

PENGEMBANGAN SISTEM PENYEDIAAN VM MENGGUNAKAN TERRAFORM SEBAGAI ALAT PRAKTIK PaaS DI SMK NEGERI 2 YOGYAKARTA

Afnan Ngathour Rahman¹, Eko Marpanaji²

Fakultas Teknik Universitas Negeri Yogyakarta

E-mail: afnanngathour.2020@student.uny.ac.id

ABSTRAK

Penelitian ini bertujuan mengembangkan sistem penyediaan *Virtual Machine* (VM) menggunakan Terraform sebagai alat praktik *Platform as a Service* (PaaS) di SMK Negeri 2 Yogyakarta. Tujuan utama adalah memanfaatkan *private cloud* milik sekolah secara maksimal untuk meningkatkan pengalaman praktikum siswa. Sistem ini diterapkan pada infrastruktur Proxmox VE Cluster dengan empat *node server*. Metode penelitian menggunakan model *Network Development Life Cycle* (NDLC) yang terdiri dari tahap *analysis, design, simulation prototyping, implementation, monitoring, and management*. Pengumpulan data dilakukan melalui observasi, wawancara, dan pengujian langsung di SMK Negeri 2 Yogyakarta. Subjek penelitian mencakup dua ahli infrastruktur *cloud* dan satu guru PaaS sebagai pengguna akhir. Teknik analisis data melibatkan analisis deskriptif dan uji standar ISO/IEC 25010 untuk aspek *functional suitability, usability, and maintainability*. Hasil penelitian menunjukkan bahwa sistem penyediaan VM menggunakan Terraform dapat meningkatkan efisiensi dan efektivitas pengelolaan infrastruktur *cloud* di sekolah. Pengujian aspek *functional suitability* menunjukkan persentase keberhasilan 100%, sementara aspek *usability* mendapatkan skor 84,17 (A) berdasarkan *System Usability Scale* (SUS). Pengujian *maintainability* juga mendapatkan nilai 100%, menunjukkan bahwa sistem ini mudah dikelola dan dikembangkan.

Kata kunci: *Cloud Computing, ISO 25010, NDLC, Terraform, Virtual Machine*

ABSTRACT

This research aims to develop a Virtual Machine (VM) provisioning system using Terraform as a Platform as a Service (PaaS) practice tool at SMK Negeri 2 Yogyakarta. The main objective is to maximally utilize the school's private cloud to improve students' practicum experience. The system is implemented on a Proxmox VE Cluster infrastructure with four server nodes. The research method uses the Network Development Life Cycle (NDLC) model which consists of analysis, design, simulation prototyping, implementation, monitoring, and management stages. Data collection was conducted through observation, interviews, and direct testing at SMK Negeri 2 Yogyakarta. The research subjects included two cloud infrastructure experts and one PaaS teacher as the end user. Data analysis techniques involved descriptive analysis and ISO/IEC 25010 standard tests for functional suitability, usability, and maintainability aspects. The results showed that the VM provisioning system using Terraform can improve the efficiency and effectiveness of cloud infrastructure management in schools. Testing the functional suitability aspect shows a 100% success percentage, while the usability aspect gets a score of 84.17 (A) based on the System Usability Scale (SUS). The maintainability test also scored 100%, indicating that the system is easy to manage and develop.

Keyword: *Cloud Computing, ISO 25010, NDLC, Terraform, Virtual Machine*

PENDAHULUAN

Cloud computing secara sederhana berarti menyimpan dan mengakses data dan program melalui internet, bukan *hard drive* komputer. *Cloud* hanyalah sebuah metofora untuk internet [1]. Banyak perusahaan teknologi informasi telah menggunakan konsep *Infrastructure as Code* (IaC), khususnya Terraform, untuk menentukan dan menyiapkan infrastruktur seperti jaringan dan komputer virtual menggunakan kode. Sebagai sekolah yang berbasis industri, Sekolah Menengah Kejuruan (SMK) harus mengikuti perkembangan dunia industri dalam hal pembelajaran dan sarana dan prasarana yang dapat dimanfaatkan siswa dan guru untuk mencapai tujuan pembelajaran dengan efektif dan efisien.

