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*:

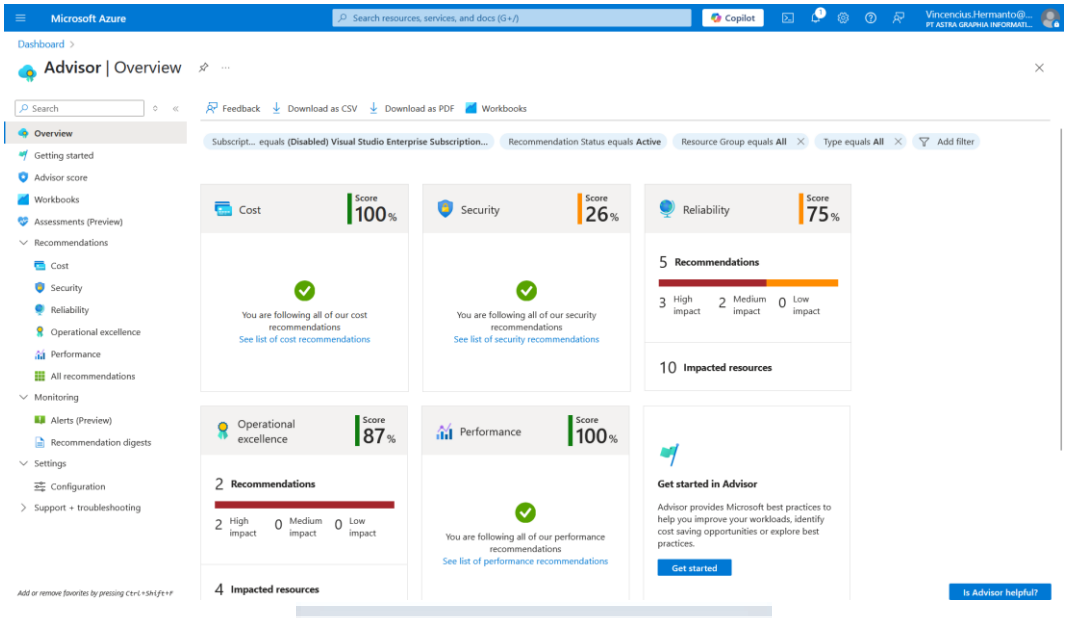
4.1.1 *Azure Advisor*

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*).

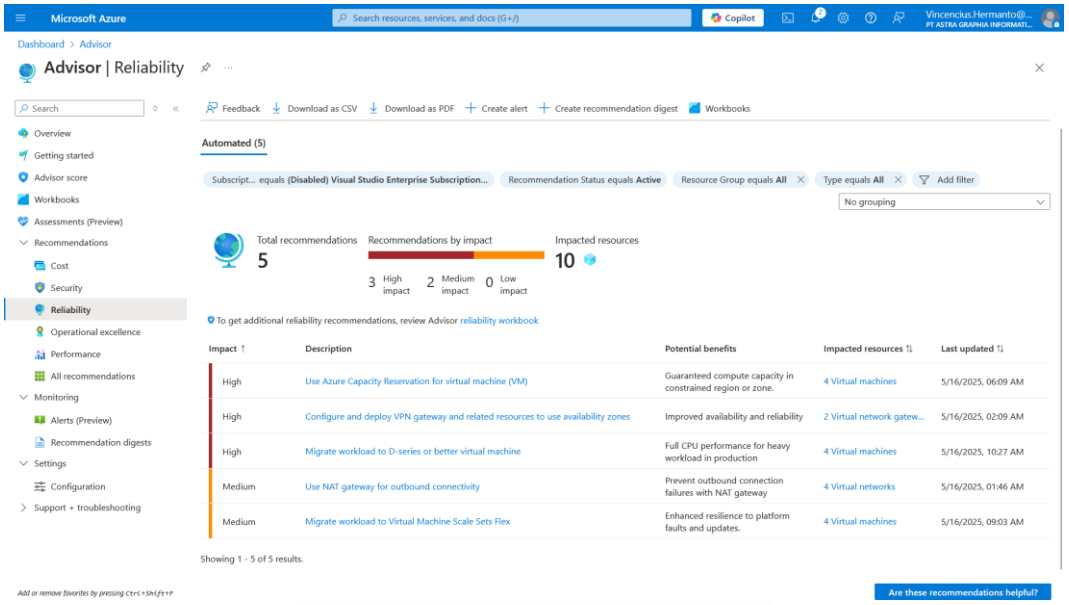
Advisor sendiri akan melakukan *auto-scan* setiap interval 24 jam dan memberikan laporan terhadap temuan yang ada kedalam masing- masing kategori tersebut.

- 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.



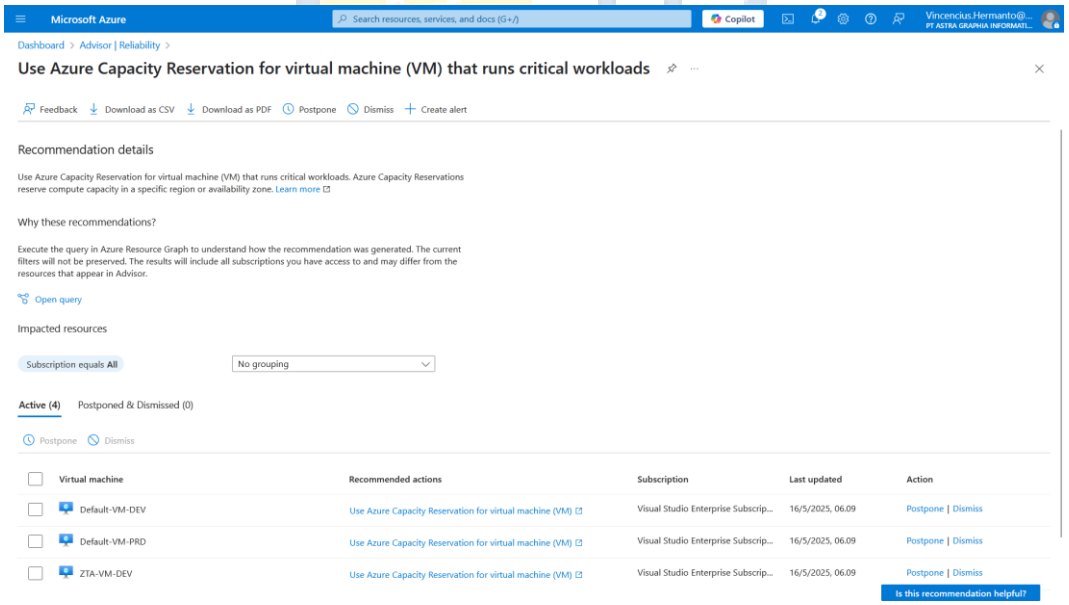
Gambar 4.1 Tampilan *Dashboard Azure Advisor*

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.

