas untuk kebutuhan tersebut saja [5]. Dengan menggunakan konsep ini, suatu sistem dapat ditingkatkan keamanannya tidak hanya dari ancaman eksternal, melainkan juga ancaman internal. Walaupun ancaman internal banyak disebabkan oleh kesalahan teknis, faktor *human error* merupakan faktor yang menyebabkan banyak ancaman keamanan pada suatu sistem [6]. Faktor tersebut menjadi salah satu penyumbang pelanggaran keamanan terbanyak, hal tersebut membuat manusia menjadi penghubung terlemah (*Weakest Link*) dalam keamanan siber [6]. Dengan menerapkan konsep *zero trust*, faktor-faktor yang menjadi penyebab dari pelanggaran keamanan dapat dieliminiasi satu-persatu. Faktor kesalahan manusia dapat ditanggulangi dengan menerapkan salah satu peraturan dalam konsep *zero trust* yaitu *princple of least privilege*. *Principle of least privilege* merupaan suatu konsep dimana semua akses yang diberikan kepada *user* eksternal maupun *user* internal dibatasi sesuai dengan keperluan masing-masing *user* atau akses granular [5]. Tidak hanya melindungi suatu sistem dari serangan internal saja, konsep ini juga mementingkan semua bagian yang bekerja dalam suatu infrastruktur sistem, konsep *zero trust* juga mementingkan keamanan semua komunikasi yang terjadi di dalam arsitektur sistem, polis atau peraturan terhadap suatu *resource* yang dinamis, dan observasi dan pemantauan kondisi integritas sistem yang rutin [5].

Gambar 1.3 : Potensi Ancaman Terhadap Keamanan *Cloud Data*

Sumber : *StationX*, 2024

Penerapan arsitektur *Zero Trust* dapat menyelesaikan berbagai permasalahan keamanan yang sering terjadi dalam lingkungan *cloud computing*, khususnya yang berkaitan dengan akses tidak sah dan kurangnya kontrol terhadap identitas pengguna. *Zero Trust Architecture* menerapkan prinsip "*never trust, always verify*", di mana setiap permintaan akses harus divalidasi berdasarkan identitas, konteks, dan kepatuhan terhadap kebijakan yang telah ditentukan. Pendekatan ini sangat relevan untuk menjawab tantangan keamanan modern yang kompleks dan dinamis, terutama dalam lingkungan *cloud* yang sangat terdistribusi. Berikut merupakan dua contoh studi kasus yang dapat diatasi melalui penerapan *Zero Trust Architecture*:

- *Unauthorized Access* di Waktu yang Tidak Tepat

  Seorang karyawan yang bertugas melakukan perubahan konfigurasi pada sistem database dijadwalkan untuk melakukannya pada malam hari. Namun karena ia telah memperoleh akses sejak siang hari, ia mencoba melakukan perubahan lebih awal guna mengantisipasi kendala saat eksekusi. Hal ini justru menyebabkan

downtime akibat kesalahan yang tidak disengaja. Dengan pendekatan *Zero Trust*, akses terhadap *resource* dapat dibatasi secara kontekstual, termasuk berdasarkan waktu, sehingga akses hanya aktif sesuai jadwal yang ditentukan. Kebijakan akses berbasis waktu (*time-limited access*) seperti ini dapat meminimalkan risiko gangguan operasional yang disebabkan oleh aktivitas di luar waktu yang diotorisasi.

- *Unaudited Access List* oleh Eks-Karyawan

  Dalam kasus lainnya, seorang mantan karyawan yang telah mengundurkan diri masih memiliki akses aktif terhadap cloud resource milik salah satu klien. Ia memanfaatkan akses tersebut untuk membuat *virtual machine* dan menjalankan aktivitas penambangan *cryptocurrency* secara ilegal. Hal ini menyebabkan konsumsi resource yang tidak sah dan berdampak pada kerugian finansial yang besar bagi perusahaan klien. Penerapan *Zero Trust* dapat mencegah insiden ini melalui automasi audit akses dan pemutusan hak akses berdasarkan perubahan status identitas pengguna. Selain itu, *Zero Trust* juga memungkinkan penerapan prinsip *least privilege* dan *monitoring* akses secara berkelanjutan guna mendeteksi serta memblokir aktivitas mencurigakan secara real-time.

Melihat kompleksitas dan risiko yang ditimbulkan dari kedua kasus tersebut, diperlukan pendekatan keamanan yang lebih adaptif dan presisi dalam memitigasi akses yang tidak sah serta penyalahgunaan hak akses di lingkungan *cloud*. *Zero Trust Architecture* menjadi pendekatan yang relevan karena mampu mengatur kontrol akses secara ketat berdasarkan identitas, waktu, dan konteks penggunaan. Penelitian ini bertujuan untuk menerapkan *Zero Trust Architecture* pada lingkungan *cloud computing* menggunakan *Microsoft Azure* sebagai upaya preventif terhadap potensi insiden keamanan, seperti *unauthorized access* dan *unaudited access list*. Melalui penerapan *Zero Trust*, diharapkan dapat tercipta

arsitektur keamanan cloud yang lebih tangguh, terukur, dan sesuai dengan kebutuhan keamanan sistem informasi modern.

1.2 Rumusan Masalah

Berdasarkan penjelasan yang telah diuraikan pada latar belakang, maka masalah yang akan diteliti dalam penelitian ini adalah: "Bagaimana cara mengimplementasikan konsep arsitektur *zero trust* pada *cloud environment* menggunakan *Microsoft Azure* sebagai *cloud provider* ?"

1.3 Tujuan Penelitian

Berdasarkan ruusan masalah yang ada, maka tujuan dari dilakukannya penelitian ini adalah: "Berhasil melakukan demonstrasi implementasi konsep *zero trust architecture* dalam *cloud environment* menggunakan *Microsoft Azure* sebagai *cloud provider*."

1.4 Manfaat Penelitian

Berikut merupakan manfaat yang ada dari dilakukannya penelitian ini:

1.4.1 Manfaat Bagi Praktisi

Informasi yang diperoleh dari hasil penelitian ini diharapkan dapat menjadi panduan bagi *cloud engineer* yang menggunakan *Microsoft Azure* sebagai *cloud provider* dalam menerapkan prinsip *Zero Trust Architecture* secara praktis. Selain itu, secara konseptual, hasil penelitian ini juga dapat dijadikan referensi oleh *cloud engineer* yang menggunakan penyedia layanan *cloud* lainnya, serta oleh praktisi yang ingin mulai mengimplementasikan pendekatan *Zero Trust* pada *cloud environment* yang sedang digunakan maupun yang akan dibangun. Meskipun pendekatan teknis yang digunakan dalam penelitian ini berfokus pada Azure, prinsip-

prinsip dasar *Zero Trust* yang diterapkan bersifat universal dan dapat diadaptasi sesuai dengan fitur dan layanan dari masing-masing *cloud provider*.

1.4.2 Manfaat Bagi Akademisi

Informasi yang diperoleh dari hasil penelitian ini dapat menjadi pedoman atau kerangka awal (*framework*) dalam mengimplementasikan konsep *Zero Trust Architecture* ke dalam suatu sistem, khususnya pada lingkungan *cloud*. Hasil penelitian ini juga memberikan pemahaman mendalam terkait konsep *Zero Trust Architecture*, *cloud computing*, serta potensi celah keamanan yang dapat ditemukan baik pada lingkungan *cloud* (*cloud environment*) maupun sistem lokal (*on-premises*). Selain itu, informasi yang dihimpun melalui penelitian ini dapat memperkaya literatur yang sudah ada serta menjadi data pendukung untuk penelitian selanjutnya yang mengangkat topik serupa dalam konteks keamanan sistem informasi dan arsitektur *cloud*.

1.5 Batasan Masalah

Batasan masalah yang akan diterapkan dalam penelitian ini adalah sebagai berikut:

1. Untuk mendapatkan tahapan implementasi yang lebih akurat, penulis hanya membatasi implementasi pada *cloud environment* yang terapat atau didukung oleh *Microsoft Azure* sebagai *cloud provider* yang digunakan.
2. Lingkup dari penelitian ini meliputi hal-hal berikut:
    - Mengetahui cara mengimplementasikan konsep *zero trust architecture* dalam *Microsoft Azure cloud environment*,
    - Mengetahui penggunaan *resources* pendukung dalam pembuatan *zero trust architecture*,
    - Melakukan pengaturan atas *resources* yang digunakan dalam menerapkan konsep *zero trust architecture*,

- Menghubungkan antara *user* dan *cloud resources* yang ingin diakses dengan berkaca terhadap konsep *zero trust*.

1.6 Sistematika Penelitian

Skripsi ini disusun dengan struktur sebagai berikut:

Bab 1 Pendahuluan

Dalam bab ini, akan dipaparkan tentang latar belakang penelitian, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah, metodologi penelitian, dan sistematika penelitian.

Bab 2 Landasan Teori

Dalam bab ini akan dibahas literatur yang berisi tentang penjelasan terhadap teori pendukung, batasan konseptual, dan kerangka hipotesis yang bersumber dari penelitian, buku, dan literatur terdahulu yang akan digunakan dalam proses penelitian.

Bab 3 Metode Penelitian

Bab ini mencangkup metode dan teknik penelitian yang digunakan dalam proses penelitian yang akan dilakukan, meliputi rancangan penelitian, teknik implementasi, dan teknik analisis data.

Bab 4 Hasil Penelitian dan Pembahasan

Dalam bab ini akan dijelaskan hasil dan pembahasan dari analisi data terkait penelitian yang telah dilakukan.

Bab 5 Kesimpulan dan Saran

Bab ini mencangkup kesimpulan terhadap penelitan dan analisa yang telah dilakukan, keterbatasan penelitian, serta saran penulis yang berguna meningkatkan keamanan arsitektur sistem.

BAB 2

LANDASAN TEORI

2.1 Kajian Pustaka

Dalam melakukan dan menuliskan penelitian ini, penulis melakukan tinjauan pustaka terhadap penelitian- penelitian serupa maupun yang berhubungan dengan judul yang telah dilakukan oleh peneliti terdahulu. Hal ini dilakukan guna memperoleh informasi terkait kekurangan maupun kelebihan dari penelitian yang telah dilakukan sebelumnya. Penulis juga memperoleh teori yang akan digunakan dalam pembuatan landasan teori ilmiah pada skripsi ini.

Penelitian pertama yang ditinjau oleh penulis merupakan jurnal ilmiah yang ditulis oleh Sina Ahmadi pada tahun 2024. Dalam jurnal ini[7], dibahas tentang implementasi *Zero Trust Architecture* dalam *Cloud Network*, hal ini termasuk pengaplikasian, tantangan dalam implementasi, dan juga peluang pengembangan dalam pengaplikasiannya. Implementasi *zero trust architecture* (ZTA) dalam c*loud environment* akan memperkuat keamanan suatu arsitektur secara signifikan dibanding dengan penggunaan struktur keamanan tradisional, karena struktur yang digunakan dalam *zero trust policy* memerlukan otentikasi lebih guna melindungi informasi yang bersifat sensitif dalam suatu sistem[7] . Jurnal ini juga implementasi ZTA di dalam arsitektur *cloud* seperti pemberian akses terhadap *user* untuk suatu *resource* yang ada di *cloud*, perlindungan akses jaringan yang digunakan dalam arsitektur *cloud*, serta segmentasi yang dapat dilakukan dalam penerapan ZTA. Seperti data yang telah disertakan oleh penulis pada bab sebelumnya, faktor internal menjadi penyumbang pelanggaran keamanan terbanyak dikarenakan manusia merupakan penghubung terlemah (*weakest link)* dalam suatu infrastruktur sistem. Model keamanan *zero trust* dapat mencegah terjadinya pelanggaran keamanan yang disebabkan oleh faktor internal karena ZTA sendiri menganggap tidak ada *user* yang dapat dipercaya, yang memungkinkan untuk melakukan pengecekan dan

pemantauan *user* yang berada di dalam sistem tersebut[7]. Setelah membahas tentang tantangan yang ada, diakhir jurnal ini juga dibahas tentang peluang implementasi ZTA pada arsitektur *cloud* kedepannya. Banyak peluang positif yang dapat digunakan dalam melakukan implementasi ZTA, hal ini dikarenakan seiring berkembangnya teknologi *cloud*, teknologi lain seperti *machine learning* (ML) dan *artificial intelligence* (AI) juga akan ikut berkembang dan dapat memperkuat deteksi ancaman yang ada secara *real-time[7]*. Kedepannya, ZTA akan mengintegrasikan sistem keamanan yang berfokus kepada *user* yang ada[7].

Penelitian kedua yang ditinjau oleh penulis merupakan artikel ilmiah yang ditulis oleh Himanshu Sharma pada tahun 2022. Dalam jurnal ini[8], dibahas tentang implementasi *Zero Trust Architecture* dalam *Cloud*, yang mengacu pada tujuannya untuk memperkuat sistem keamanan yang ada di dalam arsitektur *cloud*. Jurnal ini membantu memahami bagaimana evolusi model keamanan sistem yang ada, hingga akhirnya berkembang menjadi *Zero Trust Architecture* (ZTA). Sistem keamanan yang sekarang sering kali diterapkan merupakan hasil dari berkembangnya sistem keamanan terdahulu. Dalam artikel ini ditunjukkan bahwa perkembangan sistem keamanan komputer dimulai dari *perimeter-based security model*; Konsep keamanan ini mengacu pada penilaian ancaman yang ada pada sistem komputer lama, yang mana semua ancaman diyakini berasal dari faktor eksternal[8]. Paradigma dari ZTA adalah untuk memberikan akses terhadap suatu sistem dan memelihara akses tersebut agar terhindar dari ancaman yang ada[8]. Dalam menerapkan konsep ZTA di dalam suatu arsitektur *cloud*, diperlukan beberapa komponen utama yang akan mendukung sistem keamanan yang ada. Penggunaan *Identity and Access Management* (IAM) dengan mengharuskan setiap *user* memiliki *multi-factor authorization,* segmentasi jaringan dengan mengonfigurasi *virtual private network* yang ada dan menerapkan penggunaan VPN sebagai jembatan penghubung antara sistem *user* dengan sistem utama, dan mengintegrasi sistem dengan komponen *security information and system management* (SIEM) sebagai bentuk pemantauan sistem merupakan komponen utama yang perlu digunakan dalam mengimplementasikan ZTA terhadap suatu arsitektur *cloud*[8].

Kedua tinjauan pustaka yang dilakukan oleh penulis membahas tentang implementasi ZTA dalam suatu arsitektur *cloud*. Adapun perbedaan antara kedua tinjauan pustaka tersebut dengan penelitian yang akan dilakukan adalah keduanya hanya melakukan analisa terhadap implementasi ZTA yang dilakukan dalam arsitektur *cloud*, sedangkan penelitian ini bertujuan untuk melakukan pembuktian atas Implementasi ZTA yang dilakukan. Proses implementasi akan dilakukan dengan menggunakan *Microsoft Azure*, yang merupakan salah satu *cloud provider* yang ada. Dengan memahami kedua tinjauan pustaka di atas, penulis dapat mengetahui fondasi dan tantangan yang akan dihadapi dalam melakukan implementasi.

Tabel 2.1 Penelitian Terdahulu

No.	Nama Penulis	Tahun	Judul Penelitian	Rangkuman
1.	Sina Ahmadi	2024	Zero Trust Architecture in Cloud Networks: Application, Challenges, and Future Opportunities	• Memperkuat keamanan arsitektur cloud secara signifikan dibandingkan dengan struktur keamanan tradisional. • Memerhatikan akses yang diberikan kepada pengguna cloud secara detil dan dinamis. • Faktor internal sebagai penyumbang pelanggaran keamanan terbesar (manusia sebagai titik terlemah). • Peluang pengembangan ZTA dalam cloud ada

				pada perkembangan teknologi *Machine Learning* (ML) dan *Artifiial Intelligence* (AI) yang dapat mendukung deteksi ancaman secara *real-time*.
2.	Himanshu Sharma	2022	Zero Trust in the Cloud: Implementing Zero Trust Architecture for Enhanced Cloud Security	• ZTA muncul sebagai perkembangan lanjutan untuk mengatasi ancaman baik internal maupun eksternal. • Konsep ZTA memberikan dan memelihara akses ke sistem dengan memastikan perlindungan terhadap ancaman yang ada. • Membahas komponen-komponen utama dalam ZTA, yaitu *Identity Access Management* (IAM), segmentasi jaringan, dan integrasi *Security Information & Event Management* (SIEM).

2.2 Zero trust architecture

Zero trust architecture atau disingkat ZTA, merupakan salah satu kerangka (*framework*) berpikir yang menjadi standar dalam membuat suatu sistem yang aman. Kerangka ZTA sudah ada secara konseptual jauh sebelum maraknya penggunaan ZTA di kalangan sistem keamanan, konsep atau strategi ini disebut dengan "*Black Core*" (BCORE)[5]. Inti dari strategi ini adalah sistem keamanan yang ada pada suatu sistem akan terfokus kepada masing-masing transaksi (*transaction*) atau aksi (*action/event*) yang terjadi di dalam sistem tersebut. Secara definisi resmi menurut *National Institute of Standards and Technology* (NIST), ZTA adalah paradigma keamanan siber yang memiliki fokus terhadap proteksi setiap komponen dalam suatu sistem secara individu dan memiliki premis bahwa akses yang diberikan untuk masuk ke dalam sistem harus terus dievaluasi secara berkelanjutan[5].

Kerangka arsitektur ini terfokus terhadap setiap komponen yang ada di dalam sistem, meliputi semua komponen, koneksi antar komponen, dan semua *user* yang sedang ataupun dapat mengakses sistem tersebut secara lokal maupun *remote.* Konsep keamanan *zero trust* sendiri memberikan kumpulan konsep dan ide yang didesain untuk meminimalisir permukaan serangan yang dimiliki oleh suatu sistem, kerangka keamanan ZTA menggunakan konsep dan ide tersebut untuk membangun arsitektur dengan sistem keamanan yang mengedepankan hubungan antar komponen, perencanaan alur sistem, dan kebijakan akses dari sistem yang ada[5] .

2.2.1 Prinsip Zero trust architecture

Suatu infrastruktur yang akan menggunakan kerangka ZTA harus berpatokan pada prinsip-prinsip dasar berikut[5] :

- **Seluruh sumber data (*data sources*) dan layanan komputasi (*computing services*) akan dianggap sebagai komponen sistem**; Dalam suatu infrastruktur pastinya akan ada banyak segmentasi jaringan yang memiliki

komponen dan layanan masing-masing dengan skala yang beragam, dalam ZTA seluruh komponen dan sumber data yang keluar ataupun masuk ke dalam sistem akan dianggap sebagai komponen yang harus dilindungi.