SMK Negeri 2 Yogyakarta terletak di Cokrodingratan, Kecamatan Jetis, Kota Yogyakarta, Daerah Istimewa Yogyakarta. Pendidikan menengah kejuruan dimaksudkan untuk menyediakan siswa untuk memasuki dunia kerja atau mengikuti pendidikan keprofesian pada tingkat yang lebih tinggi. SMK Negeri 2 Yogyakarta telah menggunakan Kurikulum Merdeka sejak angkatan 2022. Kurikulum Merdeka memberikan kepada sekolah dan guru kebebasan untuk memilih metode dan perangkat ajar yang sesuai dengan kemampuan mereka, serta dengan sarana dan prasarana yang tersedia di sekolah. Karena memiliki sejumlah server atau *mini data center*, sarana dan prasarana SMK Negeri 2 Yogyakarta sudah siap untuk memungkinkan

pembelajaran menggunakan *private cloud*. Meskipun demikian, proses pembelajaran masih kurang memanfaatkan *private cloud*, terutama di jurusan Sistem Informasi Jaringan dan Aplikasi (SIJA). Ini karena sistem administrasi memiliki banyak tugas dan sumber daya yang kurang.

Salah satu mata pelajaran wajib di jurusan SIJA adalah *Platform as a Service* (PaaS). Materi ini meliputi instalasi Ubuntu Server, DNS Server, Web Server, Database Server, dan Mail Server. Siswa menggunakan VirtualBox sebagai alat pembelajaran selama praktikum, dan menggunakan untuk membuat *virtual machine* dengan sistem operasi Ubuntu. Hasilnya menunjukkan bahwa banyak siswa mengalami masalah selama praktikum. Ini termasuk komputer yang lambat saat melakukan *multitasking*, kesalahan saat menginstal server Ubuntu, kesalahan saat menjalankan VirtualBox, *virtual machine* dihapus oleh blok pembelajaran sebelumnya, dan sejumlah kesalahan pada sistem operasi Ubuntu itu sendiri.

Penelitian ini dilakukan dengan tujuan mengembangkan sistem penyediaan *virtual machine* dengan menggunakan Terraform sebagai alat praktik *platform as a service* di SMK Negeri 2 Yogyakarta. Tujuan dari penelitian ini adalah untuk memaksimalkan penggunaan *private cloud* yang dimiliki SMK Negeri 2 Yogyakarta. Dalam beberapa penelitian sebelumnya, sistem penyediaan *virtual machine* telah dikembangkan di berbagai penyedia *cloud* publik seperti Google

Cloud Platform, Microsoft Azure, dan Amazon Web Services. Perbedaan dari penelitian sebelumnya adalah sistem yang akan dikembangkan akan menggunakan Proxmox Virtual Environment sebagai sumber daya *private cloud*, dan akan diuji terlebih dahulu sebelum digunakan oleh pengguna akhir. Untuk menguji sistem ini, standar ISO/IEC 25010 akan diterapkan pada aspek fungsional *suitability*, *usability*, dan *maintainability*.

METODE

Jenis Penelitian

Penelitian ini merupakan penelitian *Research & Development* (R&D) sebagai pendekatan yang digunakan untuk menciptakan dan menguji sistem yang dikembangkan menggunakan metode *Network Development Life Cycle* (NDLC). NDLC memiliki 6 tahapan, yaitu *analysis*, *design*, *simulation prototyping*, *implementation*, *monitoring* dan *management*.

Prosedur Pengembangan

Prosedur pengembangan akan menggunakan model pengembangan *Network Development Life Cycle* (NDLC) yang memiliki 6 tahapan:

1. Analysis

Tahapan *analysis* merupakan tahapan awal, dimana fokus utamanya adalah memahami kebutuhan, tantangan, dan lingkungan yang ada sebelum memulai pengembangan.

2. Design

Pada tahap ini, akan dibangun sebuah rancangan topologi jaringan dan rancangan fitur sistem.