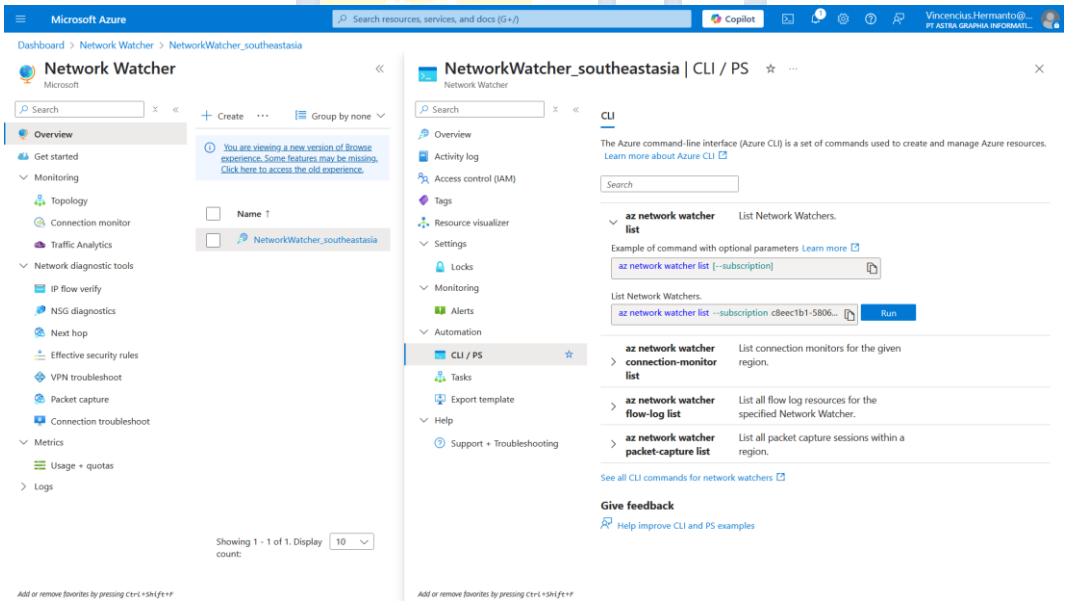


Gambar 4.3 Tampilan *Issue* pada *Advisor*

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*.

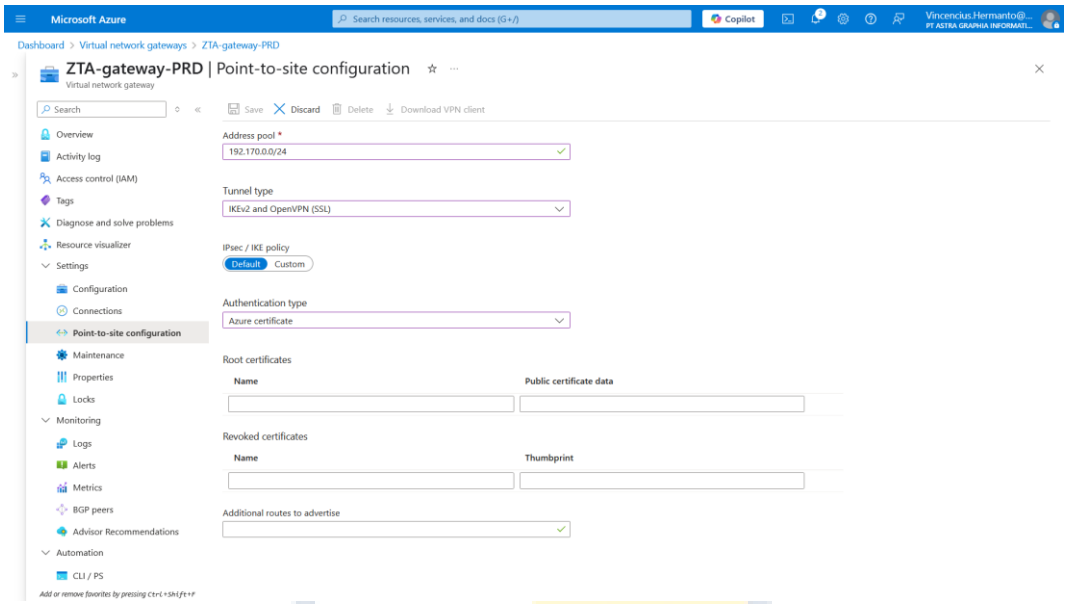
4.1.3 *Azure Virtual Network Gateway*

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.

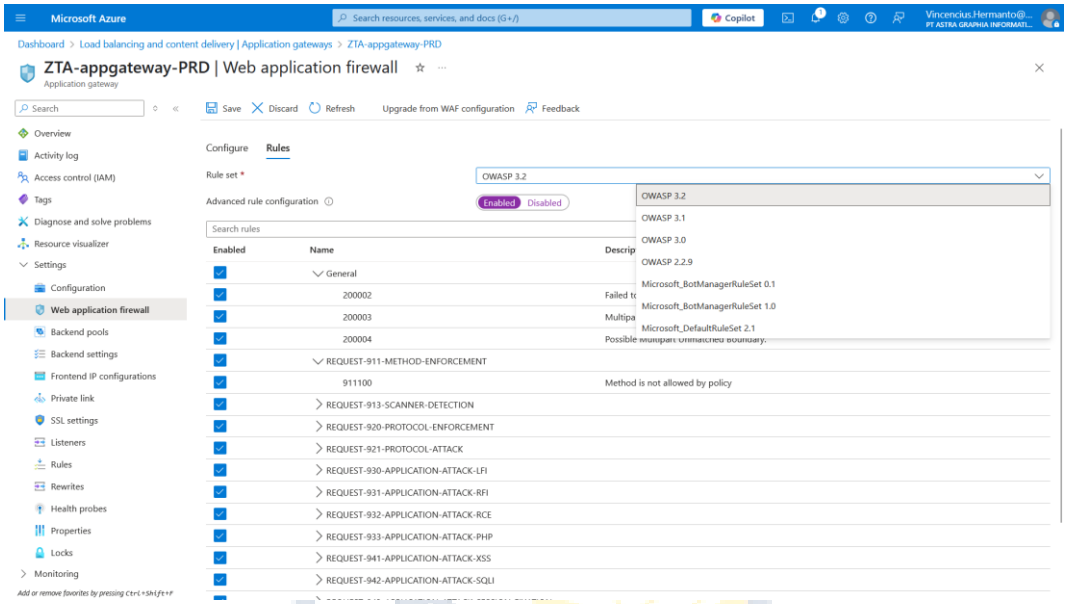
Dengan menggunakan VPN *point-to-site*, jaringan *private* yang berada pada *cloud environment* dapat diakses melalui perangkat eksternal melalui VPN *tunneling*. Hal ini memberikan tingkat keamanan yang lebih baik dibanding dengan membuka akses terhadap layanan atau komponen yang ada, dalam hal ini 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.

