

**PENERAPAN LAYANAN CLOUD SERVER SECARA SELF-SERVICE  
MENGUNAKAN MODEL FINITE STATE AUTOMATA**

**IMPLEMENTATION OF SELF-SERVICE CLOUD SERVER USING THE FINITE  
STATE AUTOMATA MODEL**

**Muchamad Bachram Shidiq<sup>1</sup>, Windu Gata<sup>2</sup>, Hafifah Bella Novitasari<sup>3</sup>, Achmad  
Bayhaqy<sup>4</sup>, Hendra Setiawan<sup>5</sup>**

<sup>1,2,4</sup>Universitas Nusa Mandiri,

<sup>3,5</sup>Sekolah Tinggi Manajemen Informatika dan Komputer Bani Saleh,  
[14210136@nusamandiri.ac.id](mailto:14210136@nusamandiri.ac.id)

**ABSTRACT**

*Cloud computing is a technology that allows users to use internet-based computing services. Along with the advancement of cloud computing technology, several government agencies have implemented cloud computing, one of them is PUSINTEK Secretariat General of the Ministry of Finance. One of PUSINTEK's services in the field of cloud computing is to provide cloud server services in the form of a Virtual Machine (VM) that can be accessed by users. In the provision of these services, there are various difficulties, such as creating a VM which takes a long time and the potential for errors when creating a VM. This research will discuss service design and self-service cloud server service system design using the Finite State Automata model with 7 (seven) states and transition functions that successfully accept various possible input strings. The application of the FSA model in the cloud server service cycle flow is expected to shorten the service fulfillment time and reduce the potential for errors in creating the VM because it is done independently by the user.*

**Keywords:** *Cloud Computing, Cloud Server, Finite State Automata, PUSINTEK, self-service*

**ABSTRAK**

*Cloud computing merupakan teknologi yang memungkinkan pengguna untuk menggunakan layanan komputasi berbasis internet. Seiring dengan perkembangan teknologi cloud computing, beberapa instansi pemerintahan sudah melakukan implementasi cloud computing, salah satunya adalah PUSINTEK Sekretariat Jenderal Kementerian Keuangan. Salah satu layanan PUSINTEK dalam bidang cloud computing adalah menyediakan layanan cloud server berupa Virtual Machine (VM) yang dapat diakses oleh pengguna. Pada penyelenggaraan layanan dimaksud terdapat berbagai kesulitan, seperti pembuatan VM yang membutuhkan waktu lama dan adanya potensi kesalahan saat pembuatan VM. Penelitian ini akan membahas desain layanan dan rancangan sistem layanan cloud server secara self-service menggunakan model Finite State Automata dengan 7 (tujuh) state serta fungsi transisi yang berhasil menerima berbagai kemungkinan string input. Penerapan model FSA pada alur siklus layanan cloud server ini diharapkan dapat mempersingkat waktu pemenuhan layanan dan mengurangi potensi kesalahan pembuatan VM tersebut karena dilakukan secara mandiri oleh pengguna.*

**Kata Kunci:** *Cloud Computing, Cloud Server, Finite State Automata, PUSINTEK, self-service*

**PENDAHULUAN**

*Cloud Computing* merupakan teknologi yang memungkinkan pengguna untuk memakai atau menggunakan layanan komputasi berupa server, storage, network dan lain sebagainya dengan menggunakan akses jaringan internet (Kurniawan, 2015). Pada umumnya, *cloud computing* bersifat *public*, *private*, *community* dan

*hybrid cloud* (Ashari et al., 2011). *Public cloud* adalah layanan *cloud* yang disediakan oleh pihak tertentu dan memiliki jaringan yang tersebar di seluruh tempat dunia sehingga layanan ini bisa digunakan oleh siapa saja (Ashari et al., 2011; Darmayunata, et al., 2021) *Private cloud* adalah layanan *cloud* yang memiliki kelebihan hampir menyerupai *public cloud*, namun memiliki tingkat

keamanan data yang baik serta fleksibilitas yang bisa disesuaikan dengan organisasi namun memiliki infrastruktur yang bersifat pribadi atau eksklusif (Ashari et al., 2011), *community cloud* merupakan infrastruktur *cloud* yang digubakan secara bersama oleh organisasi dan mendukung komunitas tertentu (Ashari et al., 2011), sedangkan *hybrid cloud* adalah perpaduan infrastruktur yang sudah ada, yaitu *public*, *private*, dan *community cloud* (Ashari et al., 2011; Rizki, et al., 2021).