3. Simulation Prototyping

Tahap *simulation prototyping* melibatkan pembuatan versi awal atau *development* dari solusi yang dirancang sebelumnya untuk menguji fungsionalitasnya sebelum implementasi penuh.

4. Implementation

Tahap implementasi adalah tahap di mana solusi yang telah dirancang dan diuji diimplementasikan dalam lingkungan produksi.

5. Monitoring

Tahap *monitoring* melibatkan pemantauan kinerja dan ketersediaan sistem yang telah diimplementasikan untuk memastikan bahwa semuanya berjalan dengan baik dan memenuhi kebutuhan pengguna.

6. Management

Tahap *management* merupakan tahap berkelanjutan, di mana sistem dan infrastruktur yang telah diimplementasikan akan dikelola dan dipelihara secara teratur.

Tempat dan Subjek Penelitian

Penelitian ini dilaksanakan di SMK Negeri 2 Yogyakarta dengan subjek penelitian mencakup dua ahli infrastruktur *cloud* dan satu guru PaaS sebagai pengguna akhir.

Metode dan Alat Pengumpulan Data

Teknik Analisis Data

Teknik analisis data melibatkan analisis deskriptif dan uji standar ISO/IEC

25010 untuk aspek *functional suitability*, *usability*, dan *maintainability*.

1. Functional Suitability

Pengujian kualitas pada aspek *functional suitability* menggunakan skala Guttman dimana jawaban dari *test case* yang disediakan hanya "berhasil-gagal" yang mewakili dari 1 dan 0. Dengan menggunakan skala ini akan didapat jawaban yang tegas terhadap suatu permasalahan yang ditanyakan [2]. Hasil dari pengujian dapat dilakukan analisis menggunakan perhitungan berikut [3]:

$$\frac{\sum x}{n} \times 100\%$$

$$\frac{\sum y}{n} \times 100\%$$

Keterangan:

n : Jumlah semua fungsi yang ada

$\sum x$: Jumlah fungsi yang berjalan dengan baik

$\sum y$: Jumlah fungsi yang tidak berjalan dengan baik

Hasil dari perhitungan tersebut menunjukkan persentase jumlah fitur yang berjalan dengan baik atau tidak berjalan dengan baik. Pada pengujian aspek *functional suitability*, perangkat lunak dianggap baik jika proporsi fungsi yang berfungsi dengan baik (X) lebih besar daripada persentase fungsi yang tidak berfungsi dengan baik (Y).

2. Usability

Analisis instrumen pada pengujian *usability* menggunakan skala likert yang memiliki lima opsi jawaban, mulai dari "Sangat Tidak Setuju", "Tidak Setuju", "Netral", "Setuju", dan "Sangat Setuju",

sehingga setiap opsi jawaban memiliki skornya masing-masing. Skala Likert adalah alat yang digunakan untuk menilai sikap, pendapat dan persepsi seseorang atau sekelompok orang tentang fenomena sosial [2]. Untuk menghitung skor dari SUS langkah pertama adalah menyesuaikan skor mentah menjadi skor yang berkisar 0 (terburuk) hingga 4 (terbaik), dengan penyesuaian yang berbeda untuk nomor ganjil (pertanyaan nada positif) dan genap (pertanyaan nada negatif). Untuk pertanyaan bermomor ganjil (1, 3, 5, 7, 9), kurangi 1 dari skor mentah dan untuk pertanyaan bermomor genap (2, 4, 6, 8, 10) kurangi skor mentah dari 5. Hitung jumlah skor yang telah disesuaikan kemudian kalikan dengan 2,5 untuk mendapatkan skor SUS standar dengan nilai akhir 0 sampai 100 [4]. Untuk lebih mudahnya dapat dilihat pada persamaan dibawah ini:

$$SUS = 2.5((q1 - 1) + (5 - q2) + (q3 - 1) + (5 - q4) + (q5 - 1) + (5 - q6) + (q7 - 1) + (5 - q8) + (q9 - 1) + (5 - q10))$$

