

# BIZNET QUERY LANGUAGE PADA INFRASTRUCTURE AS CODE

Eka Tresna Irawan, Dondy Bappedyanto, Dedy Hariyadi, Azzam Syawqi  
Aziz

PT Biznet Gio Nusantara  
Universitas Jenderal Acmad Yani Yogyakarta  
Universitas Brawijaya

ibnu@biznetgio.com, dondy@biznetgio.com, milisdad@gmail.com,  
azzamsa@student.ub.ac.id

## Abstrak

*Infrastructure as Code berdasarkan TOSCA yang dimplementasikan pada perusahaan penyediaan layanan komputasi awan (Amazon Web Services, Azure, Google Cloud Platform, Alibaba Cloud) menggunakan format data JSON atau YAML. Format data JSON dan YAML hanya mendukung proses CREATE dan UPDATE dalam pengelolaan sumber daya infrastruktur komputasi awan. Proses CREATE dan UPDATE dirasa masih kurang mendukung manajemen sumber daya infrastruktur komputasi awan. Pada penelitian ini diusulkan penyempurnaan proses CREATE, READ, UPDATE dan DELETE pada Infrastructure as Code. Jadi pada Infrastructure as Code tidak hanya mendukung Markup Language namun juga mendukung Query Language untuk mengelola sumber daya infrastruktur komputasi awan.*

**Kata Kunci:** JSON, YAML, komputasi awan, Infrastructure as Code, AST, DevOps, BiznetQL, SQL AST, TOSCA, CRUD, REPL

## 1. Pendahuluan

DevOps merupakan pendekatan pengembangan perangkat lunak yang memiliki tujuan untuk mengurangi waktu dalam proses pengembangan termasuk ketika proses pengembangan memasuki fase produksi. Manfaat lain dari DevOps adalah membantu pengembang dalam menjaga kualitas proses pengembangan. Dalam DevOps terdapat lima kategori proses di antaranya (Bass, Weber, dan Zhu, 2015):

- Divisi yang menangani infrastruktur (operasional) dijadikan dalam satu bagian dengan divisi pengembangan;
- Penguatan divisi pengembangan dalam penanganan kendala;
- Melakukan penyeragaman lingkungan pengembangan antara divisi pengembangan dan infrastruktur;
- Menerapkan pengembangan berkelanjutan;
- Melakukan pengembangan kode yang sama antar proses.

Berdasarkan lima kategori tersebut maka dapat dibangun sebuah ekosistem yang meliputi *Version Control System (VCS)*, *Test-Driven Development (TDD)*, *Continuous Intergration (CI)*, *Continuous Delivery (CD)*, dan

*Continuous Deployment*. Pada ekosistem ini maka diperlukan otomatisasi infrastruktur dengan pendekatan pengembangan perangkat lunak yang disebut *Infrastructure as Code* (IaC) (Morris, 2016). Melalui *Infrastructure as Code* maka dapat dilakukan konfigurasi dan otomatisasi infrastruktur yang terpusat melalui sebuah kode. Oleh sebab itu *Infrastructure as Code* dapat dikatakan sebagai dasar dari implementasi DevOps untuk mempercepat pengembangan perangkat lunak dan memberikan layanan lainnya kepada pengguna (Rahman, Mahdavi-Hezaveh, dan Williams, 2018).

Saat ini, penyedia layanan teknologi komputasi awan yang mendukung *Infrastructure as Code* diantaranya adalah Amazon Web Services, Azure, Google Cloud Platform, dan Alibaba Cloud. AWS CloudFormation yang diluncurkan pada tahun 2011 merupakan *Infrastructure as Code* untuk mengendalikan infrastuktur pada Amazon secara mudah selayaknya menulis kode (Amazon Web Services, 2016). AWS CloudFormation ini juga dapat diterapkan pada lingkungan komputasi awan berbasis OpenStack. Teknologi OpenStack yang serupa dan kompatibel dengan AWS CloudFormation adalah OpenStack Heat (Radez, 2015).