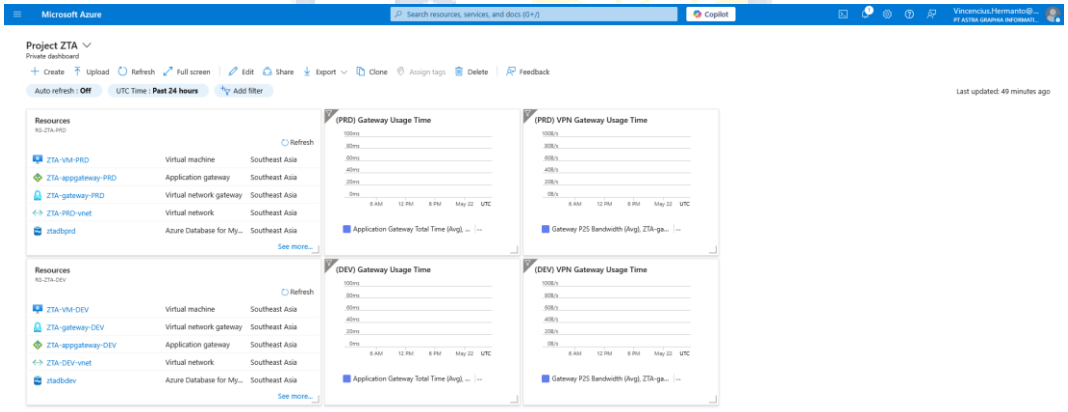
4.1.5 *Azure Resource Health and Monitoring*

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.

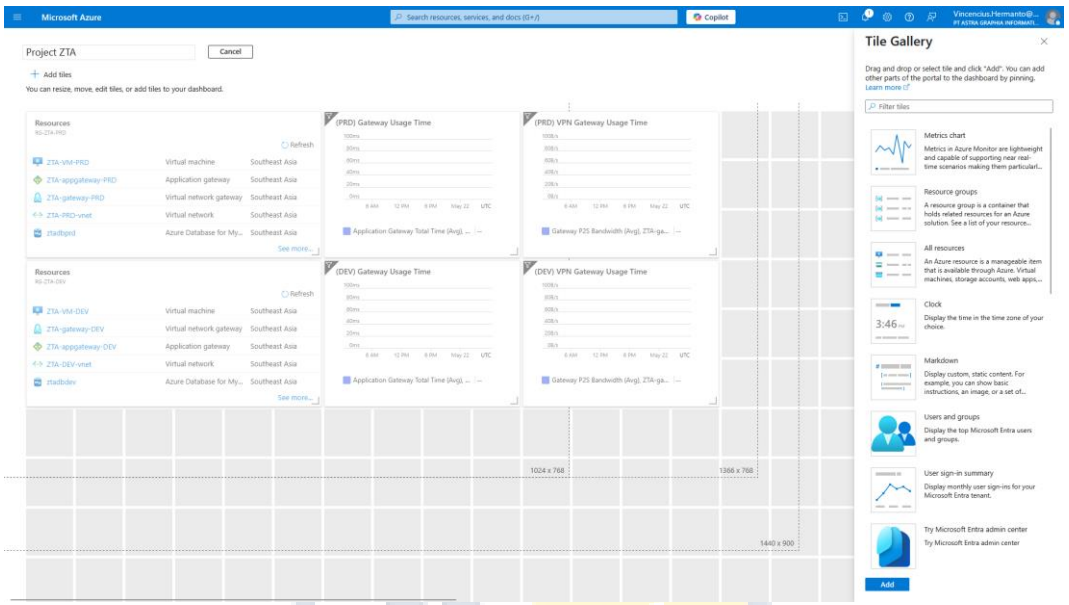
4.1.6 *Azure Dashboard*



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.



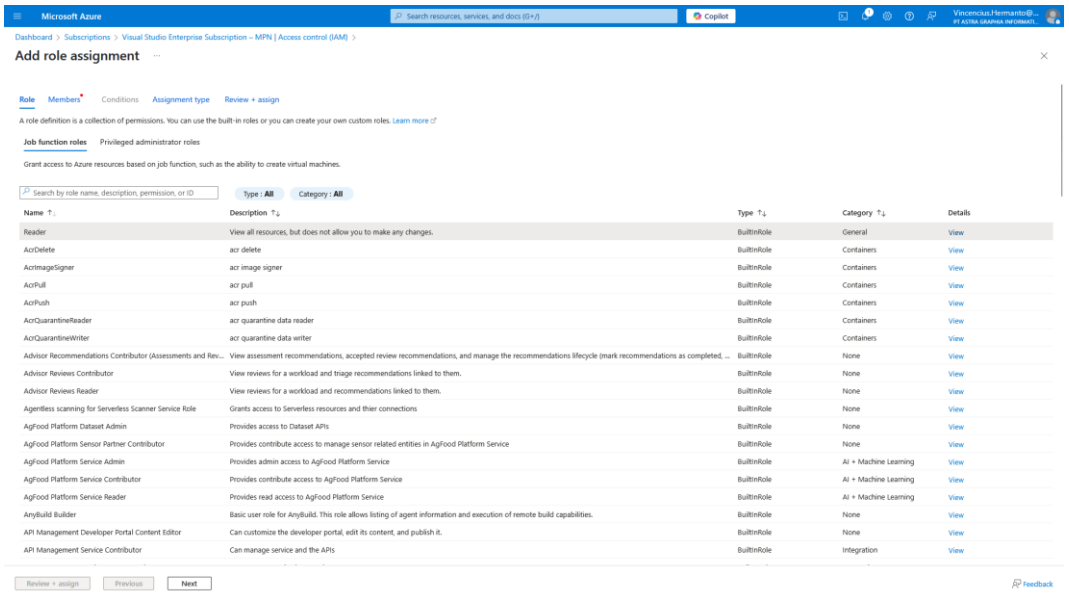
Gambar 4.8 Tampilan modifikasi *Azure Dashboard*

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)

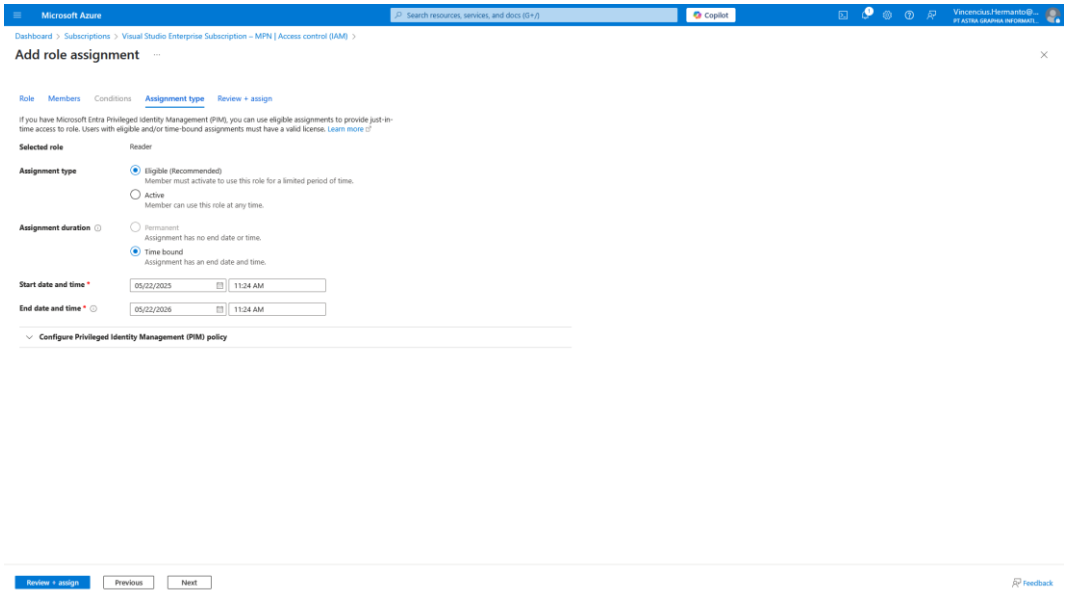
Azure identity access management (IAM) merupakan pusat kontrol atas hak akses yang diberikan kepada pengguna dalam mengelola komponen maupun layanan yang ada pada *Azure*. Dengan ini *Azure* IAM akan memenuhi tiga prinsip terakhir yang belum dapat terpenuhi oleh komponen maupun layanan lain yang telah 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.