Keterangan:

q : Skor mentah dari setiap nomor

3. Maintainability

Pengujian ini meliputi 3 aspek yaitu *instrumentation*, *consistency*, dan *simplicity*. Hasil penilaian untuk setiap aspek tersebut dicatat, kemudian ditentukan kesesuaian antara hasil dan penilaian yang diharapkan. Kesimpulan diambil menggunakan skala Guttman, di mana isian untuk setiap *test case* hanya berupa "ya" atau "tidak", yang masing-masing mewakili nilai 1 dan 0. Setelah disimpulkan hasil dari pengujian dapat

dianalisis menggunakan perhitungan yang sama dengan pengujian *functional suitability* dengan sedikit penyesuaian sebagai berikut:

$$\frac{\Sigma x}{n} \times 100\%$$

$$\frac{\Sigma y}{n} \times 100\%$$

Keterangan:

n : Jumlah semua aspek yang ada

HASIL DAN PEMBAHASAN

Tujuan dari penelitian ini adalah untuk mengembangkan sistem penyediaan virtual machine (VM) menggunakan Terraform sebagai alat praktik platform PaaS di SMK Negeri 2 Yogyakarta. Model pengembangan Network Development Life Cycle (NDLC) terdiri dari tahap analisis, desain, simulasi prototipe, penerapan, pengawasan, dan manajemen.

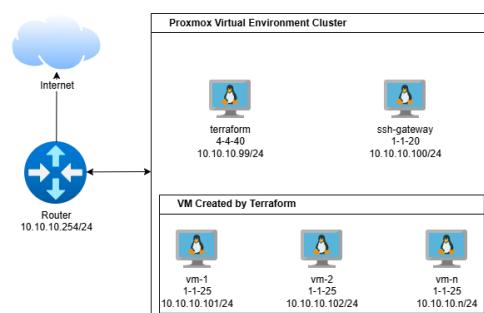
Tahap analisis merupakan proses untuk memahami kebutuhan, kendala, dan lingkungan yang ada sebelum memulai pengembangan sistem. Tahap ini melibatkan observasi, wawancara, dan analisis lingkungan untuk mengidentifikasi kebutuhan fungsional dan non-fungsional. Kebutuhan pengguna diidentifikasi melalui observasi pembelajaran *Platform as a Service* (PaaS) dan wawancara dengan Bapak Haryanto, S.T., M.Kom., yang menjabat sebagai kepala ICT di SMK Negeri 2 Yogyakarta. Berdasarkan hasil tersebut, diketahui bahwa Jurusan SIJA di SMK Negeri 2 Yogyakarta memiliki *private cloud* dengan menggunakan virtualisasi Proxmox VE yang terdiri dari 4 *node server* (pve1, pve2, pve3, dan pve4). Keempat *node server* ini kemudian

Σx : Jumlah aspek yang sesuai
 Σy : Jumlah aspek yang tidak sesuai

Hasil dari perhitungan tersebut menunjukkan persentase jumlah aspek yang sesuai atau tidak sesuai. Pada pengujian aspek *maintainability*, perangkat lunak dianggap baik jika proporsi kesesuaian aspek (X) lebih besar daripada persentase aspek yang tidak sesuai (Y).

digabungkan menjadi sebuah Proxmox VE Cluster. Selain itu, saat ini proses penyediaan VM dilakukan secara manual oleh siswa menggunakan virtualisasi VirtualBox di komputer masing-masing.

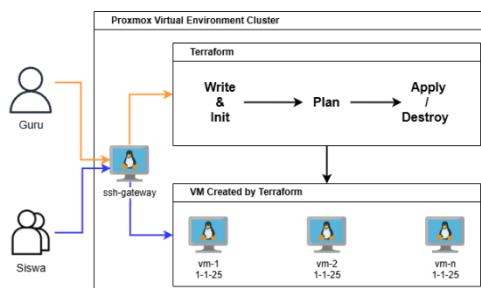
Pada tahap desain, telah dilakukan perancangan sistem penyediaan VM menggunakan Terraform yang memiliki sebelas fitur utama. Selain itu, desain infrastruktur (*physical topology*) Terraform dilakukan dalam Proxmox Cluster SMK Negeri 2 Yogyakarta, yang melibatkan koneksi router, Proxmox VE Cluster dengan 4 node servers, VM Terraform, dan ssh-gateway.