Saat ini penggunaan *cloud computing (cloud)* menjadi sangat penting baik bagi unit organisasi atau perusahaan yang bergerak di bidang Teknologi Informasi (TI) maupun tidak. *Cloud* menjadi angin segar bagi perusahaan rintisan yang belum memiliki cukup modal dalam pembiayaan infrastruktur Teknologi Informasi dan Teknologi (TIK) untuk memulai proses bisnisnya (Rumetna, 2018). Dengan adanya *cloud*, sebuah perusahaan hanya perlu melakukan sewa untuk menggunakan teknologi komputasi yang diperlukan untuk membangun sistem informasi perusahaannya, tanpa harus membeli perangkat server, storage, jaringan dan lain sebagainya. Selain itu, terdapat banyak alasan lain mengapa teknologi *cloud* lebih menjadi pilihan utama dibandingkan *on-premise*, antara lain:

- 1 teknologi *cloud* memiliki elastisitas yang jauh lebih baik dibandingkan dengan *on-premise*;
- 2 teknologi *cloud* memiliki keluasaan dan kedalaman fungsi yang lebih signifikan dibandingkan dengan server *on-premise*;
- 3 teknologi *cloud* lebih *agile* dalam menyokong kebutuhan para perusahaan rintisan (Utami, 2018).

Tentunya penggunaan teknologi *cloud* tidak hanya digunakan oleh para

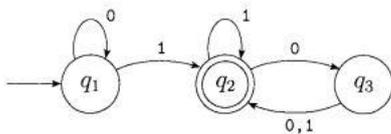
perusahaan rintisan, di lingkungan pemerintahan juga sudah dilakukan pemanfaatan teknologi *cloud*. Pemanfaatan teknologi *cloud* dirasa dapat membantu pemerintah dalam mengintegrasikan data, selain itu dari sisi pengguna akan lebih hemat biaya dan waktu, bebas risiko dan juga tidak perlu menyediakan ruangan server (Wildana, 2017).

Pada instansi Kementerian Keuangan, Pusat Sistem Informasi dan Teknologi Keuangan (PUSINTEK) merupakan salah satu unit organisasi setingkat eselon 2 di bawah Sekretariat Jenderal yang bertugas untuk menjadi penyedia layanan TIK bagi para unit organisasi di bawah Kementerian Keuangan. Salah satu layanan yang disediakan oleh PUSINTEK adalah memberikan layanan *cloud server development* bagi para stakeholder. Layanan *cloud server development* adalah layanan penyediaan server development bagi pengguna terkait dalam rangka pengembangan atau uji coba pengembangan sistem informasi bagi masing-masing unit organisasi. Bentuk layanan yang diberikan kepada pengguna adalah sebuah server berupa *virtual machine (VM)* dengan berbagai macam sistem operasi (Windows Server, Ubuntu Server, dan Centos Server) dengan batas waktu pemakaian adalah 12 bulan (Pusintek, 2020).

Pokok masalah dalam pemberian layanan *cloud* ini adalah layanan masih disediakan dengan cara yang manual, sehingga membutuhkan waktu tersendiri bagi Tim Pengelola VM Pusintek untuk membagi tiket atau *request* tersebut ke masing-masing petugas pengelola VM. Hal ini menyebabkan tidak optimalnya waktu pembuatan VM. Selain itu, terdapat kemungkinan potensi kesalahan konfigurasi apabila pembuatan server dilakukan secara manual oleh tim Pengelola VM.

Dalam rangka mengatasi permasalahan tersebut di atas, diperlukan otomatisasi layanan *cloud server* secara *self-service* oleh pengguna. Hal ini nantinya akan memudahkan pengguna untuk menggunakan layanan *cloud server*, menjadikan waktu pembuatan VM menjadi optimal, dan mengurangi potensi kesalahan konfigurasi (Muhammad, et al., 2022).