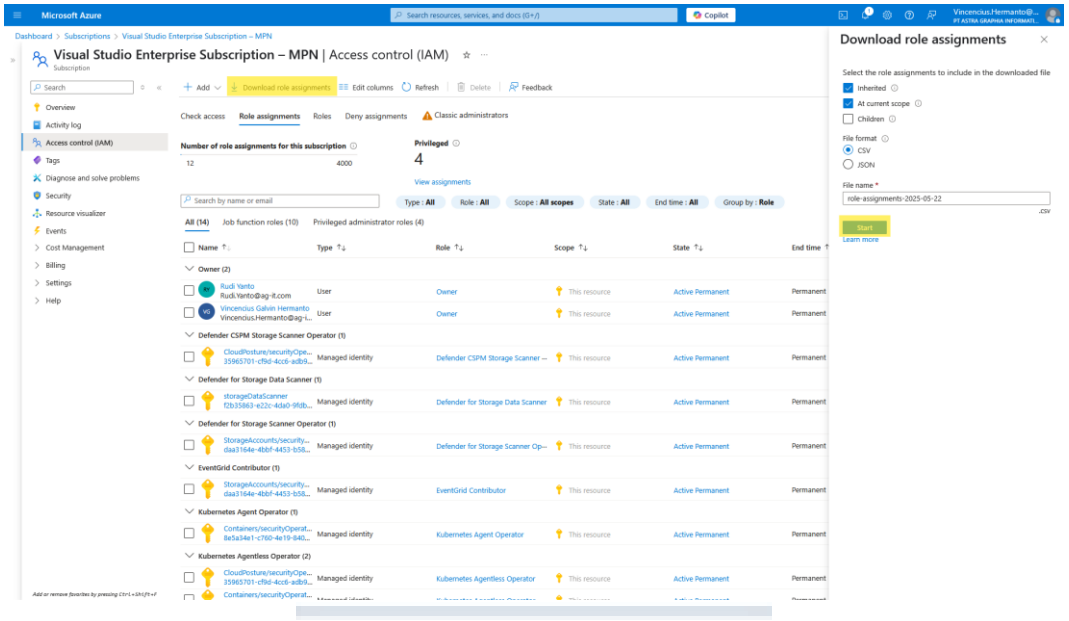
Gambar 4.9 Tampilan *Role Assignment* pada *Azure* IAM

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.

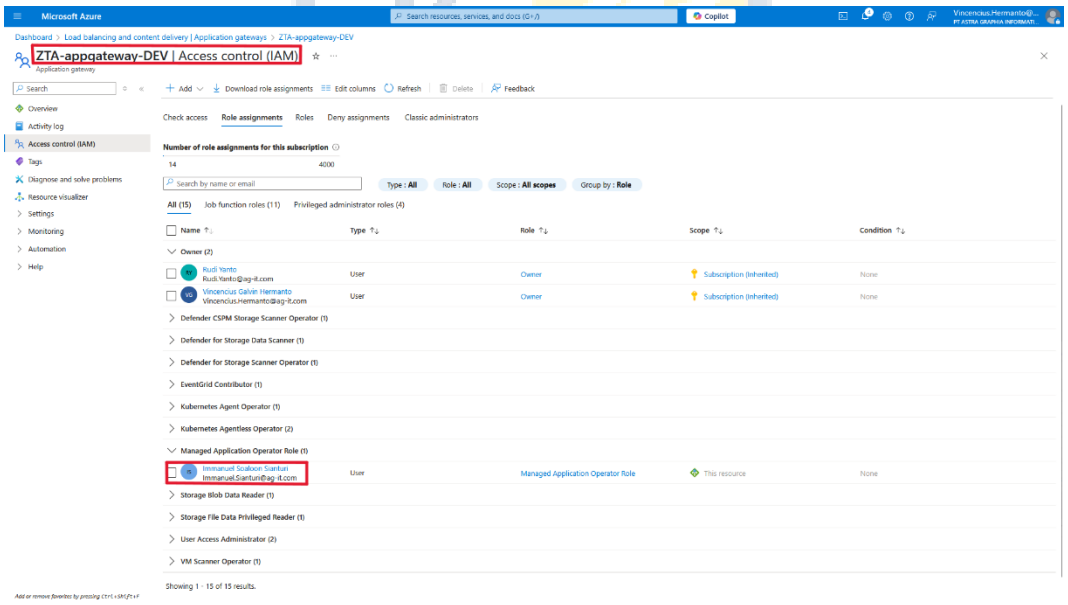
4.2 *Testing* Infrastruktur atas 7 Prinsip *Zero Trust*

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.

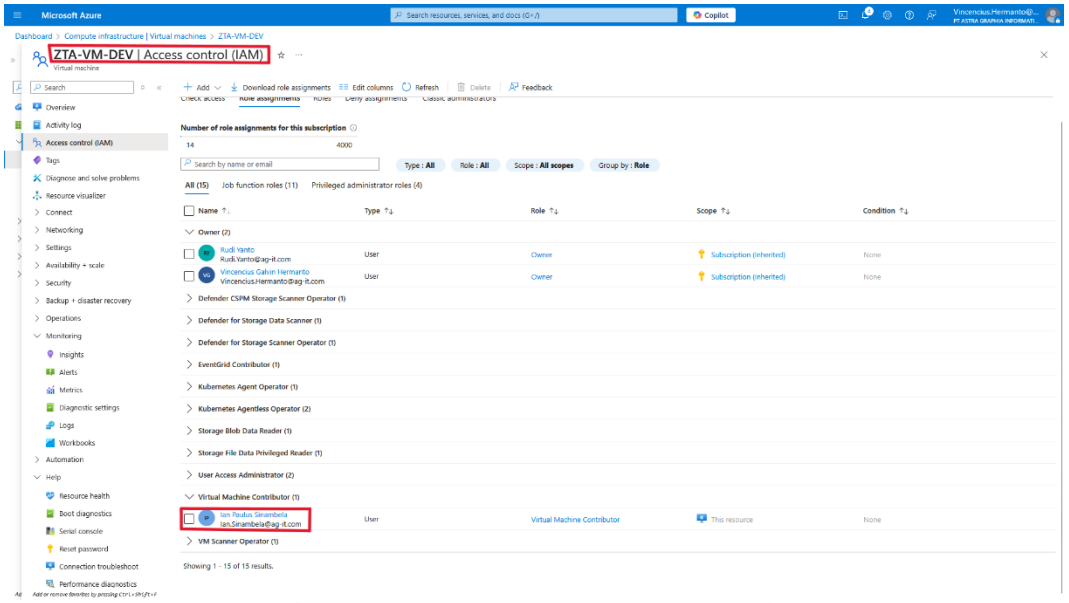
Berikut merupakan hasil pengujian dan penjelasan lebih lanjut terkait setiap prinsip ZTA:

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.