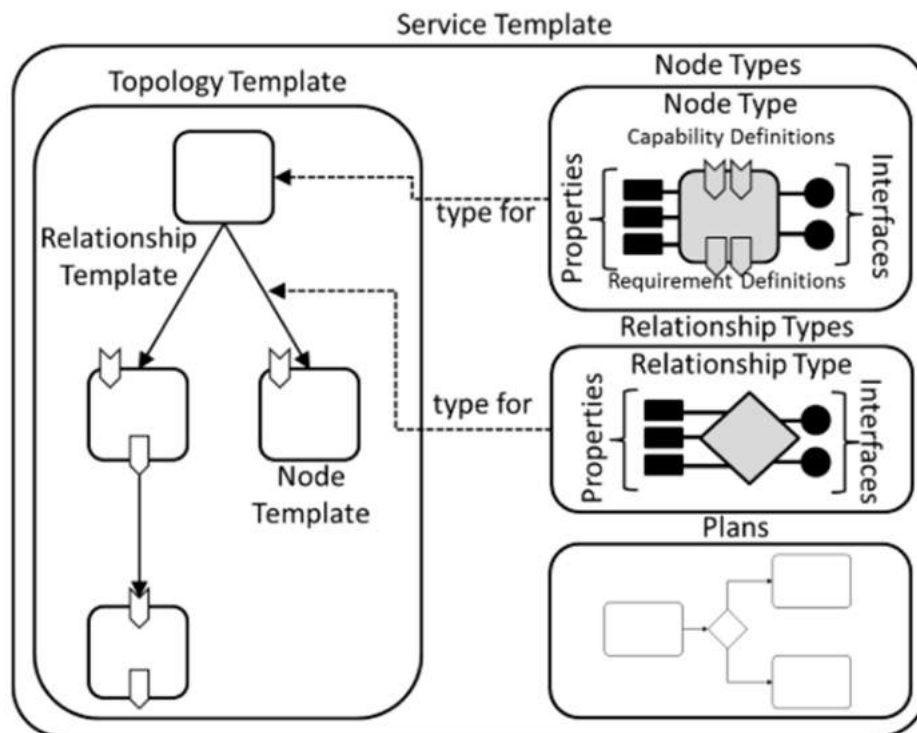
## **2. Konsep Dasar *Infrastructure as Code***

### **2.1 TOSCA (*Topology and Orchestration Specification for Cloud Applications*)**

OASIS adalah konsorsium nirlaba yang mendorong pengembangan, konvergensi dan adopsi standar terbuka untuk masyarakat informasi global. OASIS mempromosikan konsensus industri dan menghasilkan standar dunia untuk keamanan, *Internet of Things*, komputasi awan, energi, teknologi konten, manajemen darurat, dan lain-lain. Standar terbuka OASIS menawarkan potensi untuk menurunkan biaya, mendorong inovasi, menumbuhkan pasar global, dan melindungi hak atas pilihan bebas terhadap teknologi ("OASIS," 2018).

Komputasi awan akan mendapatkan nilai jika mendukung proses semi-otomatis dan memberikan pengelolaan layanan saling terhubung dengan sistem komputasi awan lainnya sehingga dapat berjalan dengan baik. OASIS mengeluarkan sebuah rekomendasi yang disebut *TOSCA (Topology and Orchestration Specification for Cloud Applications)* dalam bentuk penyediaan bahasa untuk mendeskripsikan komponen layanan, menghubungkan melalui *service topology* dan pengelolaan prosedur melalui proses *orchestration* (Lipton, Moser, Palma, dan Spatzier, 2013).

Konsep dasar dari TOSCA terdiri dari *Service Template* dan *Topology Template*. Pada *Topology Template* didefinisikan dalam bentuk model yang saling terkait antara *Relationship Template* dan *Node Template*. Baik *Node Type* dan *Relationship Type* didefinisikan secara terpisah supaya dapat digunakan kembali. TOSCA saat ini dapat dikatakan sebagai standar *defacto* dari *Infrastructure as Code* dalam mempercepat siklus DevOps (Artac, Borovssak, Di Nitto, Guerriero, dan Tamburri, 2017). Adapun struktur elemen dasar dari TOSCA dapat dilihat pada Gambar .



Gambar 1: Struktur Elemen TOSCA

## 2.2 Markup Language

*Markup language* digunakan sebagai format pertukaran data yang digunakan dalam teknologi komputasi awan, di antaranya JSON yang merupakan bahasa yang mudah dipahami oleh manusia maupun mesin dengan basis bahasa pemrograman Javascript dan YAML yang merupakan format data yang ramah dalam bahasa pemrograman. JSON atau YAML digunakan pada AWS CloudFormation untuk menerjemahkan *resources/stack*, dependensi dan parameter yang diperlukan (Amazon Web Services, 2017).