- **Seluruh komunikasi yang terjadi antar komponen harus secara aman dimanapun lokasi komponen yang berkomunikasi *(secure line communication)***; Dalam suatu infrastruktur *cloud*, komponen dapat tersebar tergantung dari lokasi dibuatnya komponen tersebut, dalam kerangka keamanan ZTA lokasi tersebut tidak akan dipertimbangkan tingkat keamanannya, maka yang akan diamankan adalah jalur komunikasi yang digunakan antara komponen-komponen tersebut.
- **Akses sistem yang ada akan diberikan dengan batasan waktu *(time-limited access)***; Kerangka ZTA sangat ketat terhadap akses yang diberikan, terutama terhadap komponen yang akan mengakses titik-titik sensitif suatu sistem (panel admin, basis data, panel konfigurasi, dll.) yang dimana akan diberikan akses dengan adanya batasan waktu sesuai dengan penggunaan komponen yang ada dalam sistem.
- **Akses yang diberikan akan ditentukan oleh kebijakan yang dinamis *(dynamic access policy)***; Akses yang diberikan akan bergantung keras terhadap kebijakan yang ada, dimana kebijakan tersebut harus tetap dinamis menyesuaikan dengan praktik terbaik dalam keamanan siber.
- **Komponen yang ada di dalam sistem harus selalu dipantau dan dievaluasi tingkat keamanannya secara berkala *(continuous system monitoring dan evaluation)***; Komponen yang ada harus selalu dipantau baik dari segi kinerja maupun statusnya, tingkat keamanan yang ada juga harus dievaluasi secara berkala karena dalam kerangka ZTA tidak ada komponen yang dapat terus menerus dipercaya tingkat keamanannya.
- **Seluruh akses yang diberikan harus selalu dievaluasi secara dinamis sebelum diotorisasi *(evaluated access)***; Akses yang diberikan terhadap suatu komponen atau *user* yang akan terhubung harus dievaluasi dengan jelas dan teliti sebelum melakukan otorisasi terhadap akses tersebut.

- **Informasi yang ada dalam komponen, jaringan, maupun akses yang diberikan harus selalu digunakan untuk analisa peningkatan postur keamanan *(system log analysis)*;** Informasi yang didapat dari setiap komponen, komunikasi antar jaringan, dan akses yang diberikan dapat berupa *logs* harus dikumpulkan dan digunakan untuk menganalisis potensi peningkatan maupun ancaman yang dapat digunakan untuk meningkatkan postur keamanan yang telah diterapkan, hal ini dilakukan agar postur keamanan tetap dinamis dan selalu diperbarui dengan praktik terbaik.

2.3 Cloud computing

Cloud computing merupakan salah satu tahap perkembangan teknologi informasi yang mengedepankan prinsip aksesibilitas *on-demand* pada layanan & produk komputasi. "*Cloud"-computing* merupakan teknologi infrastruktur siber yang menggabungkan teknologi terdahulu seperti *virtualization*, *distributed computing*, *networking,* dan *software services* ke dalam satu platform dengan aksesibilitas dan skalabilitas yang tinggi[9]. Dengan kelebihan tersebut, *cloud computing* menawarkan platform *on-demand* yang memiliki layanan dan produk komputasi yang beragam yang dapat digunakan sesuai dengan kebutuhan pengguna. Konsep yang mendukung *cloud computing* menjadi salah satu teknologi yang marak digunakan adalah komputasi melalui *service-oriented architecture* (SOA) yang merupakan layanan komputasi yang sudah diatur dan terintegrasi untuk pengguna yang dapat langsung dikonfigurasi dan digunakan[9]. Dengan SOA, pengguna dapat mengonfigurasi layanan komputasi dengan fungsi, spesifikasi, dan skala yang diinginkan yang dapat langsung digunakan[9].

Konsep *virtualization* adalah pilar yang cukup banyak menarik perhatian pengguna dalam menggunakan teknologi *cloud computing*. *Virtualization* adalah teknologi yang mengabstraksi dan mengisolasi fungsionalitas dasar dan perangkat keras dari suatu sistem, hal ini memungkinkan portabilitas dari fungsi-fungsi yang lebih tinggi dengan membagi/ mengagregasi perangkat lunak yang digunakan[9]. Secara lebih sederhananya, *virtualization* adalah pembuatan perangkat keras,

perangkat lunak, platform, perangkat penyimpanan, sistem operasi, maupun perangkat jaringan secara virtual (tidak asli) yang secara skala, funsional, maupun spesifikasinya dapat dirubah dengan cepat[4].

2.3.1 Model layanan dalam *Cloud computing*

Dalam *cloud computing*, layanan seperti perangkat lunak, infrastruktur perangkat keras, infrastruktur jaringan, hingga perangkat penyimpanan akan disediakan untuk pengguna, *cloud computing* memiliki 3 model layanan:

1. ***Private Cloud***

   Private cloud merupakan model layanan *cloud computing* yang lingkupnya hanya digunakan oleh organisasi tertentu yang dimana semua layanan yang digunakan hanya dapat diakses oleh sejumlah orang[4].

2. ***Public Cloud***

   Public cloud merupakan model layanan *cloud computing* yang dapat diakses secara publik dengan koneksi internet, akses publik ini diperuntukkan kepada pengguna yang ingin menggunakan layanan *cloud computing* secara personal dengan basis pembayaran *pay-per use[4]*.

3. ***Hybrid Cloud***

   Hybrid Cloud merupakan gabungan antara kedua model sebelumnya dan menawarkan kelebihan dari masing-masing model. Layanan *hybrid cloud* memungkinkan organisasi untuk menggunakan layanan *cloud computing* secara privat dalam lingkup internalnya dan juga publik dalam lingkup eksternalnya (*public-facing services*)[10].

2.4 Microsoft Azure

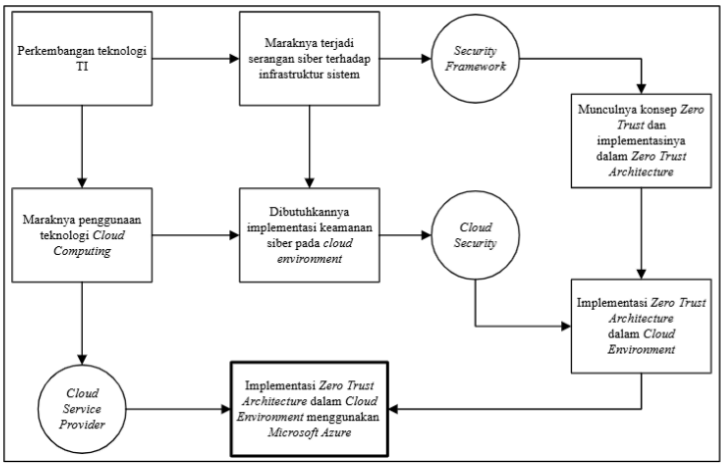
Dalam menggunakan layanan *cloud computing*, pengguna harus mengakses layanan yang ditawarkan melalui penyedia layanan (*cloud service providers*) yang beragam. Penyedia layanan *cloud service* merupakan model bisnis teknologi informasi yang menyediakan layanan *cloud computing* yang dapat diakses melalui internet[2]. Layanan yang ditawarkan biasanya memiliki berbagai bentuk, mulai dari *Infrastructure as a Service* (IAAS), *Platform as a Service* (PAAS), *Software as a Service* (SAAS), hingga yang paling baru *Infrastructure as a Code* (IAAC).

Sebagai penyedia layanan *cloud computing*, perusahaan yang bergerak dibidang ini menyediakan layanan dengan skalabilitas tinggi, yang dapat diakses secara *on-demand* melalui jaringan internet, meliputi *cloud-based computing*, penyimpanan data, platform, hingga aplikasi-aplikasi pendukung bagi pelaku bisnis, organisasi, hingga penggunaan personal[2]. Salah satu penyedia layanan ini adalah *Azure* yang dikelola oleh *Microsoft*. *Microsoft Azure* merupakan produk dari *Microsoft* yang masih terus berkembang dan memiliki cakupan layanan yang luas untuk keperluan penggunaan layanan *cloud computing[11]* . *Microsoft Azure* sendiri menawarkan layanan *cloud* dengan tingkat *availability* yang tinggi, integrasi antara *private* dan *public cloud*, integrasi sistem keamanan *cloud* dengan *Microsoft Defender for Cloud*, hingga integrasi dengan produk & perangkat lunak dari *Microsoft[12]*. Penggunaan *Microsoft Azure* dalam penelitian ini akan menujukan implementasi ZTA yang dapat dilakukan dengan memanfaatkan layanan keamanan yang ditawarkan oleh *Micrososft Azure*, implementasi juga akan dilakukan dengan menggunakan model *hybrid cloud* yang dapat digunakan dengan menggunakan *Microsoft Azure*. Selain itu, pemilihan *Microsoft Azure* sebagai media penelitian disebabkan karena *Microsoft Azure* merupakan salah satu dari 3 penyedia layanan *cloud* yang terdiri atas *Microsoft Azure, Amazon Web Services* (AWS), dan *Google Cloud Platform* (GCP) [2].

BAB 3

METODOLOGI PENELITIAN

3.1 Kerangka Pemikiran



Gambar 3.1: Diagram Kerangka Pemikiran

Kerangka pemikiran yang digunakan oleh penulis dalam melakukan penelitian ini didasari oleh perkembangan teknologi informasi yang sudah menjadi standar industri di masa sekarang. Teknologi *cloud computing* adalah perkembangan yang menjadi fokus dalam penelitian ini. Seiring berkembangnya teknologi yang ada, serangan siber terhadap infrastruktur sistem juga semakin marak terjadi. Hal ini memicu bertumbuhnya permintaan atas implementasi keamanan siber dalam *cloud environment* yang merupakan dasar munculya sektor *cloud security*.Atas permintaan tersebut, terciptalah sektor *Cloud Security*. Maraknya serangan siber juga memicu dibuatnya standarisasi kerangka keamanan yang digunakan (*Security Framework*), salah satunya adalah penggunaan konsep *Zero Trust Architecture*. Implementasi *Zero Trust Architecture* ini akan di

implementasikan kedalam *cloud environment* dengan menggunakan *Microsoft Azure*, yang merupakan salah satu *Cloud Service Provider* yang marak digunakan. Penelitian ini ditujukan untuk memperjelas dan menyediakan kerangka implementasi *Zero Trust Architecture* menggunakan *Microsoft Azure* sebagai *Cloud Service Provider*.

3.2 Metode Penelitian

Dalam melakukan implementasi *Zero Trust Architecture* (ZTA) pada *cloud environment* dengan menggunakan *Microsoft Azure*, akan dibuat dua infrastruktur *cloud* yang berbeda. Hal ini dilakukan guna memberikan perspektif perbandingan pada implementasi yang dilakukan. Salah satu infrastruktur ini merupakan infrastruktur biasa tanpa implemetasi ZTA (*Project-Default*), sedangkan yang lainnya merupakan infrastruktur yang telah diimplementasi konsep ZTA (*Project-ZTA*).

Setelah kedua infrastruktur tersebut dibuat, maka akan dilakukan *testing* pada komponen yang ada guna mencerminkan funginya dalam memenuhi 7 prinsip *Zero Trust* yang telah di jelaskan pada bagian sebelumnya. Akan dilakukan juga perbandingan antara implementasi *Zero Trust Architecture* yang dibuat oleh *Microsoft* sendiri, dengan implementasi yang dilakukan pada penelitian ini. Perbandingan akan dilakukan melalui dua hal, yaitu komponen yang digunakan dan total harga dari masing- masing implementasi. Menurut panduan yang dipublikasi oleh *Microsoft* sendiri berikut merupakan komponen yang diperlukan dalam implementasi *Zero Trust Architecture* pada *Azure*[13]:

1. **Azure *Key Vault***

   Layanan untuk menyimpan dan mengelola *secrets*, *keys*, dan *certificates* secara aman. Membantu melindungi informasi sensitif seperti kredensial dan token dari akses yang tidak sah.

2. **Azure Bastion**

Layanan untuk mengakses mesin virtual (VM) secara aman melalui *browser* menggunakan RDP atau SSH tanpa perlu membuka IP publik. Ini mengurangi risiko serangan langsung ke VM dari internet.

3. ***Just-in-time Access***

   Fitur yang memberikan akses sementara dan terbatas waktu ke *resource* tertentu, seperti VM. Mengurangi permukaan serangan dengan hanya membuka akses saat dibutuhkan.

4. **Azure *Firewall***

   Layanan firewall jaringan berbasis cloud yang menyediakan kontrol lalu lintas masuk dan keluar berdasarkan aturan keamanan. Mendukung fitur seperti *filtering layer* 3–7 dan *logging* aktivitas jaringan.

5. **Azure *DdoS Protection***

   Layanan untuk melindungi aplikasi dan layanan dari serangan *Distributed Denial* of *Service* (DDoS). Mendeteksi dan merespons serangan secara otomatis untuk menjaga ketersediaan sistem.

6. **Azure AD**

   Layanan identitas berbasis cloud untuk manajemen autentikasi dan otorisasi pengguna. Mendukung fitur seperti *Single Sign-On* (SSO) dan *Multi-Factor Authentication* (MFA).

7. **Azure *Purview***

   Layanan tata kelola data (*data governance*) yang memungkinkan pemetaan, pelacakan, dan pengelolaan aset data di

seluruh lingkungan Azure dan non-Azure. Membantu organisasi memahami dan mengamankan data sensitif.

8. ***Application Gateway***

   Load balancer layer 7 yang mengelola lalu lintas HTTP/HTTPS dengan fitur seperti URL-based routing dan *Web Application Firewall* (WAF). Melindungi aplikasi *web* dari serangan umum seperti SQL *injection* dan XSS.

9. ***Virtual Network Gateway***

   Layanan yang digunakan untuk membangun koneksi VPN antara Azure dan lingkungan *on-premises*. Mendukung koneksi *site-to-site*, *point-to-site*, dan ExpressRoute.

10. **Azure *Monitor***

    Platform untuk mengumpulkan, menganalisis, dan merespons data telemetri dari aplikasi dan *resource* Azure. Membantu mengidentifikasi masalah performa dan memantau status sistem secara *real-time*.

11. **Azure *Advisor***

    Layanan rekomendasi berbasis AI yang memberikan saran untuk meningkatkan kinerja, keamanan, dan efisiensi biaya dari *resource* Azure. Menyediakan panduan berbasis praktik terbaik Microsoft.

Pengujian dilakukan pada masing-masing infrastruktur untuk mencerminkan sejauh mana penerapan prinsip *Zero Trust Architecture* dapat diterapkan dan memberikan dampak terhadap aspek keamanan. Setiap pengujian disesuaikan dengan tujuh prinsip dasar *Zero Trust Architecture* dari NIST yang telah dijelaskan sebelumnya, dengan skenario sebagai berikut:

1. ***Resources Includes All Data and Services***

   Pada prinsip ini, pengujian difokuskan pada penerapan *granular access control* terhadap seluruh sumber daya (*resources*) yang berada dalam satu *subscription*. Tidak ada akses yang diturunkan secara otomatis antar *resource* yang berbeda, sehingga setiap *resource* dilindungi secara individual dan hanya dapat diakses oleh entitas yang memiliki izin eksplisit.

2. ***Secure-line Communication***

   Seluruh komunikasi *inbound* maupun *outbound* dilakukan melalui *gateway* terproteksi seperti *Application Gateway* dan *VPN Gateway*. Tidak ada resource yang memiliki IP publik langsung, sehingga jalur komunikasi terenkripsi dan melewati lapisan validasi, guna mencegah akses langsung dari internet terbuka.

3. ***Time-limited Access***

   Akses ke salah satu resource diberikan kepada pengguna dengan batasan waktu tertentu melalui *time-based access policy*. Pengaturan ini memastikan bahwa akses yang diberikan bersifat sementara dan akan dicabut secara otomatis setelah melewati jangka waktu yang ditentukan.

4. ***Dynamic Access Policy***

   Untuk mencerminkan kebijakan akses yang dinamis, setiap pengguna diwajibkan untuk melakukan autentikasi dua faktor (*multi-factor authentication*/MFA) sebelum memperoleh akses ke *resource*. Hal ini memberikan lapisan keamanan tambahan dan meminimalkan risiko dari kredensial yang disalahgunakan.

5. ***Continuous System Monitoring***

Pengujian dilakukan dengan mengaktifkan fitur *resource health monitoring* pada setiap komponen infrastruktur. Dengan ini, status kesehatan sistem dapat dipantau secara berkala, memungkinkan deteksi dini terhadap potensi gangguan atau penyimpangan yang terjadi pada *resource*.

6. ***Evaluated Access***

   Evaluasi akses dilakukan sebelum pemberian hak akses baru. Setiap permintaan akses dievaluasi berdasarkan prinsip *least privilege*, sehingga hanya akses yang benar-benar dibutuhkan dan sesuai dengan tugas pengguna yang diberikan izin.

7. ***System Log Analysis***

   Aktivitas sistem dicatat secara menyeluruh melalui *activity logs* dan *diagnostic logs*. Selain itu, aturan peringatan (*alert rules*) diaktifkan untuk memberikan notifikasi otomatis kepada administrator saat terjadi aktivitas abnormal atau percobaan akses yang mencurigakan.

Sebelum melakukan implementasi ZTA, perlu dilakukan perancangan arsitektur sistem yang akan dibuat. Hal ini merupakan tahap yang penting karena dalam tahap perancangan arsitektur ini, harus juga dilakukan perhitungan *class inter-domain routing* (CIDR) terhadap *virtual network* yang akan dibuat. Perhitungan CIDR *range* ini mencangkup seluruh komponen dalam sistem yang memerlukan *range* IP-nya masing- masing. Komponen komponen yang berjalan dalam suatu *cloud environment* memerlukan bagian *range* dalam pembuatan dan penggunaannya, dengan masing- masing komponen membutuhkan *minimum range* yang bervariasi.

3.2.1 Perancangan Arsitektur *Cloud*

Dalam melakukan perancangan arsitektur *cloud* yang ada, kita perlu mengetahui komponen apa saja yang akan digunakan dalam infrastruktur ini. Dalam penelitian ini, penulis akan membuat infrastruktur sederhana yang dapat digunakan untuk melakukan *web hosting*. Setelah mengetahui tujuan penggunaan suatu infrastruktur, kita dapat mengetahui komponen yang perlu dikonfigurasi. Komponen- komponen utama yang akan digunakan adalah sebagai berikut:

1. ***Resource Groups***

   Resource Group merupakan kumpulan *service* dan *resource* yang digunakan dalam suatu infrastruktur *cloud* pada *Microsoft Azure*. Komponen ini merupakan komponen yang wajib dimiliki setiap infrastruktur yang ada karena akan menjadi basis pengelompokan komponen yang ada.

2. ***Azure Virtual Network***

   Azure Virtual Network merupakan komponen *private virtual network* yang memungkinkan komunikasi aman antar komponen yang ada didalamnya. Penggunaan komponen ini memastikan pemenuhan salah satu syarat dalam ZTA yang memerlukan *secure communication* dalam suatu infrastruktur.

3. ***Azure Virtual Machines***

   Azure Virtual Machine (VM) merupakan layanan *virtualization* yang dimiliki *Azure* dengan menempatkan *virtual machine* yang dibuat pada *cloud*. Penggunaan komponen VM ini akan menjadi tempat di *hosting*-nya *web page* dalam infrastruktur ini. Komponen ini juga memiliki IP *public* yang dapat digunakan untuk akses *via* internet.