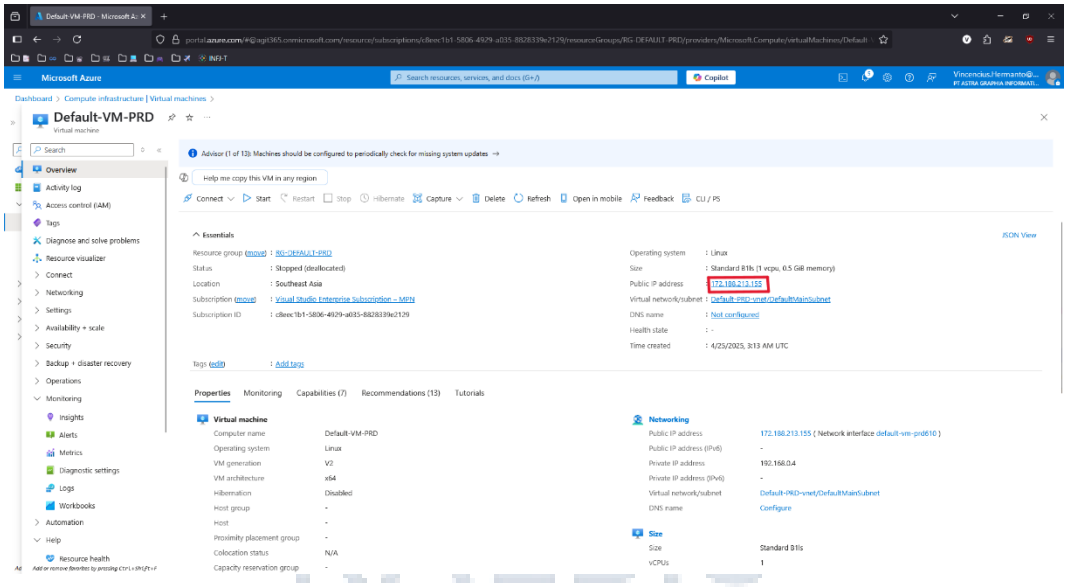
Gambar 4.12 Tampilan Azure IAM pada ZTA-appgateway-DEV



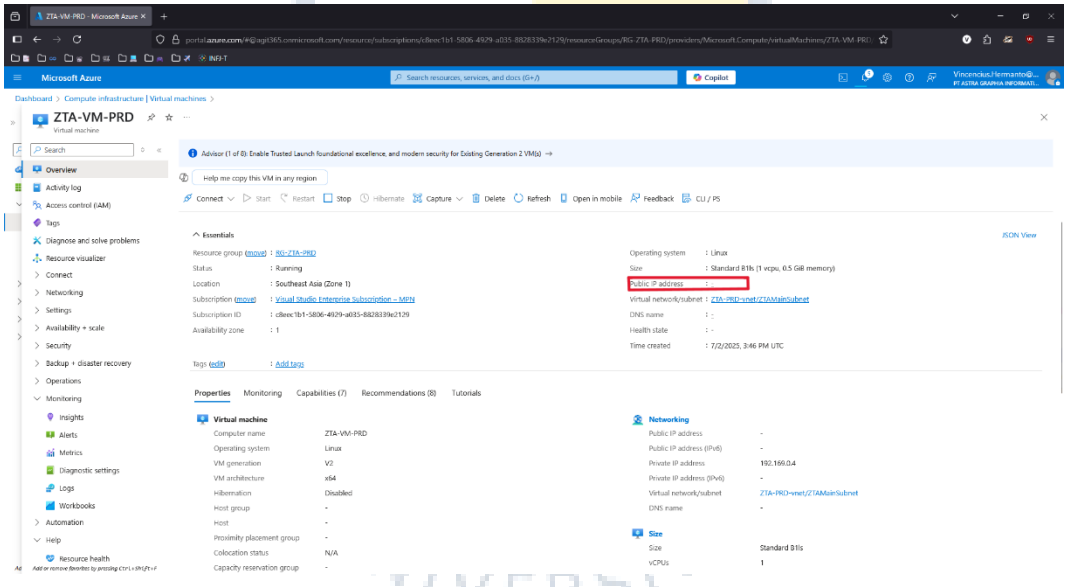
Gambar 4.13 Tampilan Azure IAM pada ZTA-VM-DEV

2. **Secure Communication**

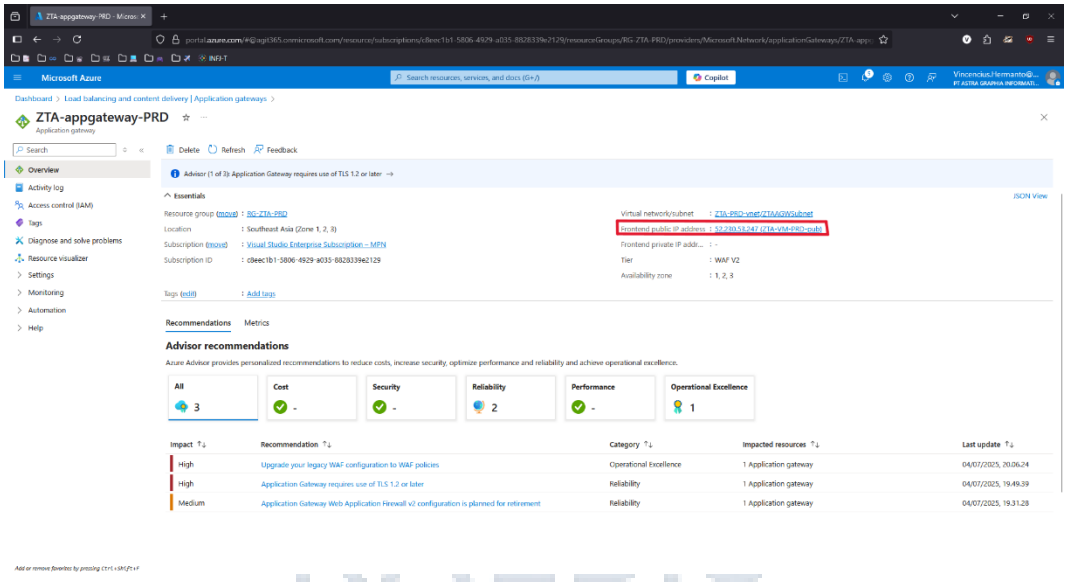
Prinsip ini memastikan bahwa komunikasi antar sistem dilakukan melalui jalur yang aman. Dalam *Project-Default*, *Virtual Machine* memiliki Public IP yang memungkinkan akses langsung melalui SSH selama pengguna memiliki private key. Ini menciptakan potensi celah keamanan, terutama dari sisi exposure 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.