### 2.3 AST (*Abstract Syntax Tree*)

Pada penelitian ini *Abstract Syntax Tree* menjadi sebuah pondasi usulan pola mendefinisikan sebuah kode yang ditulis oleh manusia agar dimengerti oleh mesin. AST sendiri merupakan pola grafik untuk mendeskripsikan sintaks kode yang abstrak dalam bentuk *tree/pohon*. Membangun AST seharusnya melakukan tiga hal dibawah ini (Fowler dan Mcwhirter, 2007):

1. Mencatat semua masukan tipe data penting dari sebuah rangkaian token (hasil olahan dari analisis *lexical code*).
2. Menyandikan token diatas menjadi susunan struktur *tree/pohon*.
3. Membuat struktur *tree/pohon* yang memudahkan komputer untuk memahami alur struktur *tree/pohon*.

### 3. Analisis dan Desain

Baik JSON dan YAML merupakan format data sehingga tidak dapat disebut bahasa pemrograman. JSON dan YAML hanya sebagai bahasa klasifikasi dari sebuah data. Oleh sebab itu JSON maupun YAML tidak mendukung tipe data, *control flow* dan menjadi sebuah baris yang dapat melakukan proses CRUD (*Create, Read, Update, Delete*).

Berdasarkan kekurangan JSON dan YAML di atas maka pada penelitian ini diusulkan sebuah *query language* untuk mengelola infrastruktur menggunakan prinsip-prinsip pemrograman yang mudah dipahami. *Query language* ini disebut *Biznet Query Language* (BiznetQL) yang diterapkan pada teknologi komputasi awan dengan tujuan:

1. *Reusable*, kode *query language* yang dibuat sebelumnya dapat digunakan kembali untuk membangun atau menduplikasi lingkungan infrastruktur komputasi awan baru.
2. *Scalable*, memudahkan dalam proses penambahan dan pengurangan dari sumber daya infrastruktur komputasi awan.
3. *Maintainable*, memudahkan integrasi dan pemeliharaan pada proses *Continuous Integration* (CI) dan *Continuous Delivery* (CD) infrastruktur komputasi awan.
4. *Testable*, memudahkan pengujian dan migrasi didalam proses pengembangan perangkat lunak. Hal ini mendukung proses *Blue/Green Deployments*.
5. *CRUD-able*, mendukung proses *Create-Read-Update-Delete* dalam pengelolaan sumber daya infrastruktur komputasi awan.

#### 4. Implementasi BiznetQL

BiznetQL dapat diterapkan pada teknologi komputasi awan secara global seperti pada lingkungan yang dibangun di atas Amazon Web Services, Azure, Google Cloud Platform, dan Alibaba Cloud. BiznetQL berkomunikasi ke *Cloud Service API* untuk membangun infrastruktur komputasi awan yang diinginkan menggunakan kode yang sesuai kaidah SQL AST. Contoh kode BiznetQL yang sesuai dengan kaidah SQL AST dapat dilihat pada Tabel 1.

**Tabel 1:** Contoh Kode BiznetQL

No	Kode BiznetQL	Penjelasan
1	CREATE neo.vm 'vm-test' ( 'images': 'Ubuntu 18.04.1 LTS', 'flavor': 'SS2.2')	Membuat mesin virtual dengan nama vm-test, sistem operasi Ubuntu 18.04.1 LTS, dan spesifikasi RAM 2 GB dan vCPU 2 <i>core</i> .
2	SELECT * FROM neo.vm	Menampilkan semua sumber daya dari infrastruktur mesin virtual yang sudah dibuat.
3	UPDATE vm.neo SET 'flavor'='SM4.4' WHERE 'name'='vm-test'	Meningkatkan spesifikasi mesin virtual dari RAM 2 GB dan vCPU 2 <i>core</i> menjadi RAM 4 GB dan vCPU 4 <i>core</i> .
4	DELETE FROM neo.vm WHERE 'name'='vm-test'	Menghapus mesin virtual dengan identitas nama vm-test.