4. ***Azure Disks***

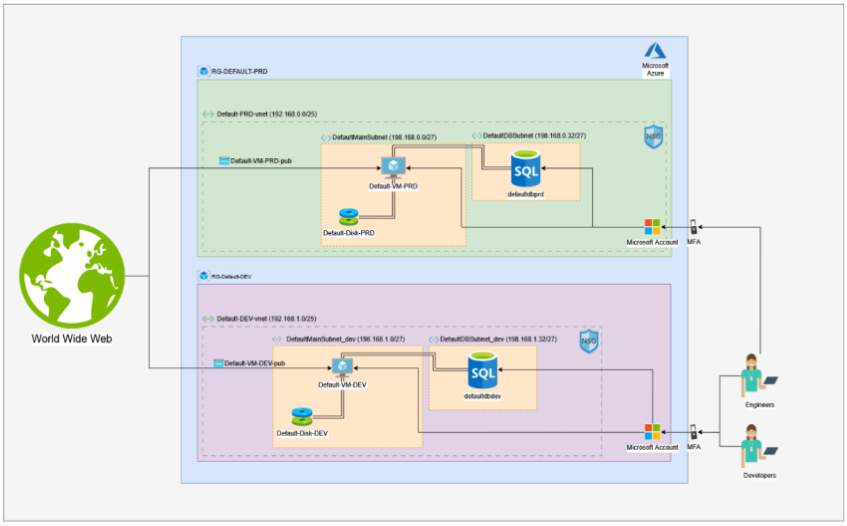
Azure Disks merupakan layanan manajemen penyimpanan berbetuk *disk* yang akan terhubung dengan VM yang telah dibuat. Komponen ini sendiri akan menyimpan seluruh data yang ada di dalam VM yang telah dibuat, hal ini termasuk OS hingga data- data individual yang ada di dalam VM tersebut.

5. ***Azure Databases***

   Azure Databases merupakan layanan basis data yang ditawarkan oleh *Azure*. Komponen ini akan digunakan sebagai *server* penyimpanan bagi *web page* yang di-*deploy* dalam *virtual machine* yang ada.

6. ***Microsoft Account***

   Microsoft Account merupakan salah satu komponen yang digunakan dalam infrastruktur ini sebagai gerbang akses standar yang digunakan dalam mengakses *Microsoft Azure*. Komponen ini dilengkapi juga dengan fitur *multi-factor authentication* (MFA) sebagai standar keamanan dalam akun *Microsoft* yang ada.



Gambar 3.2: *Project-Default* (Tanpa Implementasi ZTA)

Komponen- komponen yang ada diatas merupakan komponen utama, yang dimana akan digunakan di kedua infrastruktur yang ada. Dalam melakukan implementasi ZTA pada infrastruktur tersebut, kita perlu melakukan konfigurasi dan penambahan beberapa komponen lainnya. Komponen ini ditambahkan guna memenuhi prinsip yang ada dalam konsep *Zero Trust Architecture*, yaitu:

1. Semua komponen dalam sistem harus terlindungi (*secure component*)
2. Komunikasi antar komponen harus terjadi secara aman (*secure line communication*)
3. Akses yang diberikan harus disertakan dengan batasan waktu (*time limited access*)
4. Kebijakan sistem yang dinamis (*dynamic policy*)
5. Melakukan evaluasi terhadap akses dan kebijakan yang ada (*access and policy evaluation*)

6. Menggunakan informasi sistem yang ada guna meningkatkan postur keamanan (*system log analysis for security posture*)

Dengan adanya prinsip- prinsip tersebut, maka diperlukan komponen dan konfigurasi tambahan yang akan digunakan dalam infrastruktur implementasi ZTA, yang mencakup:

1. ***Resource Health Alerts* pada *Azure Vrtual Machines***

   Virtual Machine dapat dilengkapi dengan fitur *resource health* yang dapat memberikan informasi terkait kondisi sistem yang ada selama VM tersebut berjalan. Fitur ini juga dapat menjadi garda depan apabila terjadi ketidaksesuaian dengan status penggunaan sistem yang ada, dan dapat memberikan peringatan kepada *engineer* sebelum terjadi isu yang lebih besar. Status yang dapat di-*monitor* oleh fitur ini mencakup penggunaan CPU, RAM, dan *Storage* pada VM; Juga tingkat *latency* yang ada antara VM dan komponen- komponen yang terkoneksi.

2. ***Azure Monitoring***

   Azure Monitor merupakan komponen monitoring yang dimiliki oleh *Azure*. Komponen ini akan memenuhi persyaratan *monitoring* dan juga *system logging* guna meningkatkan postur keamanan dan investigasi *post-incident*.

3. ***Azure Application Gateway***

   Azure Application Gateway merupakan komponen *load balancer* pada *application layer* (Layer 7) yang berfungsi untuk mengatur *ingress* dan *engress* kedalam suatu aplikasi web, yang dimana dalam infrastruktur ini merupakan *web page* yang di *host* di VM. Komponen ini juga memiliki fitur keamanan tambahan untuk memenuhi prinsip ZTA seperti, *web application firewall* (WAF), *url-based routing*, dan *SSL Termination*. Fitur keamanan tersebut akan membantu dalam mengamankan layanan *web* yang terkespos ke *public* (*public facing*).

4. ***Azure Web Application Firewall***

   Azure Web Application Firewall merupakan fitur keamanan dari *Azure Application Gateway* yang melindungi aplikasi web dari serangan umum seperti SQL injection, cross-site scripting, dan OWASP *Top* 10 *vulnerabilities*.

5. ***Azure Network Watchers***

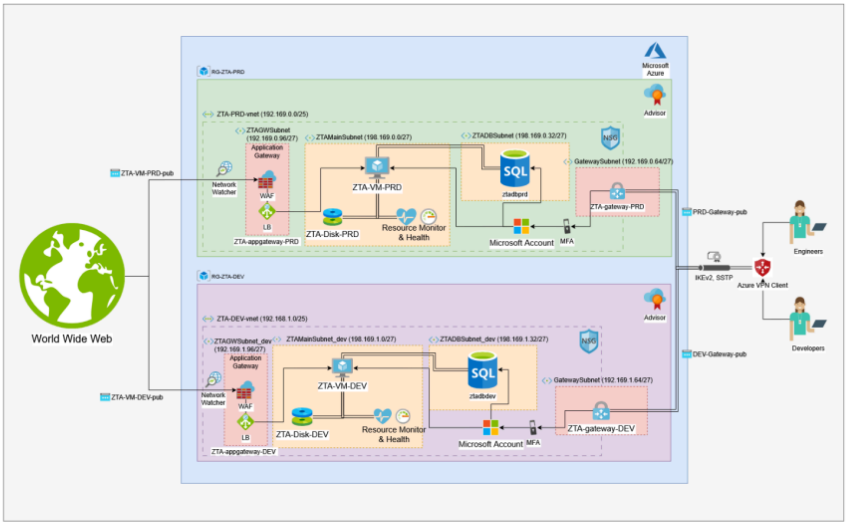
   Azure NetWatch merupakan komponen yang berfungsi sebagai alat pemantauan dan diagnostik jaringan yang dapat membantu dalam analisa dan visualisasi konektivitas pada suatu innfrastruktur. Komponen ini dapat memenuhi prinsip *secure line communication* dan *system log monitoring*.

6. ***Azure Advisor* pada *Microsoft Defender for Cloud***

   Azure Advisor merupakan komponen yang dapat menganalisa postur keamanan suatu infrastruktur *Azure* dengan berpondasikan 5 pillar, yaitu: *Cost, Reliability, Security, Performance,* dan *Operational Excellence*.

7. ***Azure Virtual Network Gateway***

   Azure Virtual Network Gateway merupakan komponen yang berfungsi untuk membuat koneksi VPN. Jenis koneksi VPN yang akan digunakan pada infrastruktur ini adalah koneksin VPN *point to site* yang menghubungkan antara perangkat personal dengan jaringan *Azure virtual network*.

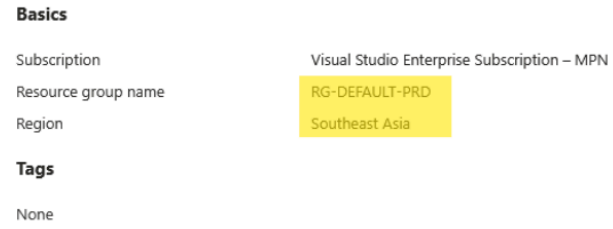


Gambar 3.3: *Project-ZTA* (Setelah Implementasi ZTA)

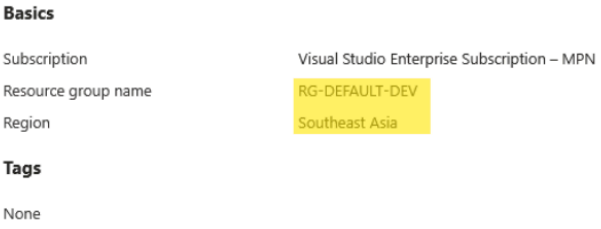
3.2.2 Pembuatan Infrastruktur *Project-Default*

Pada bagian ini, akan dijelaskan proses pembuatan infrastruktur *cloud* menggunakan *Microsoft Azure*. Bagian ini akan menjelaskan infrastruktur pertama, yaitu "Infrastruktur *Project-Default*" yang akan menjelaskan prosedur pembuatan infrastruktur *Project-Default* tanpa implementasi konsep ZTA. Seluruh prosedur akan dilakukan melalui *Azure Portal* yang diakses di *web browser*. Diagram arsitektur yang akan digunakan adalah diagram *Project-Default* pada Gambar 3.2.

Komponen pertama yang harus dibuat adalah 2 buah *resource group*, konfigurasi yang akan dilakukan hanya terdapat pada nama *resource group* (RG-DEFAULT-PRD & RG-DEFAULT-DEV) dan *region* dibuatnya komponen. Untuk seluruh komponen pada penelitian ini akan menggunakan Asia Tenggara (*Southeast Asia*) sebagai *region* yang dipilih.



Gambar 3.4 Konfigurasi RG-DEFAULT-PRD



Gambar 3.5 Konfigurasi RG-DEFAULT-DEV

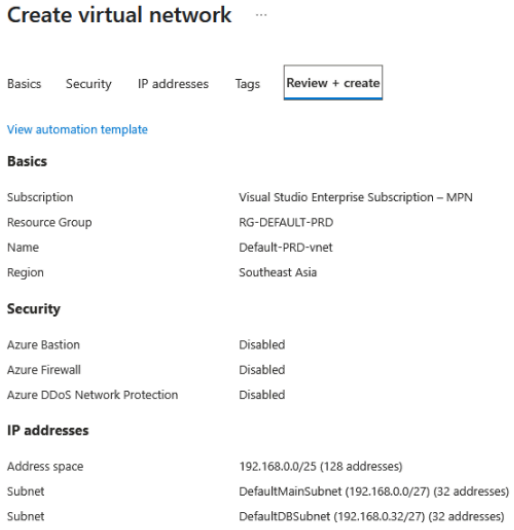
Setelah *resource group* yang diperlukan selesai dibuat, kita bisa melakukan pembuatan komponen- komponen lain dalam lingkup masing masing dari kedua *resource group* tersebut. Langkah selanjutnya adalah mengkonfigurasi *virtual network* berserta dengan *subnet* yang ada di dalamnya.

Infrastruktur ini memiliki 2 *virtual network* (Default-PRD-vnet & Default-DEV-vnet) dengan 2 *subnet* di masing- masing *virtual network* (*Main subnet* & *DB subnet*). Konfigurasi yang akan digunakan dalam membuat *virtual network* adalah sebagai berikut:

RG-DEFAULT-PRD

- Default-PRD-vnet:
  - *Resource Group*: RG-DEFAULT-PRD
  - *Name*: Default-PRD-vnet

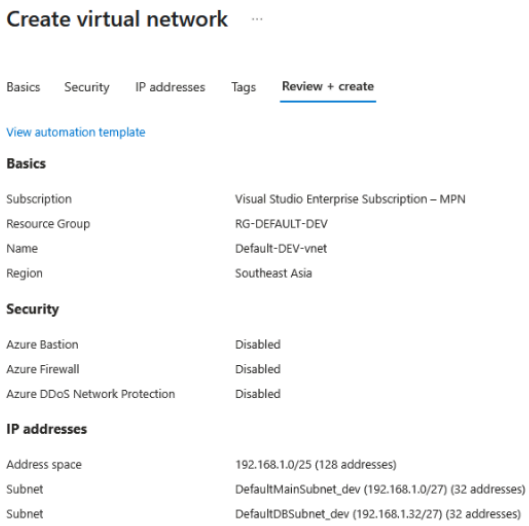
- *IP addresses*: 192.168.0.0/25
- *Subnet*:
  - DefaultMainSubnet (192.168.0.0/27)
  - DefaultDBSubnet (192.168.0.32/27)



Gambar 3.6 Konfigurasi Default-PRD-vnet

RG-DEFAULT-DEV

- Default-DEV-net:
  - *Resource Group*: RG-DEFAULT-DEV
  - *Name*: Default-DEV-vnet
  - *IP addresses*: 192.168.1.0/25
  - *Subnet*:
    - DefaultMainSubnet (192.168.1.0/27)
    - DefaultDBSubnet (192.168.1.32/27)



Gambar 3.7 Konfigurasi Default-DEV-vnet

Setelah membuat *virtual network,* akan dibuat *virtual machine* menggunakan vnet yang ada. Infrastruktur ini memiliki 2 buah VM (Default-VM-PRD & Default-VM-DEV) dan masing- masing 1 *disk* yang ada didalamnya (Default-Disk-PRD & Default-Disk-DEV). Dalam penelitian ini kita akan menggunakan konfigurasi sebagai berikut:

- ***DEFAULT-VM-PRD***
  - **Basic**
    - *Resource group*: RG-DEFAULT-PRD
    - *Name*: Default-VM-PRD
    - *Availability options*: *No infrastructure redundancy required*
    - *Security type*: *Standard*
    - *Machine Type*: *Standard*_B1 ls (1 vCPU, 0.5 GiB *memory*)
    - *Image*: *Ubuntu Server* 24.04 LTS- x64 *Gen*2

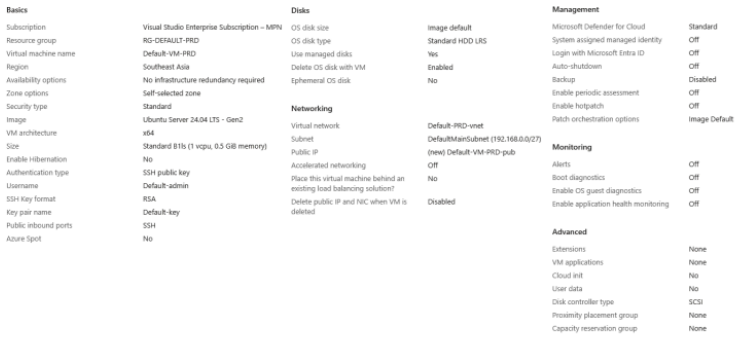
- *Authenticaiton Type*: SSH *public key*
- *Username*: *Default-admin*
- *Key pair name*: *Default-key*
- *Inbound ports*: SSH (22)

- **Disk**
  - OS *Disk size*: *Image default* (30 GiB)
  - *OS Disk type*: *Standard* HDD (*locally-redundant storage*)

- **Networking**
  - *Virtual Network*: Default-PRD-vnet
  - *Subnet*: DefaultMainSubnet (192.168.0.0/27)
  - *Public IP*: Default-VM-PRD-pub

- **Management**
  - *Auto-shutdown*: Disabled

- **Monitoring**
  - *Boot Diagnostics*: Disabled



Basics	
Subscription	Visual Studio Enterprise Subscription – MPN
Resource group	RG-DEFAULT-PRD
Virtual machine name	Default-VM-PRD
Region	Southeast Asia
Availability options	No infrastructure redundancy required
Zone options	Self-selected zone
Security type	Standard
Image	Ubuntu Server 24.04 LTS - Gen2
VM architecture	x64
Size	Standard B1ls (1 vcpu, 0.5 GiB memory)
Enable Hibernation	No
Authentication type	SSH public key
Username	Default-admin
SSH Key format	RSA
Key pair name	Default-key
Public inbound ports	SSH
Azure Spot	No

Disks	
OS disk size	Image default
OS disk type	Standard HDD LRS
Use managed disks	Yes
Delete OS disk with VM	Enabled
Ephemeral OS disk	No
Networking	
Virtual network	Default-PRD-vnet
Subnet	DefaultMainSubnet (192.168.0.0/27)
Public IP	(new) Default-VM-PRD-pub
Accelerated networking	Off
Place this virtual machine behind an existing load balancing solution?	No
Delete public IP and NIC when VM is deleted	Disabled

Management	
Microsoft Defender for Cloud	Standard
System assigned managed identity	Off
Login with Microsoft Entra ID	Off
Auto-shutdown	Off
Backup	Disabled
Enable periodic assessment	Off
Enable hotpatch	Off
Patch orchestration options	Image Default
Monitoring	
Alerts	Off
Boot diagnostics	Off
Enable OS guest diagnostics	Off
Enable application health monitoring	Off
Advanced	
Extensions	None
VM applications	None
Cloud init	No
User data	No
Disk controller type	SCSI
Proximity placement group	None
Capacity reservation group	None

Gambar 3.8 Konfigurasi Default-VM-PRD

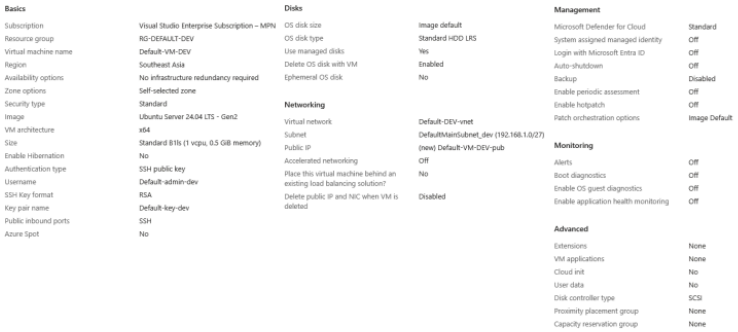
- ***DEFAULT-VM-DEV***
  - **Basic**
    - *Resource group*: RG-DEFAULT-DEV
    - *Name*: Default-VM-DEV
    - *Availability options*: *No infrastructure redundancy required*
    - *Security type*: *Standard*
    - *Machine Type*: *Standard*_B1 ls (1 vCPU, 0.5 GiB *memory*)
    - *Image*: *Ubuntu Server* 24.04 LTS- x64 *Gen*2
    - *Authenticaiton Type*: SSH *public key*
    - *Username*: *Default-admin-dev*
    - *Key pair name*: *Default-key-dev*
    - *Inbound ports*: SSH (22)

  - **Disk**
    - OS *Disk size*: *Image default* (30 GiB)
    - *OS Disk type*: *Standard* HDD (*locally-redundant storage*)