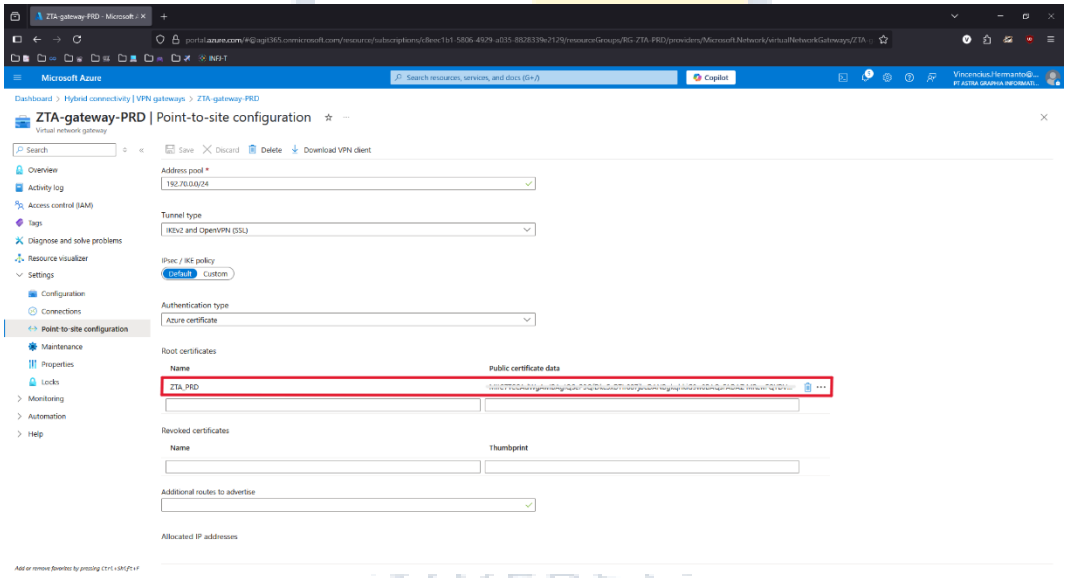
Gambar 4.14 Tampilan *Public* IP pada DEFAULT-VM-PRD



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



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

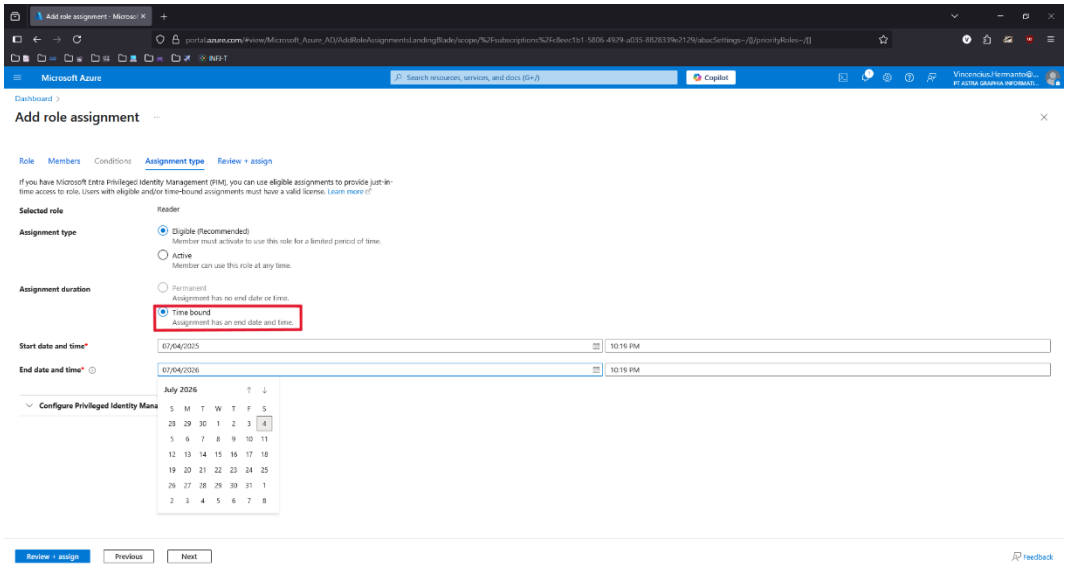


Gambar 4.17 Tampilan Konfigurasi VPN pada ZTA-gateway-PRD

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

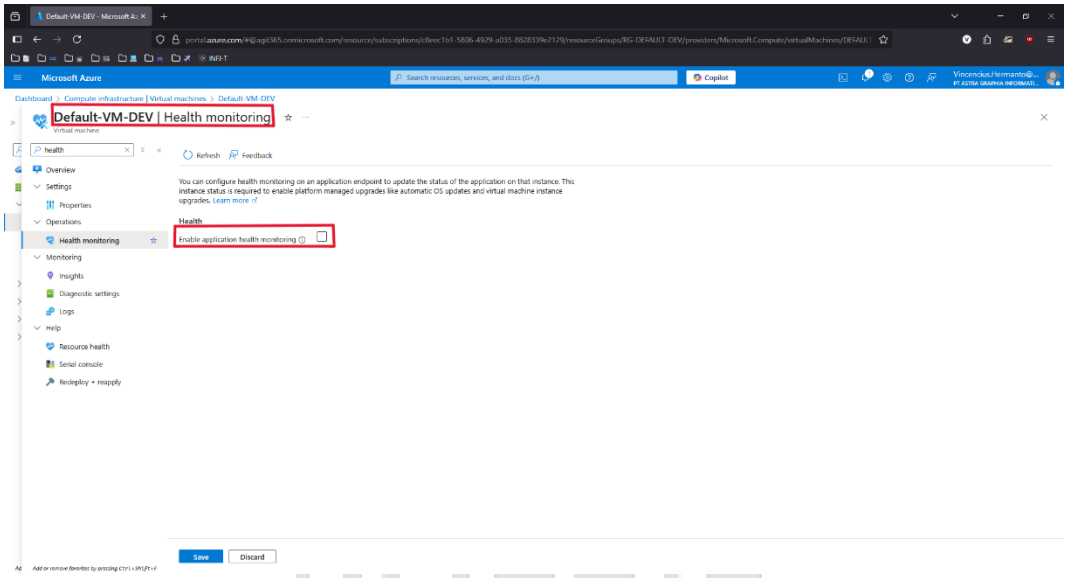
secara spesifik, dan sistem akan mencabut akses secara otomatis setelah waktu berakhir. Ini membantu mencegah kasus di mana pengguna yang telah menyelesaikan tugasnya masih memiliki hak akses aktif tanpa kontrol.