Berdasarkan alasan-alasan tersebut, pada penelitian ini penulis melakukan analisis dan desain perancangan sistem menggunakan *Finite State Automata* untuk melakukan identifikasi dan penyempurnaan proses penyediaan layanan *cloud server* Kementerian Keuangan yang dikelola oleh PUSINTEK sehingga menjadi layanan yang *self-service*. *Finite State Automata* (FSA) merupakan model komputasi paling sederhana dengan menggunakan memory yang terbatas (Sipser, 2006). Namun dengan keterbatasan tersebut, FSA dapat melakukan melakukan sesuatu yang bermanfaat sebagai inti dari peralatan elektromekanik (Sipser, 2006; Desvia, et al., 2021).



**Gambar 1. Contoh Diagram State FSA**

Pada gambar 1, terdapat 3 (tiga) *state* yang memiliki label  $q1$ ,  $q2$ , dan  $q3$ . *State* awal,  $q1$ , ditunjukkan dengan adanya arah anak panah yang datang dari sembarang awal, sedangkan lingkaran ganda menunjukkan *state* akhir, dan anak panah menunjukkan arah transisi *state*.

Terdapat 2 (dua) cara yang digunakan untuk menuliskan FSA, yaitu menggunakan diagram transisi dan tabel transisi. Diagram transisi merupakan

suatu graf yang didefinisikan, sebagai berikut:

1. Terdapat simpul untuk setiap state  $Q$ .
2. Untuk setiap state  $q \in Q$  dan setiap simbol input  $a \in \Sigma$ , berlaku  $\delta(q, a) = p$ . Diagram transisi memiliki busur berlabel  $a$  dari state  $q$  ke state  $p$ .
3. Terdapat anak panah berlabel start yang mengarah ke state awal  $q_0$  dan anak panah ini tidak berasal dari state manapun.
4. State yang merupakan state akhir (F) akan ditandai dengan lingkaran ganda, sedang state yang lain menggunakan lingkaran tunggal.

Sedangkan tabel transisi merupakan representasi dengan bentuk tabel dari fungsi  $\delta$  yang mengambil dua argumen atau lebih dan menghasilkan suatu nilai. Baris menunjukkan *state*, sedangkan kolom menunjukkan suatu input.

**Tabel 1. Contoh Tabel Transisi**

$\delta$	0	1
$\rightarrow q_1$	$q_1$	$q_2$
$q_2$	$q_3$	$q_2$
$* q_3$	$q_1$	$q_2$

Tabel di atas menjelaskan bahwa:

1. Simbol pada kolom sebelah kiri adalah *state*;
2. Simbol pada baris paling atas adalah simbol input.
3. Simbol yang berada "dalam" tabel merupakan fungsi transisi.
3. Simbol yang berada "dalam" tabel merupakan fungsi transisi.
5. Simbol (\*) pada kolom sebelah kiri menunjukkan state final.

## METODE

Metode penelitian yang diterapkan dalam Penerapan Layanan *Cloud Server* secara *self-service* menggunakan Model *Finite State Automata* disajikan pada gambar di bawah ini, antara lain pada tahap pertama adalah Pemahaman Proses Bisnis, pada tahap ini dilakukan identifikasi pada *flowchart* alur layanan.

Pada tahap kedua adalah Perancangan Siklus Layanan, pada tahap ini dirancang usulan siklus layanan *cloud server* dengan sistem terotomasi. Pada tahap ketiga adalah Perancangan Diagram State, pada tahap ini dilakukan perancangan menggunakan model FSA. Pada tahap terakhir adalah Desain Rancangan Sistem, pada tahap ini didesain rancangan sistem berupa tampilan sistem layanan *cloud server*.