  - **Networking**
    - *Virtual Network*: Default-DEV-vnet
    - *Subnet*: DefaultMainSubnet_dev (192.168.0.0/27)
    - *Public IP*: Default-VM-DEV-pub

  - **Management**
    - *Auto-shutdown*: Disabled

  - **Monitoring**
    - *Boot Diagnostics*: Disabled



Basics

Subscription	Visual Studio Enterprise Subscription – MPN
Resource group	RG-DEFAULT-DEV
Virtual machine name	Default-VM-DEV
Region	Southeast Asia
Availability options	No infrastructure redundancy required
Zone options	Self-selected zone
Security type	Standard
Image	Ubuntu Server 24.04 LTS - Gen2
VM architecture	x64
Size	Standard B1ls (1 vcpu, 0.5 GiB memory)
Enable Hibernation	No
Authentication type	SSH public key
Username	Default-admin-dev
SSH Key format	RSA
Key pair name	Default-key-dev
Public inbound ports	SSH
Azure Spot	No

Disks

OS disk size	Image default
OS disk type	Standard HDD LRS
Use managed disks	Yes
Delete OS disk with VM	Enabled
Ephemeral OS disk	No

Networking

Virtual network	Default-DEV-vnet
Subnet	DefaultMainSubnet_dev (192.168.1.0/27)
Public IP	(new) Default-VM-DEV-pub
Accelerated networking	Off
Place this virtual machine behind an existing load balancing solution?	No
Delete public IP and NIC when VM is deleted	Disabled

Management

Microsoft Defender for Cloud	Standard
System assigned managed identity	Off
Login with Microsoft Entra ID	Off
Auto-shutdown	Off
Backup	Disabled
Enable periodic assessment	Off
Enable hotpatch	Off
Patch orchestration options	Image Default

Monitoring

Alerts	Off
Boot diagnostics	Off
Enable OS guest diagnostics	Off
Enable application health monitoring	Off

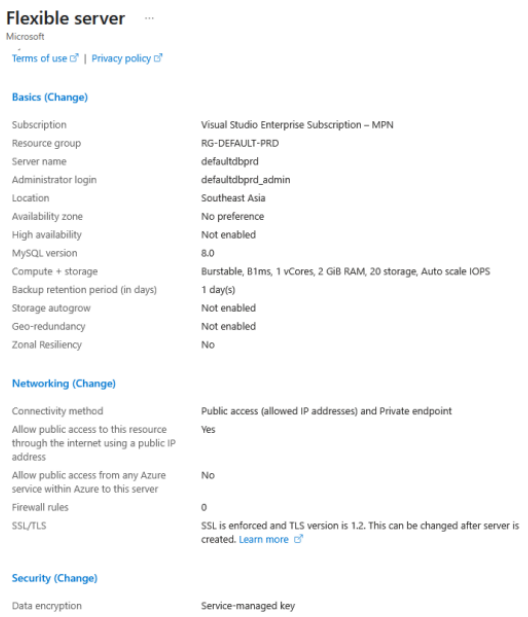
Advanced

Extensions	None
VM applications	None
Cloud init	No
User data	No
Disk controller type	SCSI
Proximity placement group	None
Capacity reservation group	None

Gambar 3.9 Konfigurasi Default-VM-DEV

Setelah membuat *virtual machine,* akan dibuat *Azure database* pada vnet yang ada. Infrastruktur ini memiliki 2 buah DB (defaultdbprd & defaultdbdev). Dalam penelitian ini kita akan menggunakan konfigurasi sebagai berikut:

- ***defaultdbprd***
  - **Basics**
    - *Resource Group*: RG-DEFAULT-PRD
    - *Server name*: defaultdbprd
    - *Workload type*: *For development and hobby projects*
    - *Store autogrow*: *Disabled*
    - *Backup retention:* 1 *day*
    - *Authentication method*: MySQL *authentication only*
    - *Administrator login*: defaultdbprd_admin
    - *Administrator password*: root123!



Flexible server ···

Microsoft

Terms of use | Privacy policy

Basics (Change)

Subscription	Visual Studio Enterprise Subscription – MPN
Resource group	RG-DEFAULT-PRD
Server name	defaultdbprd
Administrator login	defaultdbprd_admin
Location	Southeast Asia
Availability zone	No preference
High availability	Not enabled
MySQL version	8.0
Compute + storage	Burstable, B1ms, 1 vCores, 2 GiB RAM, 20 storage, Auto scale IOPS
Backup retention period (in days)	1 day(s)
Storage autogrow	Not enabled
Geo-redundancy	Not enabled
Zonal Resiliency	No

Networking (Change)

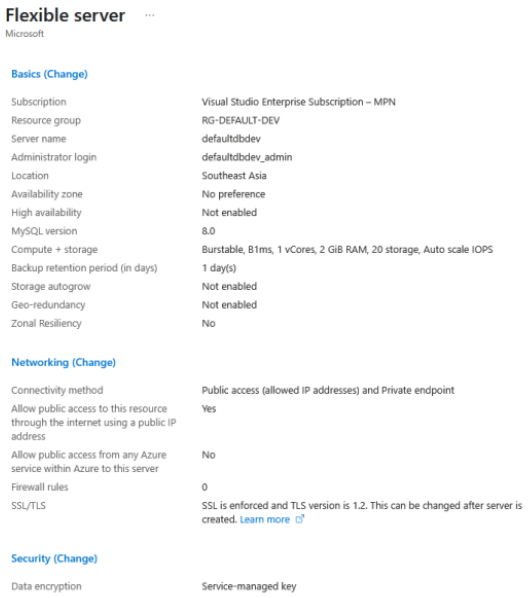
Connectivity method	Public access (allowed IP addresses) and Private endpoint
Allow public access to this resource through the internet using a public IP address	Yes
Allow public access from any Azure service within Azure to this server	No
Firewall rules	0
SSL/TLS	SSL is enforced and TLS version is 1.2. This can be changed after server is created. Learn more

Security (Change)

Data encryption	Service-managed key

Gambar 3.10 Konfigurasi DB defaultdbprd

- ***defaultdbdev***
  - **Basic**
    - *Resource Group*: RG-DEFAULT-DEV
    - *Server name*: defaultdbdev
    - *Workload type*: *For development and hobby projects*
    - *Store autogrow*: *Disabled*
    - *Backup retention:* 1 *day*
    - *Authentication method*: MySQL *authentication only*
    - *Administrator login*: defaultdbdev_admin
    - *Administrator password*: root123!



Flexible server ···
Microsoft

Basics (Change)

Subscription	Visual Studio Enterprise Subscription – MPN
Resource group	RG-DEFAULT-DEV
Server name	defaultdbdev
Administrator login	defaultdbdev_admin
Location	Southeast Asia
Availability zone	No preference
High availability	Not enabled
MySQL version	8.0
Compute + storage	Burstable, B1ms, 1 vCores, 2 GiB RAM, 20 storage, Auto scale IOPS
Backup retention period (in days)	1 day(s)
Storage autogrow	Not enabled
Geo-redundancy	Not enabled
Zonal Resiliency	No

Networking (Change)

Connectivity method	Public access (allowed IP addresses) and Private endpoint
Allow public access to this resource through the internet using a public IP address	Yes
Allow public access from any Azure service within Azure to this server	No
Firewall rules	0
SSL/TLS	SSL is enforced and TLS version is 1.2. This can be changed after server is created. Learn more

Security (Change)

Data encryption	Service-managed key

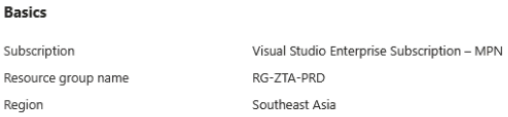
Gambar 3.11 Konfigurasi DB defaultdbdev

Pada tahap ini, seluruh komponen/ *resource* yang sudah didesain pada diagram rancangan arsitektur *project-default* sudah dibuat dan siap digunakan. Dalam infrastruktur ini, *developer* dapat melakukan upload *source code* pada VM yang tersedia dan melakukan *hosting* dengan menggunakan *runtime* yang diinginkan. *Database engineer* dapat mengkonfigurasi lebih lanjut pada laman basis data di *Azure portal*, sedangkan *Cloud engineer* dapat mengakses *resource* yang dibuat melalui *Azure portal* dengan menggunakan akun *Microsoft* yang sudah terdaftar dalam *subscription* ini.

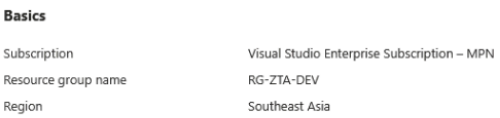
3.2.3 Pembuatan Infrastruktur *Project-ZTA*

Pada bagian ini, akan dijelaskan proses pembuatan infrastruktur *cloud* menggunakan *Microsoft Azure*. Bagian ini akan menjelaskan infrastruktur pertama, yaitu "Infrastruktur *Project-ZTA*" yang akan menjelaskan prosedur pembuatan infrastruktur *Project-ZTA* dan implementasi konsep ZTA pada arsitektur *cloud*. Seluruh prosedur akan dilakukan melalui *Azure Portal* yang diakses di *web browser*. Diagram arsitektur yang akan digunakan adalah diagram *Project-ZTA* pada Gambar 3.3.

Sama dengan prosedur yang dilakukan sebelumnya, hal pertama yang perlu dibuat adalah 2 buah *resource group* yang ada pada arsitektur *Project-ZTA*. Kedua *resource group* ini akan dibuat di *region* asia tenggara (*Southeast Asia*).



Gambar 3.12 Konfigurasi RG-ZTA-PRD



Gambar 3.13 Konfigurasi RG-ZTA-DEV

Setelah *resource group* terbuat, selanjutnya akan dibuat *virtual network* yang digunakan untuk keperluan *private network* pada lingkungan *cloud*. *Virtual network* ini akan dikonfigurasi dengan *subnet* yang ada pada desain arsitektur *Project-ZTA*.

Pada konfigurasi *virtual network* ini, *virtual network encryption* tidak perlu diaktifkan karena hanya ada satu buah *virtual machine* pada

arsitektur ini. Fitur ini dapat melakukan enkripsi anta koneksi *virtual machine* yang ada dalam suatu *network*. Fitur *Azure Firewall* dan *Azure DdoS* Protection juga tidak akan di aktivasi, melainkan akan dikompensasi melalui komponen *web application firewall* yang berada pada *application gateway*. Sedangkan *Azure Basiton* yang merupakan salah satu cara untuk mengammankan koneksi RDP/SSH ke dalam *virtual machine* dapat dikompensasi dengan pembukaan/penutupan akses *port* SSH hanya pada saat diperlukan.

Berikut merupakan konfigurasi yang dilakukan pada *virtual network* yang ada:

RG-ZTA-PRD

- ZTA-PRD-vnet:
  - *Resource Group*: RG-ZTA-PRD
  - *Name*: ZTA-PRD-vnet
  - *IP addresses*: 192.169.0.0/25
  - *Subnet*:
    - ZTAMainSubnet (192.169.0.0/27)
    - ZTADBSubnet (192.169.0.32/27)
    - GatewaySubnet (192.169.0.64/27)
    - ZTAGWSubnet (192.169.0.96/27)

Name ↑	IPv4
ZTAMainSubnet	192.169.0.0/27
GatewaySubnet	192.169.0.64/27
ZTADBSubnet	192.169.0.32/27
ZTAAGWSubnet	192.169.0.96/27

Gambar 3.14 Konfigurasi ZTA-PRD-vnet

RG-ZTA-DEV

- ZTA-DEV-vnet:
    - *Resource Group*: RG-ZTA-DEV
    - *Name*: ZTA-DEV-vnet
    - *IP addresses*: 192.169.1.0/25
    - *Subnet*:
        - ZTAMainSubnet_dev (192.169.1.0/27)
        - ZTADBSubnet_dev (192.169.1.32/27)
        - GatewaySubnet (192.169.1.64/27)
        - ZTAGWSubnet_dev(192.169.1.96/27)

Name ↑	IPv4
ZTAMainSubnet_dev	192.169.1.0/27
GatewaySubnet	192.169.1.64/27
ZTADBSubnet_dev	192.169.1.32/27
ZTAAGWSubnet_dev	192.169.1.96/27

Gambar 3.15 Konfigurasi ZTA-DEV-vnet

Pada *virtual network* ini, terdapat tambahan 2 buah *subnet* di masing- masing *virtual network* yang disebut dengan *GatewaySubnet*, *subnet* ini akan dipergunakan untuk keperluan akses seluruh *resource* yang ada pada *virtual network* tersebut walaupun hanya mempunyai *private* IP. Sedangkan *ZTAGWSubnet* akan digunakan sebagai jaringan untuk *Application Gateway*.

Resource selanjutnya yang akan dikonfigurasi adalah *virtual machine* dan *database* yang merupakan komponen utama dalam arsitektur ini. Berikut merupakan konfigurasi yang digunakan:

- ***ZTA-VM-PRD***
    - **Basic**
        - *Resource group*: RG-ZTA-PRD
        - *Name*: ZTA-VM-PRD

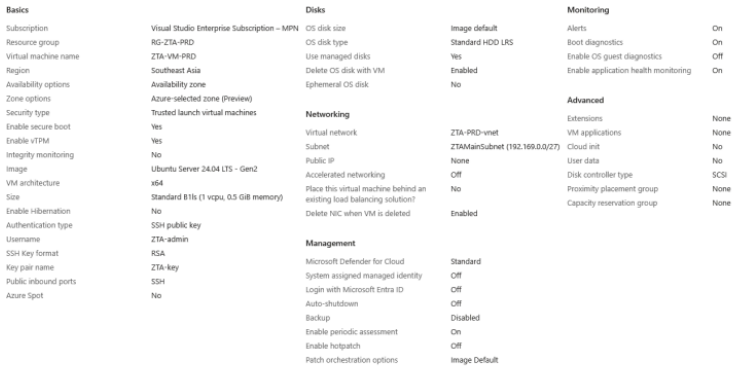
- *Availability options*: *Availability Zone (Azure Selected)*
- *Security type*: *Standard*
- *Machine Type*: *Standard*_B1 ls (1 vCPU, 0.5 GiB *memory*)
- *Image*: *Ubuntu Server* 24.04 LTS- x64 *Gen*2
- *Authenticaiton Type*: SSH *public key*
- *Username*: *ZTA-admin*
- *Key pair name*: *ZTA-key*
- *Inbound ports*: SSH (22)

- **Disk**
  - *OS Disk size: Image default (30 GiB)*
  - *OS Disk type: Standard HDD (locally-redundant storage)*
  - *Encryption at Host: Enabled*

- **Networking**
  - *Virtual Network: ZTA-PRD-vnet*
  - *Subnet: ZTAMainSubnet (192.169.0.0/27)*
  - *Public IP: -*

- **Management**
  - *Auto-shutdown: Disabled*
  - *Enable Periodic Assessment: Enabled*

- **Monitoring**
  - *Boot Diagnostics: Enabled with managed storage account*
  - *Alert rules: Enabled, Default config*
  - *Health monitoring: Enabled*



Basics		Disks		Monitoring	
Subscription	Visual Studio Enterprise Subscription – MPN	OS disk size	Image default	Alerts	On
Resource group	RG-ZTA-PRD	OS disk type	Standard HDD LRS	Boot diagnostics	On
Virtual machine name	ZTA-VM-PRD	Use managed disks	Yes	Enable OS guest diagnostics	Off
Region	Southeast Asia	Delete OS disk with VM	Enabled	Enable application health monitoring	On
Availability options	Availability zone	Ephemeral OS disk	No		
Zone options	Azure-selected zone (Preview)			Advanced	
Security type	Trusted launch virtual machines	Networking		Extensions	None
Enable secure boot	Yes	Virtual network	ZTA-PRD-vnet	VM applications	None
Enable vTPM	Yes	Subnet	ZTAMainSubnet (192.169.0.0/27)	Cloud init	No
Integrity monitoring	No	Public IP	None	User data	No
Image	Ubuntu Server 24.04 LTS - Gen2	Accelerated networking	Off	Disk controller type	SCSI
VM architecture	x64	Place this virtual machine behind an existing load balancing solution?	No	Proximity placement group	None
Size	Standard B1ls (1 vcpu, 0.5 GiB memory)	Delete NIC when VM is deleted	Enabled	Capacity reservation group	None
Enable Hibernation	No				
Authentication type	SSH public key	Management			
Username	ZTA-admin	Microsoft Defender for Cloud	Standard		
SSH Key format	RSA	System assigned managed identity	Off		
Key pair name	ZTA-key	Login with Microsoft Entra ID	Off		
Public inbound ports	SSH	Auto-shutdown	Off		
Azure Spot	No	Backup	Disabled		
		Enable periodic assessment	On		
		Enable hotpatch	Off		
		Patch orchestration options	Image Default		

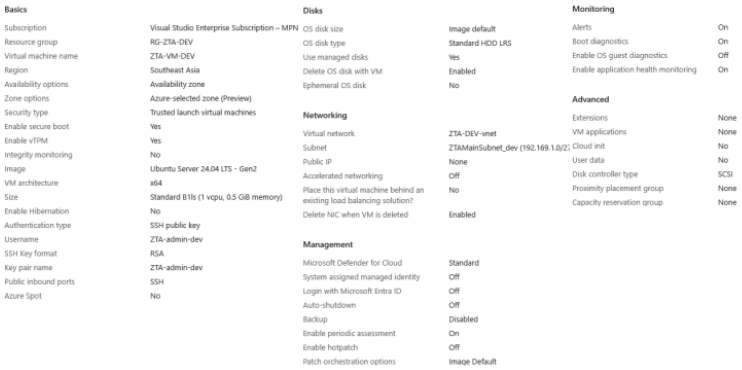
Gambar 3.16 Konfigurasi ZTA-VM-PRD

- ***ZTA-VM-DEV***
  - **Basic**
    - *Resource group*: RG-ZTA-DEV
    - *Name*: ZTA-VM-DEV
    - *Availability options*: *No Infrastructure Redundancy*
    - *Security type*: *Standard*
    - *Machine Type*: *Standard*_B1 ls (1 vCPU, 0.5 GiB *memory*)
    - *Image*: *Ubuntu Server* 24.04 LTS- x64 *Gen*2
    - *Authenticaiton Type*: SSH *public key*
    - *Username*: *ZTA-admin-dev*
    - *Key pair name*: *ZTA-key-dev*
    - *Inbound ports*: SSH (22)
  - **Disk**
    - OS *Disk size*: *Image default* (30 GiB)
    - *OS Disk type*: *Standard* HDD (*locally-redundant storage*)
    - *Encryption at Host: Enabled*

- **Networking**
  - *Virtual Network: ZTA-PRD-vnet*
  - *Subnet: ZTAMainSubnet (192.169.0.0/27)*
  - *Public IP: -*

- **Management**
  - *Auto-shutdown: Disabled*
  - *Enable Periodic Assessment: Disabled*

- **Monitoring**
  - *Boot Diagnostics: Enabled with managed storage account*
  - *Alert rules: Enabled, Default config*
  - *Health monitoring: Enabled*