Gambar 1. Topologi Fisik

Rancangan sistem penyediaan VM menggunakan Terraform mencakup desain alur sistem (*logical topology*). Proses ini melibatkan tiga tahap utama: *Write*, *Plan*, dan *Apply*, yang dilakukan

oleh guru untuk menyesuaikan modul Terraform, meninjau rancangan infrastruktur, dan menyediakan VM sesuai kebutuhan.



Gambar 2. Topologi Logis

Untuk memastikan bahwa kode dan konfigurasi yang ditulis berfungsi dengan baik sebelum diterapkan di lingkungan produksi di SMK Negeri 2 Yogyakarta, tujuan utama dari tahap *simulation prototyping* adalah menulis kode Terraform, mengunggahnya ke GitHub, dan melakukan uji coba sistem penyediaan VM di lingkungan pengembangan dengan satu *node* Proxmox VE. Dengan menggunakan Visual Studio Code, kode Terraform ditulis langsung di *virtual machine* terraform-dev melalui koneksi SSH. Selain itu, fitur *Source Control* dari Visual Studio Code digunakan untuk mengunggah setiap perubahan pada kode ke *repository* GitHub. Pengujian dilakukan di lingkungan pengembangan dengan spesifikasi *server* tertentu untuk mengetahui berapa banyak *virtual machine* (VM) Terraform yang dapat disediakan dalam satu kali eksekusi, yang saat ini diketahui sebanyak enam VM. Langkah-langkah pengujian termasuk menjalankan script *init.sh* untuk membuat batch modul terraform dan script *apply.sh* untuk menyediakan VM, yang memerlukan sekitar 19 menit 27 detik untuk menyediakan 18 VM yang dibagi dalam 3 *batch*. Hasil pengujian

menunjukkan bahwa Proxmox VE dapat menjalankan maksimal 4 tugas secara paralel, dan Terraform berhasil menyediakan 18 VM dalam konfigurasi yang ditentukan, membuktikan bahwa sistem penyediaan VM menggunakan Terraform dapat berjalan dengan baik dalam lingkungan pengembangan yang disimulasikan.

Pada tahap *implementation* pengembangan sistem penyediaan VM menggunakan Terraform di SMK Negeri 2 Yogyakarta dibagi menjadi empat tahap utama: persiapan, instalasi Terraform, modifikasi kode sumber, dan pelaksanaan sistem penyediaan VM. Pada tahap persiapan, beberapa komponen penting seperti VM, Proxmox API Token, dan *Image Template* disiapkan dengan spesifikasi tertentu dan autentikasi SSH diaktifkan. Selanjutnya, SSH key dibuat untuk pengguna di VM ssh-gateway, dan Proxmox API Token dibuat dengan langkah-langkah menciptakan pengguna, API token, dan mengatur izin yang diperlukan. Kemudian, *image template* Ubuntu diunduh, dimodifikasi, dan disalin ke server Proxmox lain untuk dijadikan VM template dibuat di setiap server. Pada tahap instalasi Terraform, Terraform diinstal pada VM terraform dengan mengaksesnya melalui SSH dan menjalankan perintah instalasi yang diperlukan. Dalam tahap modifikasi *source code*, *source code* diunduh dari GitHub, dan file utama seperti *provider.tf*, *variabel.tf*, *variabel.auto.tfvars*, dan *main.tf* dimodifikasi sesuai infrastruktur yang ada. Kapasitas penggunaan dihitung dengan menyusun 36 VM menjadi 5 *batch* untuk memenuhi kebutuhan siswa. Modifikasi *bash script* dilakukan untuk menyesuaikan informasi VM yang akan disediakan, termasuk script *init.sh*, *plan.sh*, *apply.sh*, dan