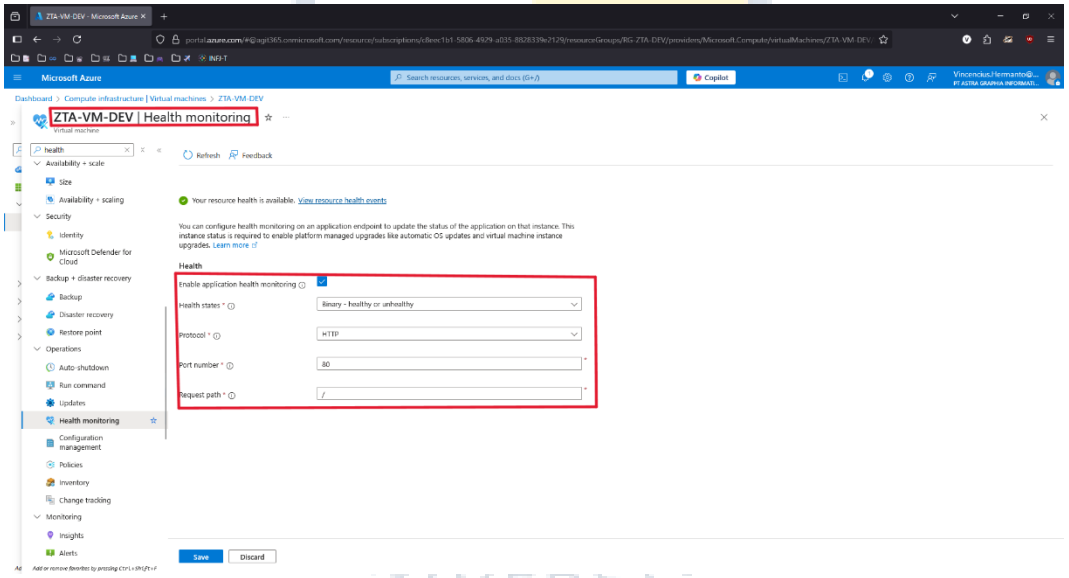
Gambar 4.18 Tampilan *Time-bound Access* pada Azure IAM

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

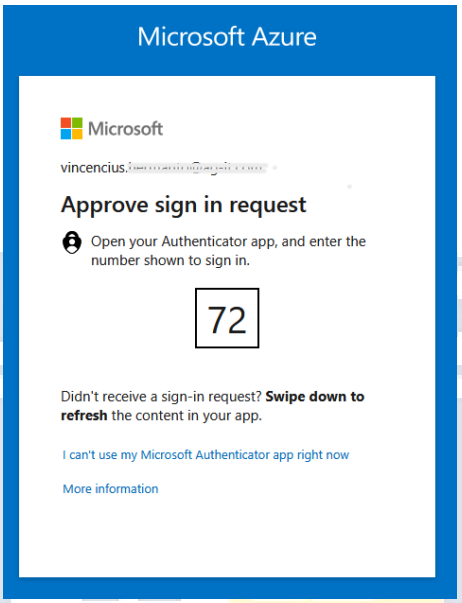


Gambar 4.20 Tampilan *Health Monitor* pada ZTA-VM-DEV

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.

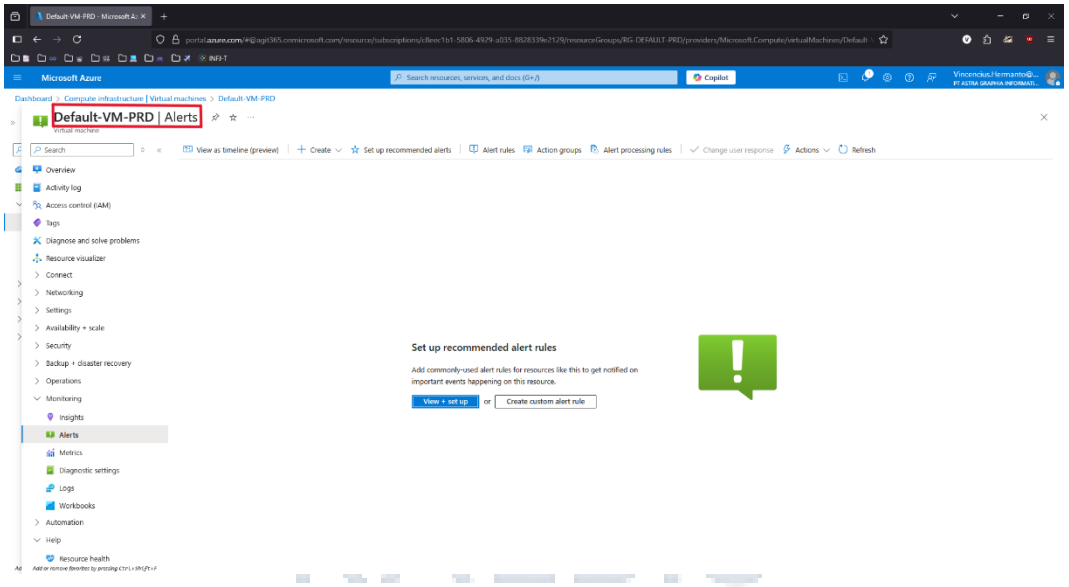
Evaluasi ini memungkinkan pemberian hak akses yang presisi dan menghindari pemberian hak akses berlebih. Praktik ini selaras dengan prinsip *Zero Trust* untuk memastikan bahwa setiap akses diperiksa terlebih dahulu dan tidak diberikan secara default.



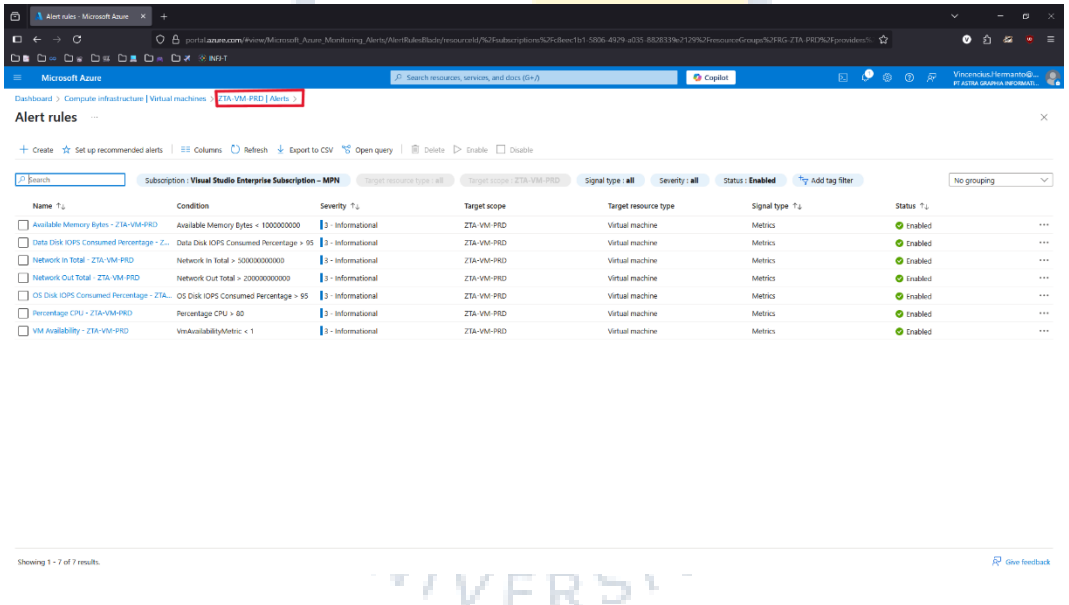
Gambar 4.21 Tampilan *Multi Factor Authentication* pada laman *Login*