Basics	
Subscription	Visual Studio Enterprise Subscription – MPN
Resource group	RG-ZTA-DEV
Virtual machine name	ZTA-VM-DEV
Region	Southeast Asia
Availability options	Availability zone
Zone options	Azure-selected zone (Preview)
Security type	Trusted launch virtual machines
Enable secure boot	Yes
Enable vTPM	Yes
Integrity monitoring	No
Image	Ubuntu Server 24.04 LTS - Gen2
VM architecture	x64
Size	Standard B1ls (1 vcpu, 0.5 GiB memory)
Enable Hibernation	No
Authentication type	SSH public key
Username	ZTA-admin-dev
SSH Key format	RSA
Key pair name	ZTA-admin-dev
Public inbound ports	SSH
Azure Spot	No

Disks	
OS disk size	Image default
OS disk type	Standard HDD LRS
Use managed disks	Yes
Delete OS disk with VM	Enabled
Ephemeral OS disk	No

Networking	
Virtual network	ZTA-DEV-vnet
Subnet	ZTAMainSubnet_dev (192.169.1.0/2;
Public IP	None
Accelerated networking	Off
Place this virtual machine behind an existing load balancing solution?	No
Delete NIC when VM is deleted	Enabled

Management	
Microsoft Defender for Cloud	Standard
System assigned managed identity	Off
Login with Microsoft Entra ID	Off
Auto-shutdown	Off
Backup	Disabled
Enable periodic assessment	On
Enable hotpatch	Off
Patch orchestration options	Image Default

Monitoring	
Alerts	On
Boot diagnostics	On
Enable OS guest diagnostics	Off
Enable application health monitoring	On

Advanced	
Extensions	None
VM applications	None
Cloud init	No
User data	No
Disk controller type	SCSI
Proximity placement group	None
Capacity reservation group	None

Gambar 3.17 Konfigurasi ZTA-VM-DEV

Pada konfigurasi VM di *Project-ZTA*, dapat diketahui ada beberapa fitur yang diaktifkan. Hal ini guna memenuhi prinsip- prinsip dalam konsep ZTA, yaitu selalu dilakukannya pemantauan (*monitoring*) yang dimana dapat dalam VM, dapat dipenuhi dengan menggunakan *resource health*

monitoring dan *alerting*. Lalu juga ada prinsip dimana diperlukannya pengamanan dalam setiap komponen yang ada, hal ini dapat dipenuhi dengan menggunakan fitur *encryption at host* pada *disk* yang ada dalam VM tersebut. Fitur *boot diagnostics* juga diaktifkan untuk memastikan VM berjalan dengan versi atau *patch* paling baru. *Zone redundancy* juga digunakan

- ***ztadbprd***
  - **Basics**
    - *Resource Group*: RG-ZTA-PRD
    - *Server name*: *ztadbprd*
    - *Workload type*: *For development and hobby projects*
    - *Store autogrow*: *Disabled*
    - *Backup retention: 7 day*
    - *Authentication method*: MySQL *authentication only*
    - *Administrator login*: ztadbprd_admin
    - *Administrator password*: root123!
  - **Networking**
    - *Connectivity Method: Private Access*
    - *Virtual Network*: *ZTA-PRD-vnet*
    - *Subnet*: ZTA-PRD-vnet/ZTADBSubnet
    - *Private DNS Zone*: (New), Default

Basics (Change)

Subscription	Visual Studio Enterprise Subscription – MPN
Resource group	RG-ZTA-PRD
Server name	ztadbprd
Administrator login	ztadbprd_admin
Location	Southeast Asia
Availability zone	No preference
High availability	Not enabled
MySQL version	8.0
Compute + storage	Burstable, B1ms, 1 vCores, 2 GiB RAM, 20 storage, Auto scale IOPS
Backup retention period (in days)	7 day(s)
Storage autogrow	Not enabled
Geo-redundancy	Not enabled
Zonal Resiliency	No

Networking (Change)

Connectivity method	Private access (VNet Integration)
Virtual network subscription	Visual Studio Enterprise Subscription – MPN
Virtual network resource group	RG-ZTA-PRD
Virtual network	ZTA-PRD-vnet
Delegated subnet	ZTADBSubnet
Private DNS zone subscription	Visual Studio Enterprise Subscription – MPN
Private DNS zone resource group	RG-ZTA-PRD
Private DNS zone	(New) ztadbprd.private.mysql.database.azure.com

Security (Change)

Data encryption	Service-managed key

Gambar 3.18 Konfigurasi *ztadbprd*

- ***ztadbdev***
  - **Basics**
    - *Resource Group*: RG-ZTA-DEV
    - *Server name*: *ztadbdev*
    - *Workload type*: *For development and hobby projects*
    - *Store autogrow*: *Disabled*
    - *Backup retention: 7 day*
    - *Authentication method*: MySQL *authentication only*

- *Administrator login*: ztadbdev_admin
- *Administrator password*: root123!

○ **Networking**

- *Connectivity Method: Private Access*
- *Virtual Network*: *ZTA-DEV-vnet*
- *Subnet*: ZTA-DEV-vnet/ZTADBSubnet_dev
- *Private DNS Zone*: (New), Default

Basics (Change)

Subscription	Visual Studio Enterprise Subscription – MPN
Resource group	RG-ZTA-DEV
Server name	ztadbdev
Administrator login	ztadbdev_admin
Location	Southeast Asia
Availability zone	No preference
High availability	Not enabled
MySQL version	8.0
Compute + storage	Burstable, B1ms, 1 vCores, 2 GiB RAM, 20 storage, Auto scale IOPS
Backup retention period (in days)	7 day(s)
Storage autogrow	Not enabled
Geo-redundancy	Not enabled
Zonal Resiliency	No

Networking (Change)

Connectivity method	Private access (VNet Integration)
Virtual network subscription	Visual Studio Enterprise Subscription – MPN
Virtual network resource group	RG-ZTA-DEV
Virtual network	ZTA-DEV-vnet
Delegated subnet	ZTADBSubnet_dev
Private DNS zone subscription	Visual Studio Enterprise Subscription – MPN
Private DNS zone resource group	rg-zta-prd
Private DNS zone	ztadbprd.private.mysql.database.azure.com

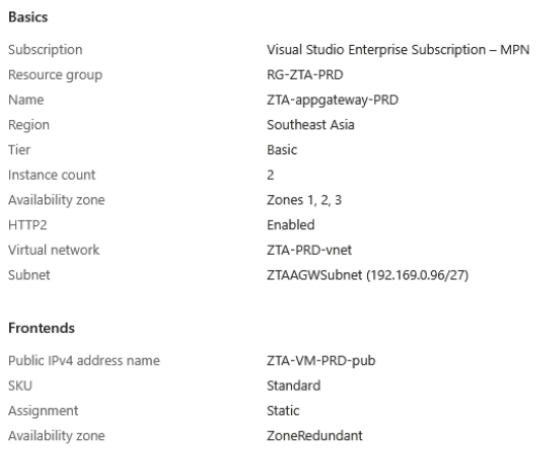
Security (Change)

Data encryption	Service-managed key

Gambar 3.19 Konfigurasi *ztadbdev*

Setelah terbuatnya komponen utama yang ada pada *Project_ZTA*, selanjutnya akan dilanjutkan dengan konfigurasi 2 buah komponen tambahan pada kedua *resource group* yang ada, yaitu *application gateway* dan *virtual network gateway*. Berikut merupakan konfigurasi yang akan digunakan untuk *application gateway* yang ada:

- ***ZTA-appgateway-PRD***
  - **Basics**
    - *Resource Group*: RG-ZTA-PRD
    - *Name*: ZTA-appgateway-PRD
    - *Virtual Network:* ZTA-PRD-vnet
    - *Subnet:* ZTAGWSubnet *(192.169.0.96/27)*
  - **Frontends**
    - *Public* IP*:* ZTA-VM-PRD-pub



Basics

Subscription	Visual Studio Enterprise Subscription – MPN
Resource group	RG-ZTA-PRD
Name	ZTA-appgateway-PRD
Region	Southeast Asia
Tier	Basic
Instance count	2
Availability zone	Zones 1, 2, 3
HTTP2	Enabled
Virtual network	ZTA-PRD-vnet
Subnet	ZTAAGWSubnet (192.169.0.96/27)

Frontends

Public IPv4 address name	ZTA-VM-PRD-pub
SKU	Standard
Assignment	Static
Availability zone	ZoneRedundant

Gambar 3.20 Konfigurasi *ZTA-appgateway-PRD*

- ***ZTA-appgateway-DEV***
  - **Basics**
    - *Resource Group*: RG-ZTA-DEV

- *Name*: ZTA-appgateway-DEV
- *Virtual Network:* ZTA-DEV-vnet
- *Subnet:* ZTAGWSubnet-dev *(192.169.1.96/27)*

- **Frontends**
  - *Public* IP*:* ZTA-VM-DEV-pub

Basics

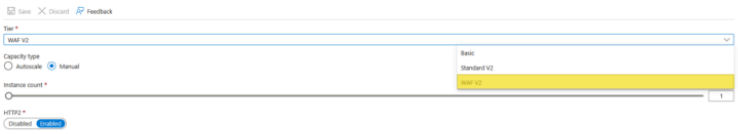
Subscription	Visual Studio Enterprise Subscription – MPN
Resource group	RG-ZTA-DEV
Name	ZTA-appgateway-DEV
Region	Southeast Asia
Tier	Basic
Instance count	2
Availability zone	Zones 1, 2, 3
HTTP2	Enabled
Virtual network	ZTA-DEV-vnet
Subnet	ZTAAGWSubnet_dev (192.169.1.96/27)

Frontends

Public IPv4 address name	ZTA-VM-DEV-pub
SKU	Standard
Assignment	Static
Availability zone	ZoneRedundant

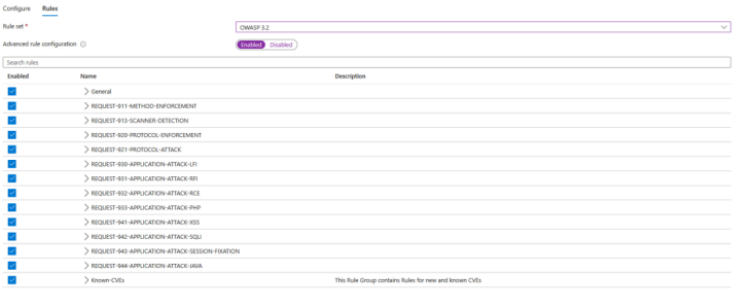
Gambar 3.21 Konfigurasi *ZTA-appgateway-DEV*

Setelah *Application Gateway* sudah selesai dibuat, selanjutnya kita perlu merubah beberapa konfigurasi yang ada pada masing masing *application gateway*. Hal yang akan diubah adalah *tier* yang digunakan dari *Basic* menjadi WAF v2. Hal ini dilakukan untuk mengaktifkan protokol keamanan *web application firewall* yang ada pada *application gateway* yang telah dibuat.



Gambar 3.22 Mengubah *Gateway Tier* pada *Application Gateway*

Dengan mengaktifkan *tier* WAF v2, *application gateway* akan mempunyai *firewall* yang bekerja dengan mengikuti salah satu protokol standar keaman dalam *web application*. Protokol yang akan digunakan pada penelitian ini adalah OWASP 3.2, protokol ini juga dapat dikonfigurasi lebih lanjut sesuai dengan kebijakan yang ada dengan mengaktifkan *advanced rule configuration*.



Gambar 3.23 Mengaktifkan WAF *Rules*.

Setelah semua konfigurasi yang harus dilakukan pada *application gateway* telah terpenuhi, akan dibuat satu komponen terakhir yang menjadi gerbang belakang bagi *developer* dan *engineer* dalam mengakses komponen yang ada di dalam lingkup *Project_ZTA*. Komponen ini adalah *Azure Virtual Network Gateway* yang akan dikonfigurasi menjadi *gateway* bagi koneksi VPN yang akan dibuat. Aplikasi VPN yang akan digunakan adalah *Azure VPN Client* yang merupakan salah satu dari beberapa VPN *client* yang dapat digunakan selaras dengan *gateway* ini. Berikut merupakan konfigurasi yang harus dilakukan pada *virtual network gateway*:

- ***ZTA-gateway-PRD***
  - **Basic**
    - *Resource Group*: RG-ZTA-PRD
    - *Name*: ZTA-gateway-PRD

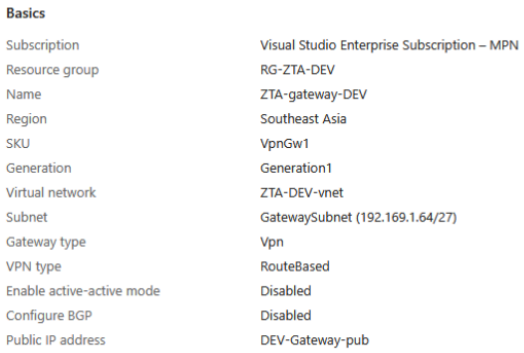
- SKU: VpnGw1
- *Generation*: 1
- *Virtual Network*: ZTA-PRD-vnet
- *Subnet*: GatewaySubnet (192.169.0.64/27)
- *Gateway type*: VPN
- *Public* IP: PRD-Gateway-pub

Basics

Subscription	Visual Studio Enterprise Subscription – MPN
Resource group	RG-ZTA-PRD
Name	ZTA-gateway-PRD
Region	Southeast Asia
SKU	VpnGw1
Generation	Generation1
Virtual network	ZTA-PRD-vnet
Subnet	GatewaySubnet (192.169.0.64/27)
Gateway type	Vpn
VPN type	RouteBased
Enable active-active mode	Disabled
Configure BGP	Disabled
Public IP address	PRD-Gateway-pub

Gambar 3.24 Konfigurasi *ZTA-gateway-PRD*

- ***ZTA-gateway-DEV***
  - **Basic**
    - *Resource Group*: RG-ZTA-DEV
    - *Name*: ZTA-gateway-DEV
    - SKU: VpnGw1
    - *Generation*: 1
    - *Virtual Network*: ZTA-DEV-vnet
    - *Subnet*: GatewaySubnet (192.169.1.64/27)
    - *Gateway type*: VPN
    - *Public* IP: DEV-Gateway-pub



Basics	
Subscription	Visual Studio Enterprise Subscription – MPN
Resource group	RG-ZTA-DEV
Name	ZTA-gateway-DEV
Region	Southeast Asia
SKU	VpnGw1
Generation	Generation1
Virtual network	ZTA-DEV-vnet
Subnet	GatewaySubnet (192.169.1.64/27)
Gateway type	Vpn
VPN type	RouteBased
Enable active-active mode	Disabled
Configure BGP	Disabled
Public IP address	DEV-Gateway-pub

Gambar 3.25 Konfigurasi *ZTA-gateway-DEV*

Setelah konfigurasi *virtual network gateway* selesai, selanjutnya akan dilakukan konfigurasi koneksi P2S dari *local device* ke *Azure VPN Gateway* yang telah dibuat. Hal ini dapat dilakukan dengan mengaktifkan dan melakukan konfigurasi VPN pada masing- masing *gateway*. Koneksi yang ada akan berbasis *self-signed certificate*, dimana *certificate* akan dibuat melalui *local device* yang akan terkoneksi dengan VPN *gateway* tersebut.

3.3 Instrumen Penelitian

Dalam penelitian ini, penulis akan menggunakan *Microsoft Azure* sebagai instrumen penelitiannya. Dalam penelitian ini, *Microsoft Azure* akan digunakan sebagai *cloud service provider* yang dimana akan menjadi tempat diimplementasikannya konsep *Zero Trust Architecture*. Penulis memilih *Microsoft Azure* sebagai instrumen penelitian karena *cloud provider* tersebut merupakan salah satu yang paling marak digunakan. Hal ini menjadi pertimbangan penulis agar hasil yang diberikan dapat bermanfaat bagi banyak penggunanya serta dapat memberikan ide dalam melakukan pengembangan keamanan di dalam *cloud*.

3.4 Objek Penelitian

Hal yang menjadi objek penelitian adalah konsep keamanan *Zero Trust Architecture* yang merupakan salah satu pengembangan kerangka kerja keamanan dalam dunia komputer. Konsep ZTA ini menjadi objek fokus dalam penelitian ini karena akan diimplementasikan kedalam *cloud environment*. Proses implementasi tersebut akan dilakukan di dalam *cloud environment Microsoft Azure* yang merupakan salah satu dari *cloud service provider* yang sangat marak digunakan.

BAB 4

HASIL PENELITIAN

4.1 Implementasi *Zero Trust Architecture*

Setelah selesai melakukan implementasi *zero trust architecure* sesuai dengan rancangan diagram yang telah dibuat, penulis akan memperlihatkan bagaimana pengaruh konsep keamanan *zero trust architecture* dalam *cloud environment* yang ada di *Microsoft Azure*. Implementasi yang berhasil merupakan implementasi yang memenuhi semua prinsip *zero trust* yang telah di bahas sebelumnya. Salah satu prinsip utama dalam *zero trust* adalah **seluruh sumber data dan layanan komputasi dan dianggap sebagai kesatuan sistem**, yang dimana sistem secara keseluruhan perlu dilindungi. Hal tersebut dapat dicapai dengan melakukan konfigurasi khusus pada komponen- komponen yang sudah ada maupun komponen tambahan. Berikut merupakan beberapa komponen yang telah dibuat dan digunakan dalam penelitian ini guna memenuhi prinsip *zero trust architecture*:

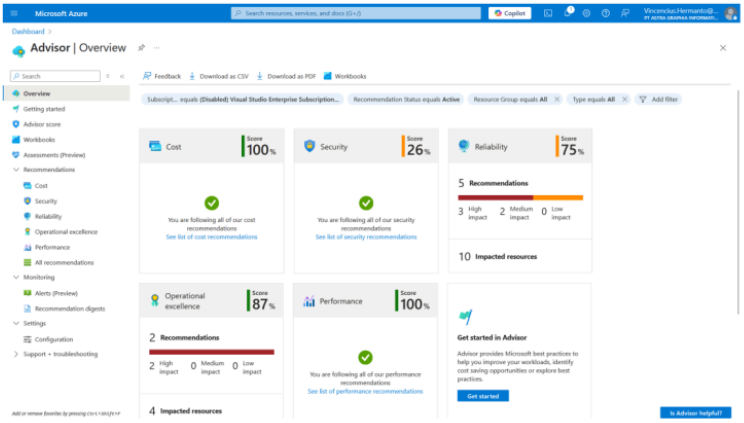
4.1.1 *Azure Advisor*

Azure advisor merupakan salah satu fitur keamanan yang ada pada *Azure*. Fitur ini dapat digunakan untuk memantau postur keamanan yang ada pada komponen atau layanan *cloud* yang ada. Dengan menggunakan *azure advisor*, dua prinsip *zero trust* akan terpenuhi yaitu "**Komponen yang ada di dalam sistem harus selalu dipantau dan dievaluasi tingkat keamanannya secara berkala *(continuous system monitoring dan evaluation)***" dan "**Informasi yang ada dalam komponen, jaringan, maupun akses yang diberikan harus selalu digunakan untuk analisa peningkatan postur keamanan *(system log analysis)***". *Azure advisor* tidak hanya melakukan pematauan pada kategori keamanan, melainkan juga melakukan pemantauan dalam 5 kategori atau pilar lainnya, yaitu: keuangan

(*cost*), keamanan (*security*), reabilitas (*reability*), operasional (*operational excellence*), dan performa (*performance*).