destroy.sh. Pada tahap menjalankan sistem penyediaan VM, Terraform diinisialisasi, rencana infrastruktur dibuat, dan konfigurasi diterapkan dengan menjalankan *script* yang telah dimodifikasi, menghasilkan 36 VM dalam 5 *batch* yang disediakan di pve1 (Gambar 3), pve2 (Gambar 4), pve3 (Gambar 5), dan pve4 (Gambar 6). Setelah VM disediakan, siswa dapat mengakses VM mereka melalui VM ssh-gateway menggunakan protokol SSH. Panduan pengguna juga disediakan untuk memastikan semua pengguna dapat memanfaatkan sistem dengan efektif.

```
root@pve1:~# qm list | awk 'NR==1 || /[v]m-batch/'
```

VMID	NAME	STATUS	MEM(MB)	BOOTDISK(GB)	PID
200	vm-batch-1-1	running	1024	25.00	2669755
204	vm-batch-1-5	running	1024	25.00	2669828
208	vm-batch-2-1	running	1024	25.00	2611373
212	vm-batch-2-5	running	1024	25.00	2610455
216	vm-batch-3-1	running	1024	25.00	2612398
220	vm-batch-3-5	running	1024	25.00	2613223
224	vm-batch-4-1	running	1024	25.00	2614785
228	vm-batch-4-5	running	1024	25.00	2614694
232	vm-batch-5-1	running	1024	25.00	2615725

Gambar 3. VM di pve1

```
root@pve2:~# qm list | awk 'NR==1 || /[v]m-batch/'
```

VMID	NAME	STATUS	MEM(MB)	BOOTDISK(GB)	PID
201	vm-batch-1-2	running	1024	25.00	2582933
205	vm-batch-1-6	running	1024	25.00	2582881
209	vm-batch-2-2	running	1024	25.00	2583423
213	vm-batch-2-6	running	1024	25.00	2584114
217	vm-batch-3-2	running	1024	25.00	2585467
221	vm-batch-3-6	running	1024	25.00	2585681
225	vm-batch-4-2	running	1024	25.00	2586989
229	vm-batch-4-6	running	1024	25.00	2586228
233	vm-batch-5-2	running	1024	25.00	2587823

Gambar 4. VM di pve2

```
root@pve3:~# qm list | awk 'NR==1 || /[v]m-batch/'
```

VMID	NAME	STATUS	MEM(MB)	BOOTDISK(GB)	PID
202	vm-batch-1-3	running	1024	25.00	2576739
206	vm-batch-1-7	running	1024	25.00	2576752
210	vm-batch-2-3	running	1024	25.00	2577798
214	vm-batch-2-7	running	1024	25.00	2578349
218	vm-batch-3-3	running	1024	25.00	2579699
222	vm-batch-3-7	running	1024	25.00	2579769
226	vm-batch-4-3	running	1024	25.00	2581832
230	vm-batch-4-7	running	1024	25.00	2580461
234	vm-batch-5-3	running	1024	25.00	2581694

Gambar 5. VM di pve3

```
root@pve4:~# qm list | awk 'NR==1 || /[v]m-batch/'
```

VMID	NAME	STATUS	MEM(MB)	BOOTDISK(GB)	PID
283	vm-batch-1-4	running	1024	25.00	2565746
287	vm-batch-1-8	running	1024	25.00	2565696
211	vm-batch-2-4	running	1024	25.00	2567214
215	vm-batch-2-8	running	1024	25.00	2566788
219	vm-batch-3-4	running	1024	25.00	2568116
223	vm-batch-3-8	running	1024	25.00	2568045
227	vm-batch-4-4	running	1024	25.00	2569675
231	vm-batch-4-8	running	1024	25.00	2569263
235	vm-batch-5-4	running	1024	25.00	2570525