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

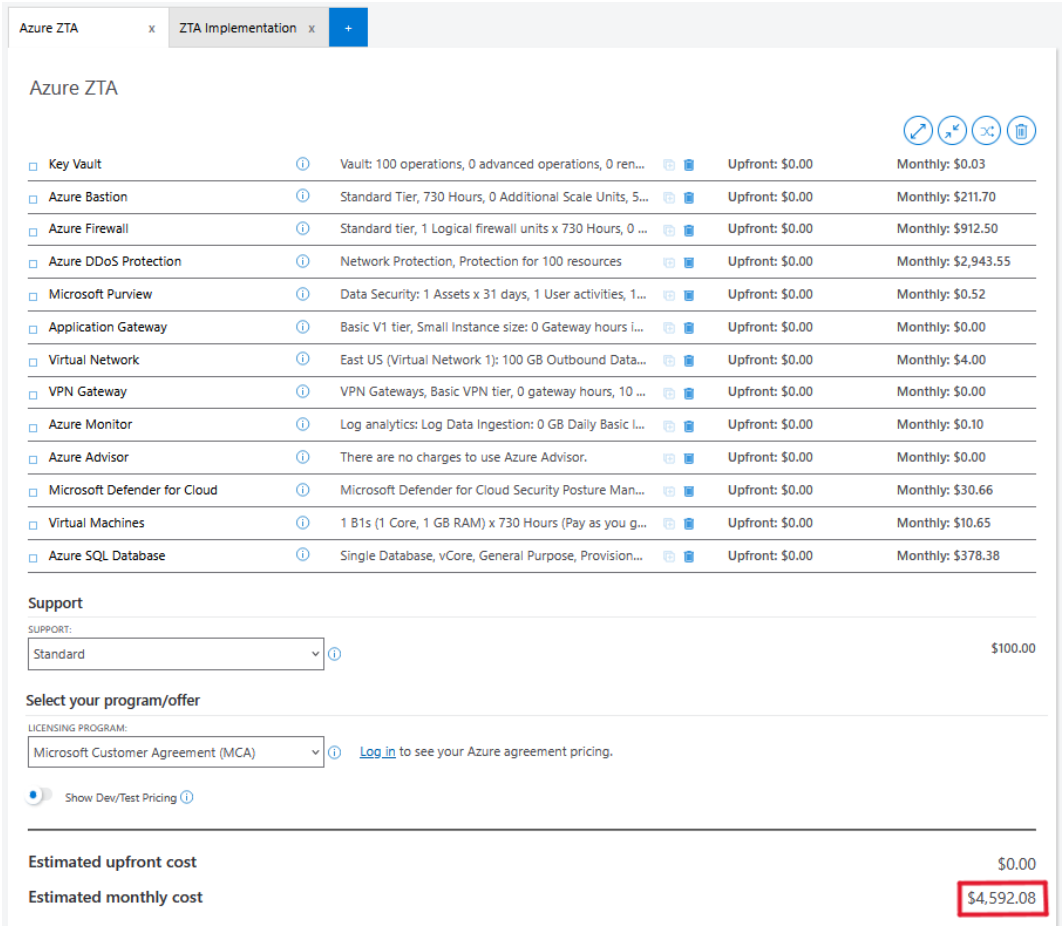
Setelah melakukan implementasi *Zero Trust Architecture* (ZTA) pada infrastruktur *cloud* menggunakan pendekatan yang dikembangkan dalam penelitian ini, langkah selanjutnya adalah membandingkannya dengan panduan resmi

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	

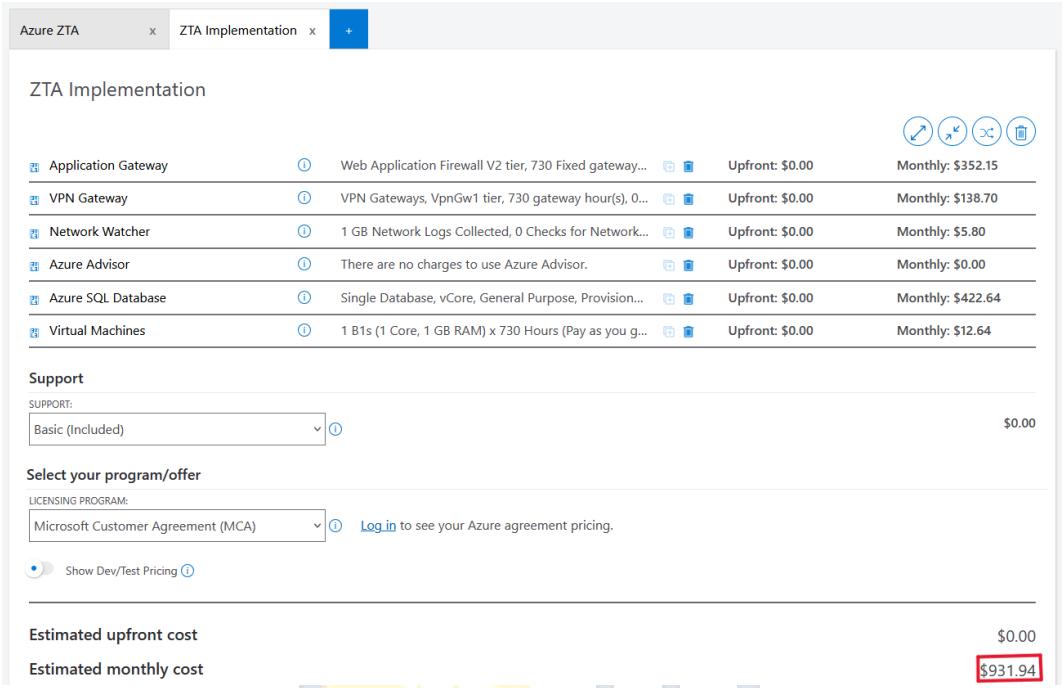
Secara umum, panduan *Zero Trust Architecture* yang disediakan oleh Microsoft Azure menggunakan rangkaian layanan keamanan yang lebih luas dan kompleks guna membangun perlindungan berlapis terhadap berbagai aspek sistem, mulai dari identitas, jaringan, hingga data. Komponen seperti Azure *Firewall* dan 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.