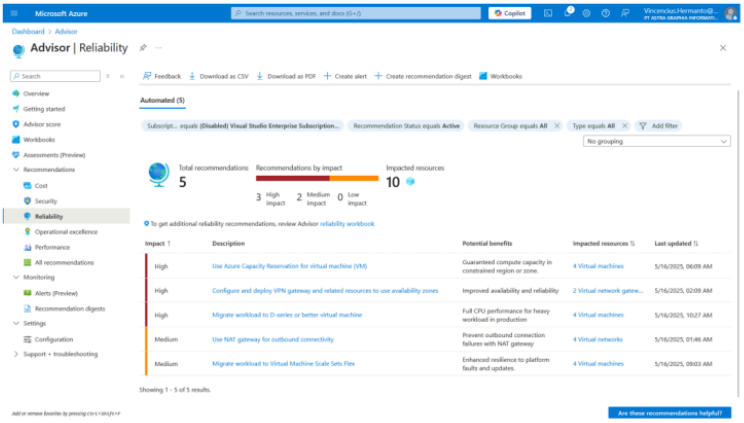
Advisor sendiri akan melakukan *auto-scan* setiap interval 24 jam dan memberikan laporan terhadap temuan yang ada kedalam masing- masing kategori tersebut.

- Keuangan (*cost*) : Mengidentifikasi komponen atau layanan *cloud* untuk memberikan rekomendasi yang dapat menurunkan pengeluaran.
- Keamanan (*security*) : Mengidentifikasi kelemahan dalam postur keamanan masing- masing komponen untuk memberikan rekomendasi keamanan.
- Reabilitas (*reability*) : Menidentifikasi permasalahan ketersediaan (*availability*) dan reabilitas dari komponen atau layanan yang ada.
- Operasional (*operational excellence*) : Mengidentifikasi komponen dan layanan yang ada untuk memberikan rekomendasi yang dapat meningkatkan operasional.
- Performa (*performance*) : Mengidentifikasi penggunaan performa dari masing- masing komponen atau layanan *cloud* untuk meningkatkan efektivitas pada konfigurasi yang ada.



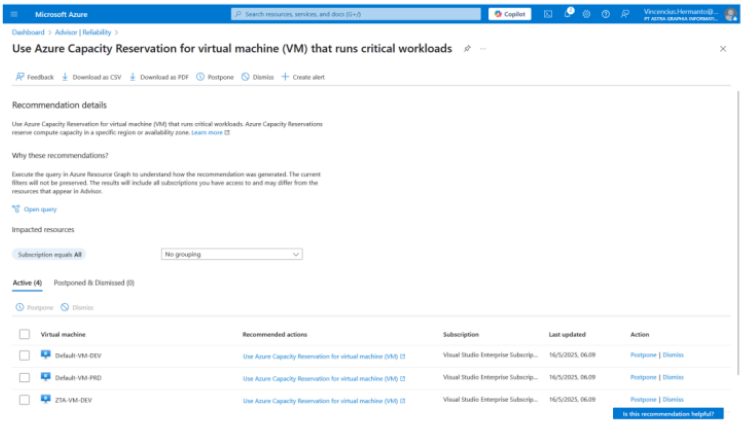
Gambar 4.1 Tampilan *Dashboard Azure Advisor*

Pada tampilan *dashboard* utama yang dimiliki oleh *advisor* terdapat beberapa hal yang dapat dicermati, yaitu terdapat persentase skor dari masing- masing kategori yang ada. Skor tersebut mereprentasikan nilai keseluruhan yang diperoleh dalam kategori tersebut. Tampilan yang ada juga dapat di *filter* sesuai dengan kebutuhan, mulai dari *subscriptions*, *resources*, dsb. Hasil dari laporan ini juga dapat di-*export* kedalam bentuk *comma-separated values* (CSV) dan *portable document format* (PDF) untuk keperluan dokumentasi.



Gambar 4.2 Tampilan *Dashboard* dalam Kategori pada *Advisor*

Setelah memilih suatu kategori, maka akan muncul tampilan *dashboard* yang berisi daftar isu bereserta dengan *impact*, deskripsi, *potential benefits*, layanan yang terpengaruh, dan waktu *update* informasi terakhir. Pada *dashbard* ini juga dapat dilakukan filtrasi sesuai dengan kebutuhan.

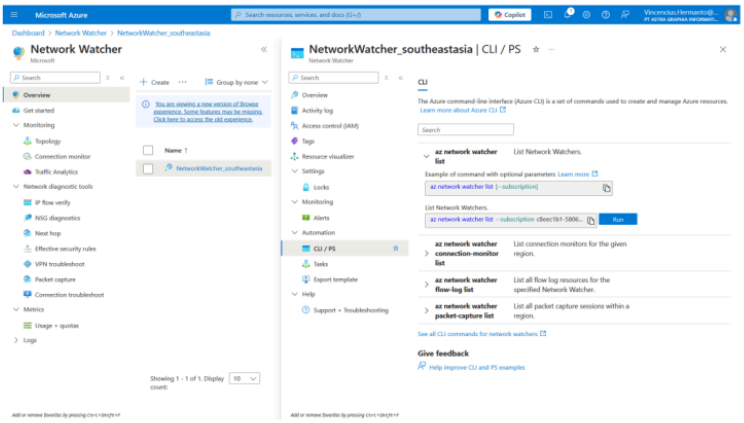


Gambar 4.3 Tampilan *Issue* pada *Advisor*

Setiap kategori dapat memberikan laporan yang detail berserta dengan solusi yang dapat digunakan. Apabila terdapat isu yang dianggap tidak relevan atau konfigurasi yang ada sudah sesuai dengan kebijakan atau keperluan, isu tersebut dapat diabaikan (*dismiss*) atau ditunda (*postponed*).

4.1.2 *Azure Network Watcher*

Azure Network Watcher merupakan salah satu fitur yang terdapat pada *Azure Portal* yang dapat digunakan untuk pemantauan (*monitoring*), diagnosa jaringan (*network diagnostic*), dan visualisasi lalu lintas jaringan (*traffic visualization*). Fitur ini akan memenuhi prinsip yang sama seperti *Azure Advisor*, yaitu "**Komponen yang ada di dalam sistem harus selalu dipantau dan dievaluasi tingkat keamanannya secara berkala *(continuous system monitoring dan evaluation)***" dan "**Informasi yang ada dalam komponen, jaringan, maupun akses yang diberikan harus selalu digunakan untuk analisa peningkatan postur keamanan *(system log analysis)***", dengan memberikan informasi yang lebih mendalam terhadap keperluan analisa jaringan.



Gambar 4.4 Tampilan *Azure Network Watcher*

Dengan menggunakan *network watcher*, tim keamanan sistem dapat melakukan *monitoring* terhadap jaringan yang ada. *Network watcher* juga dapat membantu *engineer* dalam melakukan *troubleshooting* terhadap kendala jaringan yang ada pada komponen atau layanan yang telah dibuat. Setiap *region* yang digunakan pada komponen ataupun layanan yang ada juga dapat di-*monitor* dan dianalisa secara tersendiri melalui fitur CLI (*comman line interface*) yang ada, dengan memilih region pada bagian *overview*.

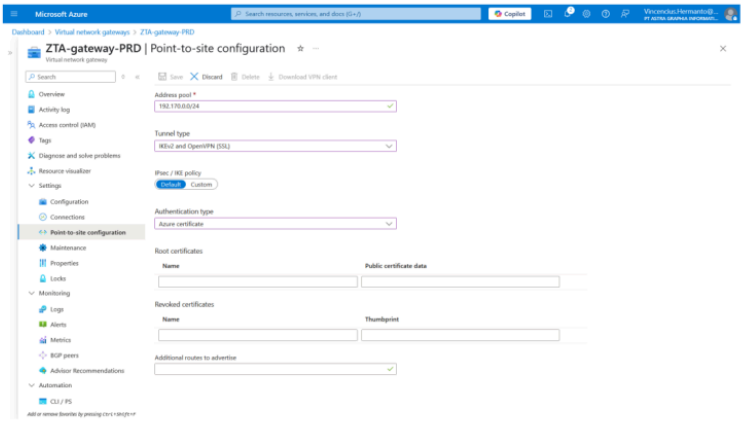
4.1.3 *Azure Virtual Network Gateway*

Azure Virtual Network Gateway yang telah dikonfigurasi sebelumnya merupakan gerbang jaringan yang menghubungkan antara *Azure private virtual network* dengan *device* diluar jaringan tersebut dengan menggunakan gerbang VPN. Hal ini memenuhi salah satu prinsip *zero trust*, yaitu "**Seluruh komunikasi yang terjadi antar komponen harus secara aman dimanapun lokasi komponen yang berkomunikasi *(secure line communication)***". Dengan menggunakan VPN *gateway* hubungan antara jaringan internal dengan perangkat yang terhubung dari jaringan eksternal dapat diamankan sesuai dengan prinsip yang ada.

Dalam menghubungkan perangkat eksternal dengan jaringan internal *Azure*, jaringan VPN harus terlebih dahulu dibuat. Sesuai dengan desain arsitektur sistem, VPN yang akan digunakan adalah jenis VPN *point-to-site*(P2S) menggunakan *Azure VPN Client* melalui VPN *tunnel* IKEv2 dan OpenVPN (SSL) menggunakan *Azure Certificate* sebagai metode otentikasinya.

Alasan penggunaan jenis VPN P2S adalah hubungan yang akan di buat merupakan koneksi antara *Azure internal network* dengan perangkat seperti *personal computer*, sedangkan apabila ingin membuat koneksi antara jaringan *Azure* dengan jaringan *server on-premises* maka harus

menggunakan jenis VPN *site-to-site*. Jenis *tunnel* yang digunakan adalah IKEv2 dan OpenVPN(SSL) dikarenan tipe *tunnel* ini memiliki metode otentikasi yang fleksibel dan dapat digunakan dengan VPN *client* lainnya. Berikut merupakan prosedur konfigurasi VPN P2S yang harus dilakukan:



Gambar 4.5 Konfigurasi awal VPN ZTA-gateway-PRD

1. Pada halaman VPN *gateway* di *Azure portal*, lakukan setup *point-to-site configuration* dan gunakan konfigurasi berikut:
   a. *Address pool*: 192.70.0.0/24
   b. *Tunnel type*: IKEv2 dan OpenVPN(SSL)
   c. *Ipsec/ IKE policy*: *Default*
   d. *Authentication type*: *Azure Authentication*

   Lanjutkan ke langkah berikutnya tanpa menutup halaman ini karena ada beberapa *field* yang perlu diisi menggunakan *certificate* yang akan dibuat.

2. Pada *desktop* perangkat eksternal, tekan ***Windows + R*** dan ketik **MMC**. Lalu pilih menu ***File*** lalu ***add/remove snap-in***, lalu pilih ***certificate***, klik ***OK***.
3. Jalankan *Windows Powershell*, gunakan script berikut:

```
$cert = New-SelfSignedCertificate -Type Custom
-KeySpec Signature -Subject
"CN=ZTARootVPN_PRD" -KeyExportPolicy
Exportable -HashAlgorithm sha256 -KeyLength
2048 -CertStoreLocation "Cert:\CurrentUser\My"
-KeyUsageProperty Sign -KeyUsage CertSign
```

4. . Lalu jalankan *command* berikut selanjutnya pada *powershell*:

```
New-SelfSignedCertificate -Type Custom -
DnsName ZTAChildCert -KeySpec Signature `

-Subject "CN=ZTAChildVPN_PRD" -
KeyExportPolicy Exportable `

-HashAlgorithm sha256 -KeyLength 2048 `

-CertStoreLocation "Cert:\CurrentUser\My" `

-Signer $cert -TextExtension
@("2.5.29.37={text}1.3.6.1.5.5.7.3.2")
```

5. Kembali ke *window* MMC, pada **ZTARootVPN_PRD**, klik kanan pilih ***All tasks***, lalu ***Export***. Gunakan konfigurasi "***No, do not export private key***" dan "***Base-64 encoded X.509 (.CER)***", pilih direktori tujuan *export* lalu klik ***Finish***.
6. Buka *certificate file* yang telah di-*export*, salin bagian *certificate* ke halaman konfigurasi *point-to-site* pada *Azure portal* di *field* ***root certificate***.
7. Klik ***Save***, lalu *download* VPN melalui tombol ***Download VPN Client***.
8. Pada perangkat eksternal, *download* ***Azure VPN Client*** melalui *Microsoft Store*.
9. Pada aplikasi *Azure* VPN *Client*, klik ***Import*** dan pilih file **azurevpnconfig.xml** yang terletak pada folder ***Azure VPN*** yang telah di-*download* melalui *Azure portal*.
10. Pada *Azure* VPN, gunakan konfigurasi berikut:

a. *Authentication Type*: *Certificate*
b. *Certificate Information*: *ZTAChildCert*

11. Setelah kedua VPN telah berhasil dikonfigurasi, maka koneksi VPN sudah dapat dibuat.

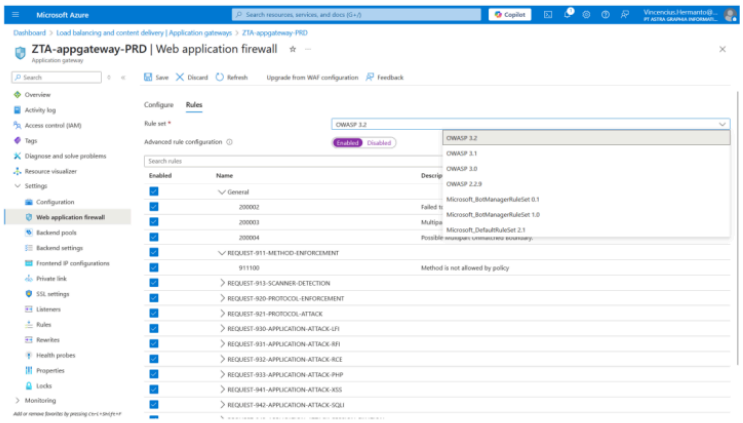
Dengan menggunakan VPN *point-to-site*, jaringan *private* yang berada pada *cloud environment* dapat diakses melalui perangkat eksternal melalui VPN *tunneling*. Hal ini memberikan tingkat keamanan yang lebih baik dibanding dengan membuka akses terhadap layanan atau komponen yang ada, dalam hal ini adalah koneksi dengan *virtual machine* yang ada pada jaringan *virtual network* dalam *Azure* hanya dengan menggunakan SSH *private key*. *Azure* VPN *Gateway* juga dapat digunakan sebagai alat pemantauan aktivitas pada *tunnel* yang ada, membuat deteksi akses pada waktu yang tidak lazim menjadi lebih cepat dan mudah.

4.1.4 *Azure Application Gateway*

Azure Application Gateway merupakan komponen yang memiliki 2 kegunaan utama dalam penelitian ini. Komponen ini akan menjadi gerbang depan yang menghubungkan *virtual machine* dengan *world wide web*. Dalam arsitektur ZTA ini, seluruh komponen seperti *virtual machine* dan *database* tidak memiliki IP publik yang dapat menghubungkan koneksi eksternal dengan komponen atau layanan yang ada pada *virtual network* pada *Azure*.

Komponen ini juga berguna sebagai *load balancer* yang dapat mengalihkan dan mengontrol lalu lintas yang ada, dalam hal ini lalu lintas menuju *virtual machine* yang berperan sebagai *web server*. *Application gateway* ini juga di konfigurasi untuk berperan sebagai *web application firewall* yang memiliki kegunaan yang sama dengan *firewall* pada umumnya, tetapi hanya melindungi jalur jaringan yang menuju ke *web server*. *Azure application gateway* juga ikut serta memenuhi prinsip *zero*

trust yaitu **"Seluruh komunikasi yang terjadi antar komponen harus secara aman dimanapun lokasi komponen yang berkomunikasi (*secure line communication*)"**.



Gambar 4.6 Konfigurasi *Firewall Rules* pada *Application Gateway*

Web application firewall yang ada pada *application gateway* ini juga dapat dikonfigurasi untuk melakukan *exception* terhadap paket yang ada apabila diperlukan. Salah satu keunggulan yang dimiliki *Azure* WAF adalah terdapat *ruleset* berdasarkan *ruleset* yang banyak digunakan, juga masing-masing *rules* pada *ruleset* tersebut dapat di konfigurasi lebih lanjut.

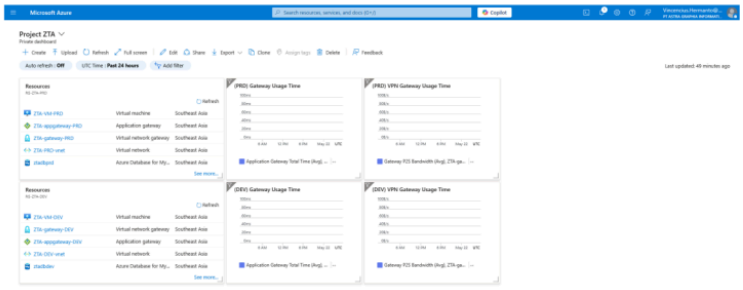
4.1.5 *Azure Resource Health and Monitoring*

Fitur *Azure resource health and monitoring* memiliki peran penting dalam memenuhi salah satu prinsip *zero trust*, yaitu "**Komponen yang ada di dalam sistem harus selalu dipantau dan dievaluasi tingkat keamanannya secara berkala *(continuous system monitoring dan evaluation)***". Fitur ini difokuskan kedalam komponen *virtual machine* yang menjadi layanan inti pada mayoritas arsitektur yang ada. Dalam konteks *virtual machine*, *Azure resource health and monitoring* memberikan

informasi terkait status kesehatan yang dipantau secara berkala setiap harinya secara otomatis melalui sistem *Azure*. Seluruh kejadian seperti kegagalan VM, VM terdampak oleh masalah layanan dari *Azure*, hingga penggunaan *resource* VM yang tidak lazim akan dilaporkan dan di tampilkan pada *resource health dashboard* ditunjukan sebagai *health events*.

Hal ini memungkinkan tim operasional dan tim keamanan yang ada untuk secara proaktif melakukan deteksi, respon, dan mitigasi potensi ancaman atau kegagalan sistem sebelum memiliki dampak yang besar. Dengan memanfaatkan fitur ini secara konsisten dalam operasional, organisasi atau perusahaan dapat memastikan bahwa *virtual machine* yang dimiliki selalu dalam keadaan yang sehat dan optimal.