Gambar 6. VM di pve4

Dalam siklus pengembangan sistem NDLC, tahap *monitoring* berpusat pada pengawasan dan evaluasi kinerja sistem secara konsisten. Tujuan dari pengawasan adalah untuk memastikan

bahwa sistem penyediaan *virtual machine* (VM) beroperasi sesuai dengan spesifikasi dan kebutuhan pengguna, serta untuk menemukan dan mengatasi masalah sejak dini sebelum menjadi masalah yang lebih serius. Selain itu, tujuan dari tahap *monitoring* ini adalah untuk menguji sistem sesuai dengan standar ISO 25010, khususnya dalam hal kesesuaian fungsional, kemudahan penggunaan, dan pemeliharaan. Pengujian ini dilakukan oleh dua praktisi profesional infrastruktur dan pengguna akhir. Hasil pengujian dapat dilihat pada Tabel 1.

Tabel 1. Hasil Pengujian

Aspek Pengujian	Hasil
Functional Suitability	100%
Usability	84,17 (A)
Maintainability	100%

Untuk sistem penyediaan *virtual machine* (VM) di SMK Negeri 2 Yogyakarta, tahap manajemen dalam siklus pengembangan sistem NDLC mencakup pengelolaan dan pemeliharaan sistem yang berkelanjutan untuk memastikan kinerja dan efisiensi penggunaan infrastruktur yang optimal. Untuk mencegah kerusakan dan memastikan sistem aman dan *up-to-date*, aktivitas pemeliharaan termasuk pemantauan kinerja, pengelolaan sumber daya, dan pembaruan perangkat lunak. Salah satu proses yang sangat penting adalah penggunaan *script* "destroy.sh" untuk menghapus *virtual machine* (VM) yang tidak lagi diperlukan untuk membantu membebaskan sumber daya.

SIMPULAN

Penelitian ini menghasilkan sistem penyediaan *virtual machine* menggunakan Terraform sebagai alat praktik *platform as a service* di SMK Negeri 2 Yogyakarta. Hasil akhir sistem

penyediaan VM ini sudah berhasil membantu guru dan siswa untuk dapat memanfaatkan *private cloud* yang dimiliki sekolah secara maksimal dan meningkatkan pengalaman praktikum siswa. Sistem penyediaan VM menggunakan Terraform telah diuji menggunakan standar *software quality model* ISO/IEC 25010 pada aspek *functional suitability*, *usability*, dan *maintainability*. Pengujian aspek *functional suitability* menunjukkan bahwa semua fungsi yang diuji berjalan dengan baik, dengan persentase keberhasilan 100%. Hal ini menunjukkan bahwa sistem ini mampu memenuhi kebutuhan yang telah direncanakan. Berdasarkan hasil pengujian *usability* menggunakan *System Usability Scale* (SUS), sistem ini mendapatkan skor 84,17 (A), yang menunjukkan bahwa pengguna merasa sistem ini mudah digunakan. Pengujian aspek *maintainability* menunjukkan bahwa sistem memberikan peringatan dan mengidentifikasi kesalahan saat terjadi, menggunakan model rancangan yang konsisten, dan mudah dikelola serta dikembangkan. Sehingga pada pengujian

aspek *maintainability* mendapatkan nilai 100%.

DAFTAR RUJUKAN

- [1] Rashid, A., & Chaturvedi, A. (2019). Cloud computing characteristics and services: a brief review. International Journal of Computer Sciences and Engineering, 7(2), 421-426. <https://doi.org/10.26438/ijcse/v7i2.421426>
- [2] Sugiyono, D. (2013). Metode penelitian pendidikan pendekatan kuantitatif, kualitatif dan R&D
- [3] Pindoyono, P., & Nurkhamid, N. (2017). Pengujian Functionality dan Performance Sistem Inforamasi Berbasis Web Menggunakan Framework Codeigniter di SMK Negeri 1 Jogonalan Klaten. Elinvo (Electronics, Informatics, and Vocational Education), 2(2), 190-194. doi:<https://doi.org/10.21831/elinvo.v2i2.17313>
- [4] Lewis, J. R. (2018). The system usability scale: past, present, and future. International Journal of Human-Computer Interaction, 34(7), 577-590.