BiznetQL selain mendukung CRUD yang tidak ditemukan pada *Infrastructure as Code* lainnya (JSON dan YAML) juga dapat dioperasikan menggunakan *Read-eval-print loop* (REPL). Kelebihan REPL adalah membuat manajemen sumber daya infrastruktur komputasi awan menjadi lebih interaktif. Gambar menunjukkan BiznetQL yang dioperasikan menggunakan REPL.

```

BQL> SELECT * FROM neo.vm
-----
| ID | Name | Image | Flavor | RAM (GiB) | vCPU | Addresses | Status |
-----
| 9286b274-d194-4e88-8717-1789ccd6bcd6 | swarm-node1 | CentOS 7.3 | SM8.8 | 8 | 8 | network : df | ERROR |
| | | | | | | | fixed IP : 192.168.2.9 | |
-----
| fcfdd0c0-a522-48e9-a4fc-6f030283d3d5 | swarm-test | CentOS 7.3 | SM8.8 | 8 | 8 | network : df | ACTIVE | |
| | | | | | | | fixed IP : 192.168.2.6 | |
| | | | | | | | Floating IP : [REDACTED] | |
-----
| 361a854c-839a-4530-839a-c4b6ac152825 | wp-bynan | CentOS 7.3 | SM8.4 | 8 | 4 | network : df | ACTIVE | |
| | | | | | | | fixed IP : 192.168.2.10 | |
| | | | | | | | Floating IP : [REDACTED] | |
-----
BQL> SELECT ID,Name,Image FROM neo.vm WHERE ID='9286b274-d194-4e88-8717-1789ccd6bcd6'
-----
| ID | Name | Image |
-----
| 9286b274-d194-4e88-8717-1789ccd6bcd6 | swarm-node1 | CentOS 7.3 |
-----
BQL> DELETE FROM neo.vm WHERE ID='9286b274-d194-4e88-8717-1789ccd6bcd6'
Delete 'neo.vm' with ID='9286b274-d194-4e88-8717-1789ccd6bcd6', Status: OK
BQL> SELECT ID,Name,Image FROM neo.vm
-----
| ID | Name | Image |
-----
| fcfdd0c0-a522-48e9-a4fc-6f030283d3d5 | swarm-test | CentOS 7.3 |
-----
| 361a854c-839a-4530-839a-c4b6ac152825 | wp-bynan | CentOS 7.3 |
-----
BQL>

```

Gambar 2: BiznetQL REPL

## 5. Kesimpulan

BiznetQL dapat mendukung proses *orchestration* dan otomatisasi pada infrastruktur komputasi awan dengan menggunakan *query language* sesuai kaidah SQL AST. Kelebihan BiznetQL dibandingkan format data JSON dan YAML yang digunakan oleh Amazon Web Services, Azure, Google Cloud Platform, dan Alibaba Cloud sebagai *Infrastructure as Code* adalah lebih interaktif dan *CRUD-able*. Jadi kode BiznetQL lebih memudahkan entitas pada divisi DevOps lebih efisien dalam bekerja.

## Ucapan Terima Kasih

Penulis menyampaikan terima kasih kepada PT. Biznet Gio Nusantara yang telah memberikan dukungan dalam kolaborasi penelitian bidang komputasi awan.

## Daftar Pustaka

Amazon Web Services., 2016. *Blue / Green Deployments on AWS*.

Amazon Web Services., 2017. *Infrastructure as Code*.

Artac, M., Borovssak, T., Di Nitto, E., Guerriero, M., dan Tamburri, D. A., 2017. DevOps: Introducing infrastructure-as-code. *Proceedings - 2017 IEEE/ACM 39th International Conference on Software Engineering Companion, ICSE-C 2017*, 497–498. <https://doi.org/10.1109/ICSE-C.2017.162>

- Bass, L., Weber, I., dan Zhu, L., 2015. *DevOps: A Software Architect's Perspective*. Addison-Wesley Professional.
- Fowler, M., dan McWhirter, B., 2007. *The Definitive ANTLR Reference*. Dallas: The Pragmatic Bookshelf.
- Lipton, P. (Ca T.), Moser, S. (Ibm), Palma, D. (Vnomic), dan Spatzier, T. (Ibm)., 2013. Topology and Orchestration Specification for Cloud Applications, (March), 1–114. Diambil dari <http://docs.oasis-open.org/tosca/TOSCA/v1.0/cs01/TOSCA-v1.0-cs01.html>
- Morris, K., 2016. *Infrastructure as Code: Managing Servers in The Cloud*. O'Reilly Media, Inc.
- OASIS., 2018. Diambil 29 Juni 2018, dari <https://www.oasis-open.org/org>
- Radez, D., 2015. *OpenStack Essentials*. Packt Publishing.
- Rahman, A., Mahdavi-Hezaveh, R., dan Williams, L., 2018. *Where Are The Gaps? A Systematic Mapping Study of Infrastructure as Code Research*, (1). Diambil dari <http://arxiv.org/abs/1807.04872>