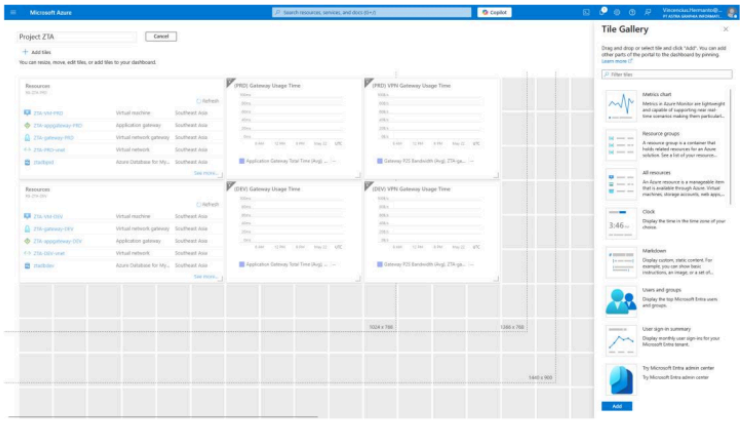
4.1.6 *Azure Dashboard*



Gambar 4.7 Tampilan *Azure Dashboard*

Azure dashboard merupakan komponen yang digunakan untuk membantu tim operasional maupun keamanan dalam melakukan pemantauan sistem secara *general*. Komponen ini memenuhi salah satu prinsip yang ada dalam *zero trust*, yaitu "**Komponen yang ada di dalam sistem harus selalu dipantau dan dievaluasi tingkat keamanannya secara berkala *(continuous system monitoring dan evaluation)***". Dengan menggunakan *Azure dashboard*, tim IT dapat membuat *dashboard* terpusat

untuk memvisualisasikan data secara *real-time* atas komponen- komponen yang dimiliki.



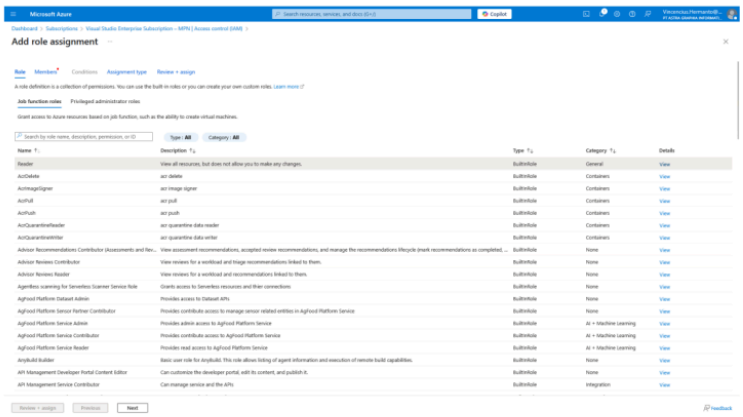
Gambar 4.8 Tampilan modifikasi *Azure Dashboard*

Dashboard ini juga dapat dimodifikasi sesuai dengan kebutuhan, mulai dari menampilkan metrik- metrik atas komponen yang ada, memanggil *Azure* REST API, hingga menampilkan utilitas lain seperti *video* maupun jam. Dengan menggunakan *Azure dashboard*, seluruh komponen yang ada pada sistem dapat dipantau dalam satu tampilan utama guna memudahkan proses *monitoring* dalam menjaga berjalan sistem yang ada.

4.1.7 *Azure Identity Access Management* (IAM)

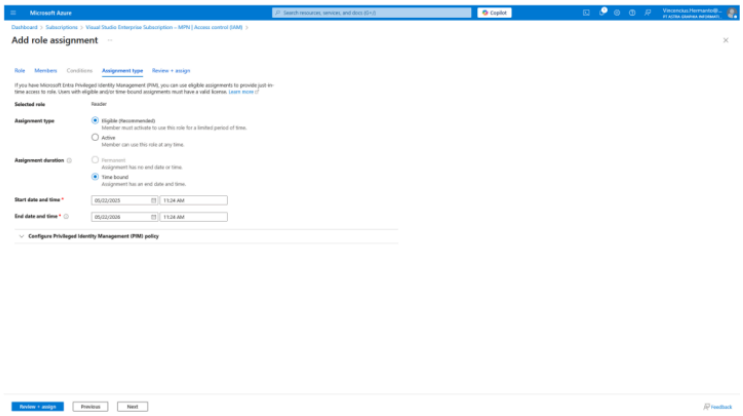
Azure identity access management (IAM) merupakan pusat kontrol atas hak akses yang diberikan kepada pengguna dalam mengelola komponen maupun layanan yang ada pada *Azure*. Dengan ini *Azure* IAM akan memenuhi tiga prinsip terakhir yang belum dapat terpenuhi oleh komponen maupun layanan lain yang telah dipaparkan, yaitu "**Akses sistem yang ada akan diberikan dengan batasan waktu (*time-limited access*)**", "**Akses yang diberikan akan ditentukan oleh kebijakan yang dinamis**

(***dynamic access policy***)", dan "**Seluruh akses yang diberikan harus selalu dievaluasi secara dinamis sebelum diotorisasi (*evaluated access*)**". *Azure* IAM memungkinkan pengaturan akses di seluruh komponen dan layanan *Azure*, dengan cangkupan yang fleksibel tergantung pada lebel pemberian akses, mulai dari *subscription* hingga ke sumber daya tertentu.



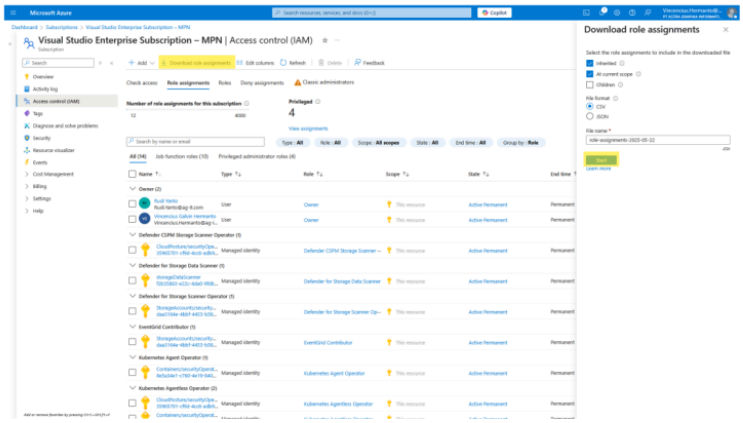
Gambar 4.9 Tampilan *Role Assignment* pada *Azure* IAM

Dalam memenuhi prinsip *evaluated access*, diketahui bahwa akses hanya dapat diberikan sesuai dengan batasan lingkup yang akan digunakan atau dalam *zerot trust* disebut dengan *Principle of Least Privilege* (PoLP). Hal ini dapat dilakukan pada IAM dengan memilih *role assignment* yang tepat dan sesuai.



Gambar 4.10 Tampilan *Assignment Type* pada *Azure* IAM

Dalam memenuhi prinsip *time limited access*, akses yang diberikan dapat diatur melalui *Azure* IAM. Pengaturan yang dapat diberikan mencakup *assignment type* yang diberikan antara *Eligible* untuk akses yang perlu diaktivasi dengan batasan waktu yang dapat ditentukan dan *Active* untuk akses yang tidak membutuhkan aktivasi terlebih dahulu. Dalam memberikan akses dengan tipe *Active*, durasi aktif dari akses tersebut juga dapat diatur dengan tipe *Permanent* untuk akses tanpa batasan waktu dan tipe *Time bound* untuk akses dengan batasan waktu yang dapat ditentukan.



Gambar 4.11 Tampilan Daftar Akses pada *Azure* IAM

Dalam memenuhi prinsip *evaluated access*, akses yang diberikan dapat selalu di modifikasi sesuai dengan perubahan kebijakan yang ada. Evaluasi akses dapat dilakukan secara berkala dalam interval waktu yang ditentukan masing- masing tim keamanan. Dalam melakukan evaluasi, seluruh akses yang ada pada suatu komponen maupun layanan, mulai dari *subscription* hingga masing- masing komponen dapat di ekspor kedalam bentuk file CSV maupun JSON.

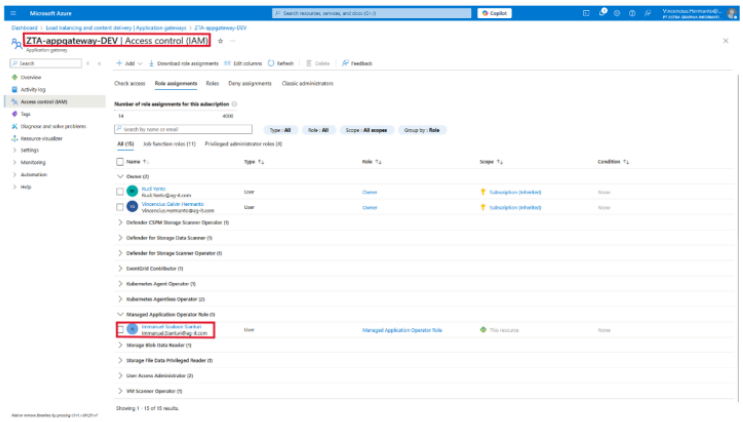
4.2 *Testing* Infrastruktur atas 7 Prinsip *Zero Trust*

Setelah membangun dua infrastruktur cloud yaitu *Project-Default* (tanpa Zero Trust Architecture) dan *Project-ZTA* (dengan penerapan Zero Trust Architecture) pada Microsoft Azure, dilakukan pengujian terhadap masing-masing infrastruktur untuk mengukur sejauh mana prinsip-prinsip *Zero Trust Architecture* (ZTA) dapat diterapkan. Pengujian ini bertujuan untuk mengamati secara langsung bagaimana setiap prinsip ZTA berperan dalam memperkuat keamanan sistem dan bagaimana perbedaan konfigurasi pada tiap proyek berkontribusi terhadap pencegahan akses yang tidak sah, pengawasan sistem, serta pengelolaan identitas.

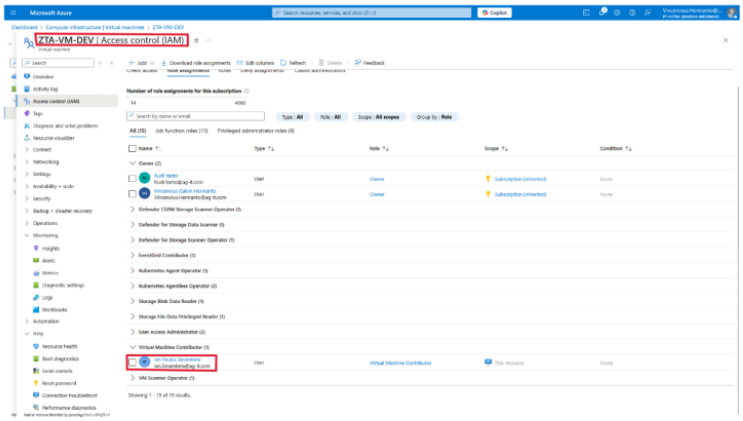
Berikut merupakan hasil pengujian dan penjelasan lebih lanjut terkait setiap prinsip ZTA:

1. **Resources Include Data and Services**

   Prinsip ini menekankan bahwa semua komponen, baik data maupun layanan, harus dianggap sebagai resource yang perlu diamankan secara individual. Dalam pengujian, dilakukan pemberian role assignment kepada dua *resource* berbeda dalam satu *subscription* yang sama. Hasilnya, akses yang diberikan ke satu *resource* tidak secara otomatis memberikan akses ke resource lainnya. Hal ini mencerminkan penerapan kontrol akses yang bersifat granular dan eksplisit. Dengan menggunakan Azure *Role-Based Access Control* (RBAC), administrator dapat menetapkan peran secara spesifik untuk setiap resource, memastikan bahwa pengguna hanya memiliki hak sesuai kebutuhan (*least privilege*), dan tidak lebih.



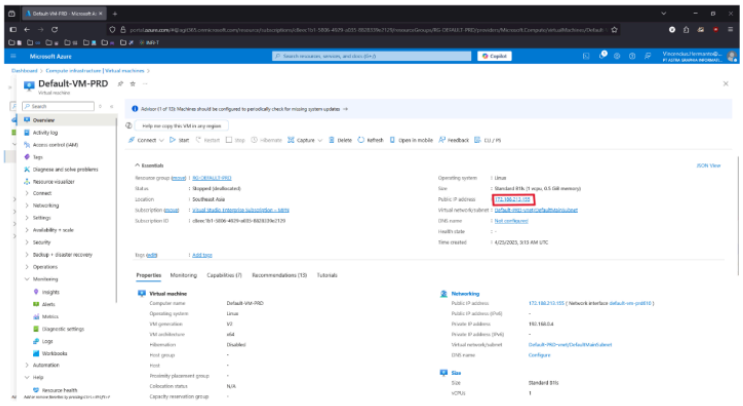
Gambar 4.12 Tampilan Azure IAM pada ZTA-appgateway-DEV



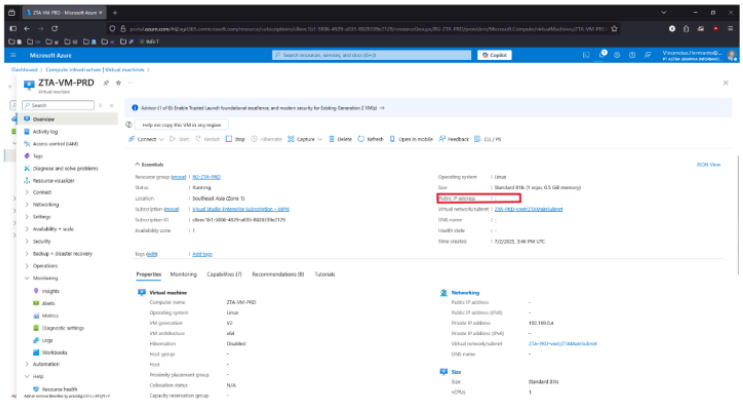
Gambar 4.13 Tampilan Azure IAM pada ZTA-VM-DEV

2. **Secure Communication**

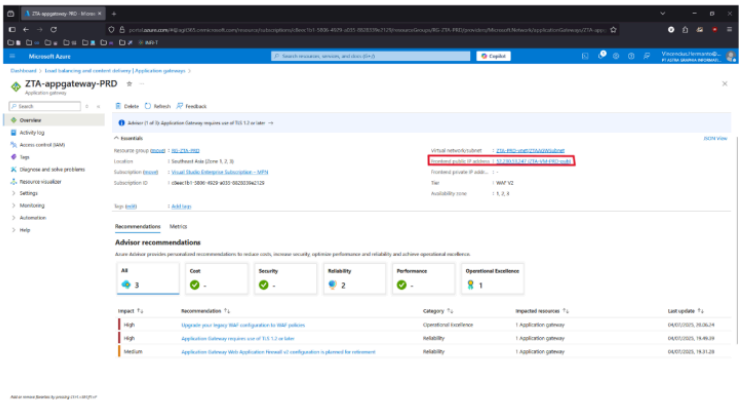
Prinsip ini memastikan bahwa komunikasi antar sistem dilakukan melalui jalur yang aman. Dalam *Project-Default*, *Virtual Machine* memiliki Public IP yang memungkinkan akses langsung melalui SSH selama pengguna memiliki private key. Ini menciptakan potensi celah keamanan, terutama dari sisi exposure ke internet publik. Sebaliknya, *Project-ZTA* menghilangkan penggunaan Public IP dan mengandalkan *Virtual Network Gateway* (VPN) untuk koneksi SSH, serta *Application Gateway* untuk akses ke web server. Dengan pendekatan ini, seluruh lalu lintas data harus melewati *gateway* yang dapat dikontrol dan diaudit, serta mendukung enkripsi data dalam perjalanan (*data in transit*), sehingga memenuhi prinsip komunikasi yang aman.



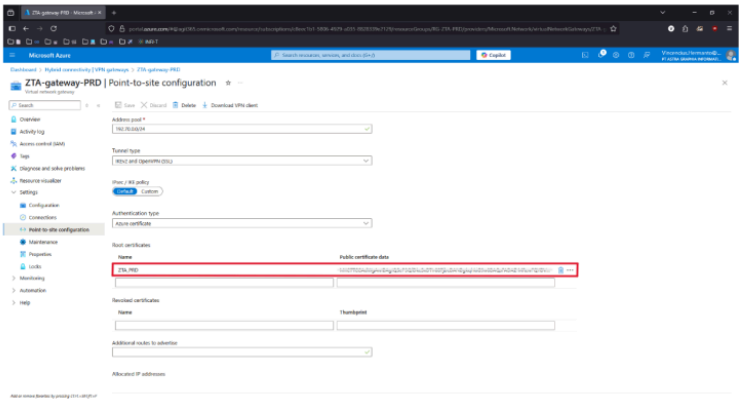
Gambar 4.14 Tampilan *Public* IP pada DEFAULT-VM-PRD



Gambar 4.15 Tampilan *Public* IP pada ZTA-VM-PRD



Gambar 4.16 Tampilan *Frontend Public* IP pada ZTA-appgateway-PRD

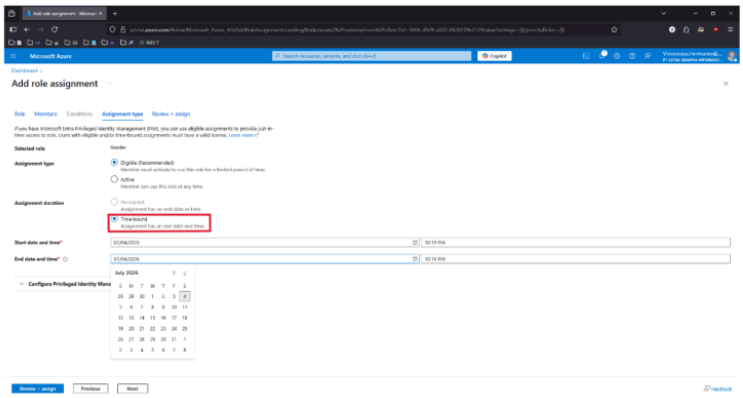


Gambar 4.17 Tampilan Konfigurasi VPN pada ZTA-gateway-PRD

3. **Access is Time-Limited**

Zero Trust mengharuskan akses bersifat sementara dan diberikan hanya saat diperlukan. Pada *Project-ZTA*, fitur *Azure Identity and Access Management (IAM)* dimanfaatkan untuk memberikan akses berbasis waktu kepada pengguna tertentu. *Administrator* dapat menentukan durasi akses

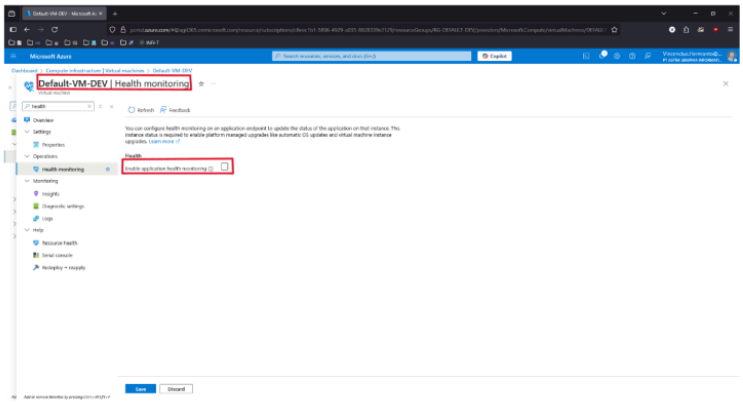
secara spesifik, dan sistem akan mencabut akses secara otomatis setelah waktu berakhir. Ini membantu mencegah kasus di mana pengguna yang telah menyelesaikan tugasnya masih memiliki hak akses aktif tanpa kontrol.



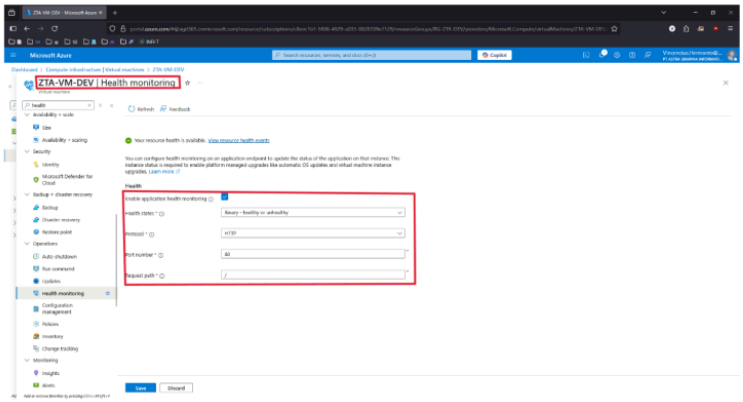
Gambar 4.18 Tampilan *Time-bound Access* pada Azure IAM

4. **Continuous System Monitoring**

Prinsip ini menekankan pentingnya pemantauan sistem secara berkelanjutan untuk mendeteksi anomali atau degradasi performa. Dalam *Project-Default*, tidak ada konfigurasi *health monitoring* yang aktif, sehingga tidak ada visibilitas langsung terhadap kondisi sistem. Sebaliknya, pada *Project-ZTA*, fitur Azure Monitor dan Resource Health diaktifkan untuk memantau status *web server* dan resource lainnya secara *real-time*. Monitoring ini memungkinkan *administrator* untuk mengidentifikasi masalah performa, ketidakstabilan layanan, atau potensi serangan lebih awal.



Gambar 4.19 Tampilan *Health Monitor* pada DEFAULT-VM-DEV

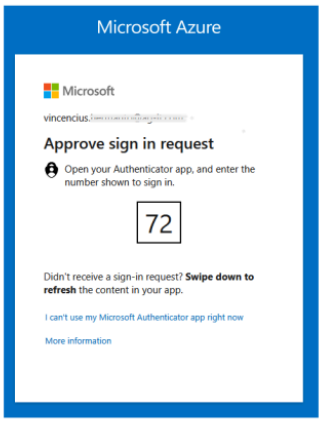


Gambar 4.20 Tampilan *Health Monitor* pada ZTA-VM-DEV

5. **Evaluated Access**

Pengujian pada *Project-ZTA* menunjukkan bahwa pemberian akses kepada pengguna dilakukan dengan mengevaluasi *scope* dan peran melalui Azure IAM. *Administrator* dapat melihat dan mengkategorikan akses berdasarkan *subscription*, *resource group*, maupun *resource* individual.

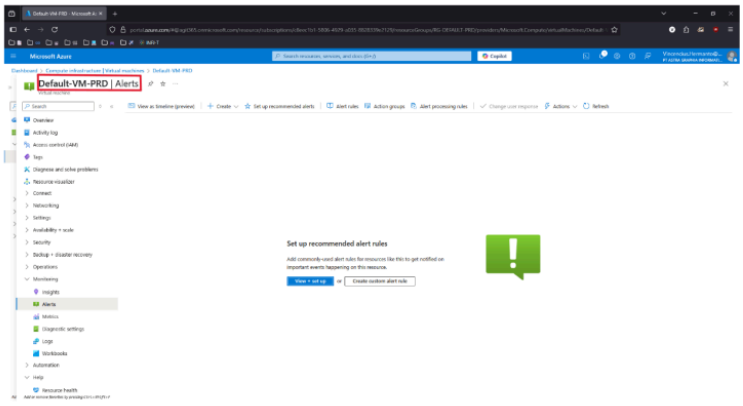
Evaluasi ini memungkinkan pemberian hak akses yang presisi dan menghindari pemberian hak akses berlebih. Praktik ini selaras dengan prinsip *Zero Trust* untuk memastikan bahwa setiap akses diperiksa terlebih dahulu dan tidak diberikan secara default.



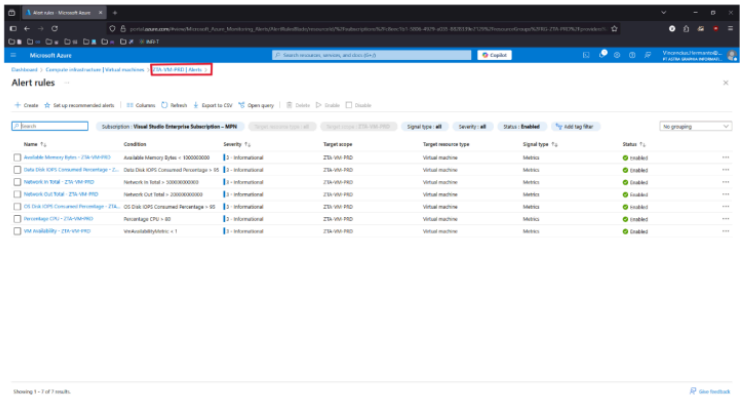
Gambar 4.21 Tampilan *Multi Factor Authentication* pada laman *Login*

6. **System Log Analysis**

Logging dan alerting merupakan elemen penting dalam *Zero Trust* untuk mendeteksi dan merespons kejadian keamanan. Pada *Project-Default*, tidak terdapat *alert rules* yang dikonfigurasi, sehingga setiap aktivitas atau insiden yang terjadi tidak akan memicu notifikasi otomatis. Di sisi lain, *Project-ZTA* menerapkan alert rules melalui Azure *Monitor* dan Azure *Activity Log*. Dengan adanya konfigurasi ini, sistem dapat mengirimkan peringatan secara real-time kepada administrator apabila terjadi aktivitas mencurigakan seperti percobaan akses yang gagal, eskalasi hak akses, atau perubahan konfigurasi kritis.



Gambar 4.22 Tampilan *Alert Rules* pada DEFAULT-VM-PRD



Gambar 4.23 Tampilan *Alert Rules* pada ZTA-VM-PRD

4.3 Perbandingan Implementasi ZTA dengan menggunakan panduan Azure dan ZTA pada penelitian ini

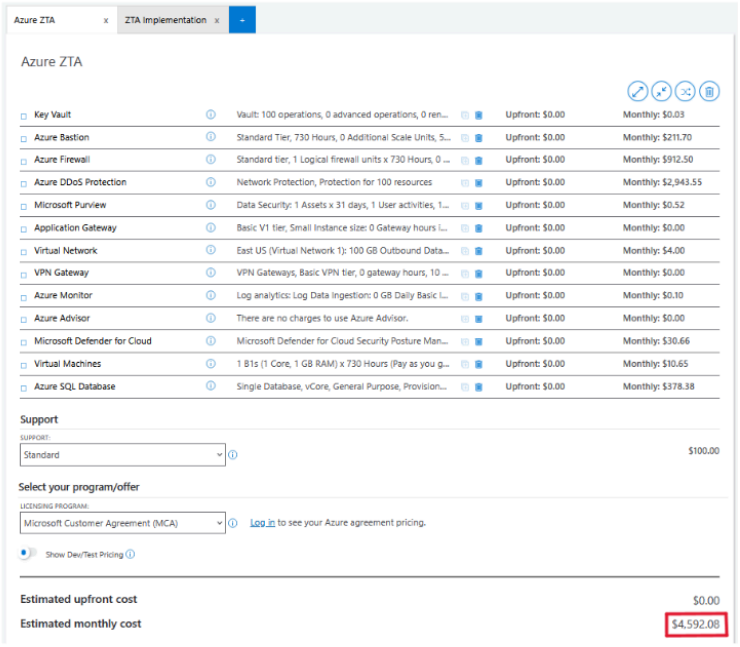
Setelah melakukan implementasi *Zero Trust Architecture* (ZTA) pada infrastruktur *cloud* menggunakan pendekatan yang dikembangkan dalam penelitian ini, langkah selanjutnya adalah membandingkannya dengan panduan resmi

implementasi *Zero Trust* pada Microsoft Azure. Perbandingan ini bertujuan untuk mengidentifikasi kesesuaian komponen yang digunakan, keefektifan pendekatan, serta efisiensi biaya dalam penerapan prinsip-prinsip *Zero Trust*. Microsoft Azure sendiri telah menyediakan panduan serta arsitektur referensi terkait penerapan ZTA yang mencakup berbagai layanan keamanan bawaan yang mendukung perlindungan berlapis terhadap data, identitas, dan jaringan.

Tabel 4.1 Perbandingan Penggunaan *Resource* dalam Implementasi ZTA

Azure *Guidance*	**Penelitian Ini**
Azure Key Vault	Application Gateway (WAF V2)
Azure Purview	Virtual Network Gateway
Azure Bastion	Azure Monitor
Application Gateway	Azure Advisor
Virtual Network Gateway	
Azure Firewall	
Azure DdoS Protection	
Azure Monitor	
Azure Advisor	

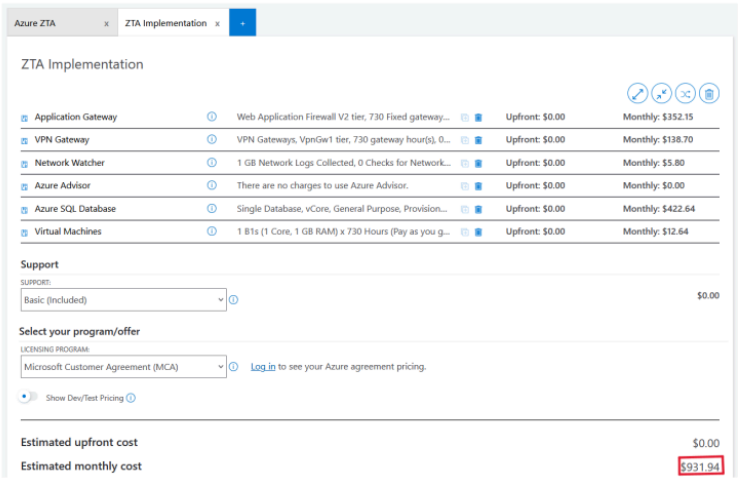
Secara umum, panduan *Zero Trust Architecture* yang disediakan oleh Microsoft Azure menggunakan rangkaian layanan keamanan yang lebih luas dan kompleks guna membangun perlindungan berlapis terhadap berbagai aspek sistem, mulai dari identitas, jaringan, hingga data. Komponen seperti Azure *Firewall* dan Azure DDoS *Protection* digunakan sebagai lapisan tambahan untuk memperkuat perimeter jaringan terhadap lalu lintas yang mencurigakan dan serangan berskala besar. Selain itu, Azure Bastion, Key *Vault*, dan Purview juga disertakan untuk memperkuat kontrol akses, perlindungan data sensitif, serta tata kelola informasi.



Gambar 4.24 Estimasi Harga Azure *Guidance* ZTA

Sebaliknya, implementasi *Zero Trust* dalam penelitian ini menggunakan pendekatan yang lebih sederhana dan terfokus, dengan mengandalkan *Application Gateway* (dengan WAF v2), VPN *Gateway*, Azure *Advisor*, dan *Health Monitoring* sebagai komponen inti. Penggunaan ***Application Gateway* dengan WAF v2** dipilih

karena mampu memberikan proteksi terhadap serangan aplikasi sekaligus menjalankan fungsi *load balancing*, sehingga **mengeliminasi kebutuhan akan Azure *Firewall* dan Azure DDoS *Protection*** dalam konteks infrastruktur ini. Beberapa layanan lain seperti Azure Bastion atau Purview tidak disertakan karena tidak secara langsung dibutuhkan dalam skenario pengujian atau lingkup pengamanan sistem yang dirancang.



Gambar 4.25 Estimasi Harga ZTA Penulis

Meskipun menggunakan jumlah resource yang lebih minimal dibandingkan dengan panduan resmi Azure, implementasi *Zero Trust* dalam penelitian ini **tetap memenuhi ketujuh prinsip utama *Zero Trust Architecture***, sebagaimana telah dibuktikan melalui serangkaian pengujian pada infrastruktur yang dibangun. Pendekatan ini juga **lebih *cost-efficient*** dan memungkinkan penerapan *Zero Trust* secara bertahap bagi organisasi yang memiliki keterbatasan anggaran namun tetap ingin membangun sistem cloud yang aman dan terukur.

BAB 5

KESIMPULAN

5.1 Kesimpulan

Berdasarkan penelitian dan pengujian yang telah dilakukan, dapat disimpulkan bahwa seluruh prinsip dalam *Zero Trust Architecture* (ZTA) berhasil diterapkan secara efektif pada lingkungan cloud menggunakan Microsoft Azure. Ketujuh prinsip *Zero Trust* yang dirumuskan oleh NIST—mulai dari perlindungan terhadap resource, komunikasi aman, akses terbatas waktu, kebijakan akses dinamis, pemantauan berkelanjutan, evaluasi akses, hingga analisis log sistem—telah diuji dan dibuktikan melalui implementasi nyata pada dua infrastruktur *cloud* yang dibangun, yaitu *Project-Default* dan *Project-ZTA*. Infrastruktur *Project-ZTA* menunjukkan bahwa prinsip-prinsip tersebut dapat terpenuhi melalui kombinasi layanan seperti Azure *Application Gateway* (WAF v2), VPN *Gateway*, Azure IAM, Azure *Advisor*, dan fitur *monitoring* yang aktif.

Penerapan *Zero Trust* pada *Project-ZTA* juga terbukti mampu menjawab permasalahan yang dipaparkan dalam studi kasus di awal penelitian. Studi kasus pertama mengenai ***unauthorized access* di waktu yang tidak semestinya** berhasil diminimalkan dengan penerapan akses berbatas waktu dan autentikasi berlapis, yang membatasi waktu serta konteks akses pengguna terhadap resource. Sedangkan studi kasus kedua terkait **evaluasi akses yang tidak dilakukan dan menyebabkan eks-karyawan menyalahgunakan hak akses**, dapat diantisipasi melalui fitur *evaluated access* dan *Role Based Action Control* (RBAC) di Azure IAM, yang memungkinkan *administrator* untuk memantau, membatasi, dan mencabut akses secara selektif dan terkontrol.

Selain itu, hasil perbandingan antara implementasi ZTA berdasarkan panduan resmi Microsoft Azure dengan implementasi ZTA dalam penelitian ini menunjukkan bahwa meskipun panduan Azure menawarkan cakupan perlindungan yang lebih komprehensif melalui penggunaan layanan tambahan seperti Azure

Firewall, Bastion, DDoS *Protection*, Key Vault, dan Purview, pendekatan tersebut memiliki biaya operasional yang relatif lebih tinggi. Di sisi lain, penelitian ini mengusulkan pendekatan ZTA yang lebih sederhana dan *cost-efficient* dengan tetap menjaga efektivitas dan kepatuhan terhadap prinsip-prinsip *Zero Trust*. Hal ini menjadikan pendekatan yang digunakan dalam penelitian ini sebagai alternatif yang layak untuk organisasi berskala kecil hingga menengah yang ingin mulai menerapkan *Zero Trust* pada sistem *cloud*-nya tanpa beban biaya yang besar.

5.2 Saran

Berdasarkan hasil implementasi *zero trust architecture* yang telah berhasil dilakukan, ada beberapa hal yang dapat dikembangkan kedepannya:

- Penggunaan *Azure Key Vault* sebagai metode penyimpanan *key* bersifat sensitif yang digunakan pada komponen dan layanan yang ada,
- Penggunaan *Recovery Service Vault* untuk keperluan *backup* untuk komponen dan layanan yang ada,
- Penggunaan *Log Analytics Workspace* sebagai tempat melakukan analisa informasi yang ada pada *Log* yang dimiliki,
- Pada sistem dengan skala yang lebih besar dan memiliki komponen dan layanan yang lebih kompleks dan banyak, dapat digunakan lebih banyak *application gateway* sebagai *load balancer* untuk meningkatkan *availability* dari sistem yang akan banyak diakses.

Thesis_ZTA

ORIGINALITY REPORT

7%	5%	2%	4%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to National University College - Online Student Paper	1%
2	docplayer.info Internet Source	1%
3	techcommunity.microsoft.com Internet Source	1%
4	www.coursehero.com Internet Source	<1%
5	repository.its.ac.id Internet Source	<1%
6	eprints.umpo.ac.id Internet Source	<1%
7	123dok.com Internet Source	<1%
8	Submitted to Purdue University Student Paper	<1%
9	docs.chef.io Internet Source	<1%
10	text-id.123dok.com Internet Source	<1%
11	andonesia.com Internet Source	<1%

12	repository.binadarma.ac.id Internet Source	<1 %
13	www.elmark.com.pl Internet Source	<1 %
14	www.lintasarta.net Internet Source	<1 %
15	etheses.uin-malang.ac.id Internet Source	<1 %
16	repository.teknokrat.ac.id Internet Source	<1 %
17	espjeta.org Internet Source	<1 %
18	Submitted to Universitas Dian Nuswantoro Student Paper	<1 %
19	Submitted to University of Bolton Student Paper	<1 %
20	Submitted to University of Portsmouth Student Paper	<1 %
21	scholar.unand.ac.id Internet Source	<1 %
22	support.exabytes.co.id Internet Source	<1 %
23	Submitted to Noroff University College Student Paper	<1 %
24	Submitted to Police Academy – University of Police Science Student Paper	<1 %
25	library.binus.ac.id	

	Internet Source	<1%
26	digilibadmin.unismuh.ac.id Internet Source	<1%
27	ejournal.uby.ac.id Internet Source	<1%
28	journal.untar.ac.id Internet Source	<1%
29	Shijimol Ambi Karthikeyan. "Practical Microsoft Azure IaaS", Springer Science and Business Media LLC, 2018 Publication	<1%
30	wjaets.com Internet Source	<1%
31	www.webagesolutions.com Internet Source	<1%
32	Saleh Dwiyatno, Muhamad Rosi Sadi, Muhamad Natsir. "RANCANG BANGUN DAN MONITORING IP CAMERA B ERBASIS OPEN-WRT PADA KANTOR PDAM TIRTA BERKAH PANDEGLANG", Jurnal Sistem Informasi dan Informatika (Simika), 2019 Publication	<1%
33	blog.ibukasti.com Internet Source	<1%
34	digilib.uinsby.ac.id Internet Source	<1%
35	docs.azure.cn Internet Source	<1%

36	id.scribd.com Internet Source	<1%
37	journal.ipb.ac.id Internet Source	<1%
38	qdoc.tips Internet Source	<1%
39	tech.glosarium.org Internet Source	<1%
40	www.dri.co.jp Internet Source	<1%
41	Puthiyavan Udayakumar. "Design and Deploy a Secure Azure Environment", Springer Science and Business Media LLC, 2023 Publication	<1%

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	On		