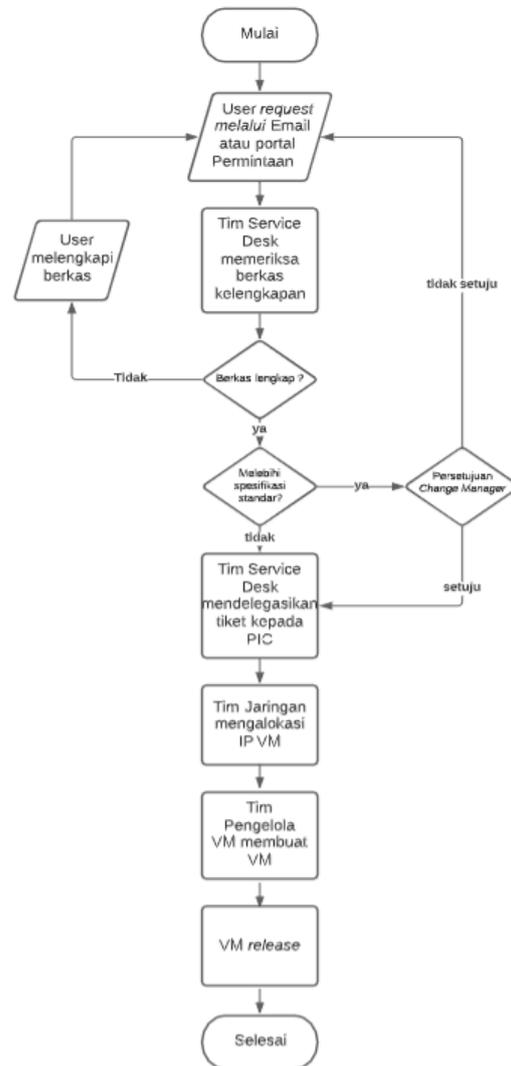
**Gambar 2. Metode Penelitian**

**HASIL DAN PEMBAHASAN**

**A. Pemahaman Proses Bisnis**

Permintaan layanan *cloud server* pada Pusintek masih dilakukan secara manual, pengguna melakukan *request* melalui *e-mail* dan tiket. Setelah permintaan masuk ke Tim Pengelola VM, maka VM dibuat oleh Tim Pengelola VM berdasarkan spesifikasi yang sudah ditentukan oleh pengguna. Proses permintaan layanan *cloud server* tersebut merupakan salah satu proses bisnis yang rutin dan berulang. Namun dibutuhkan ketersediaan Sumber Daya Manusia dan ketersediaan waktu dari Tim Pengelola VM untuk melakukan pembuatan VM dimaksud. Sehingga

dipandang perlu untuk dapat membuat perancangan sistem yang membantu dan mempermudah tugas Tim Pengelola VM serta dapat mempercepat waktu pembuatan VM itu sendiri bagi para pengguna. Sistem dimaksud memungkinkan bagi pengguna untuk dapat membuat VM nya masing-masing sesuai dengan sistem operasi dan spesifikasi yang dibutuhkan secara mandiri.



**Gambar 3. Flowchart Layanan Cloud Server**

Pada *flowchart* di atas menggambarkan proses penyiapan layanan *cloud server* dari mulai proses permintaan pengguna hingga tersedianya

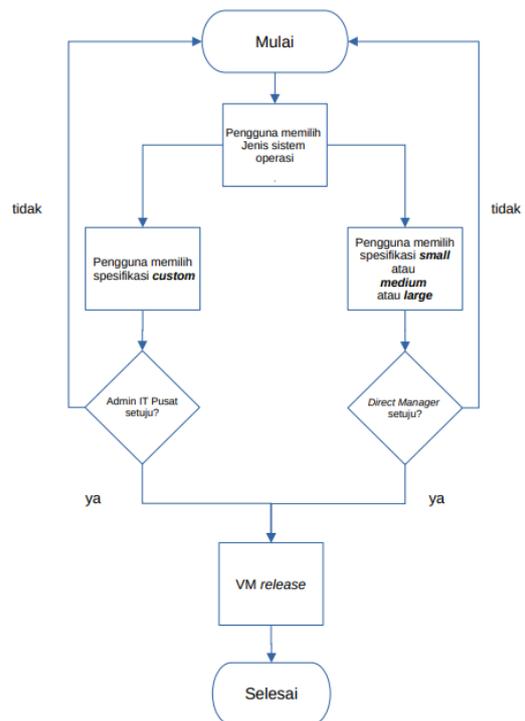
VM tersebut. Diketahui bahwa pada saat pengguna mengajukan permintaan layanan cloud tersebut ditempuh beberapa proses yang cukup panjang, sehingga membutuhkan waktu yang cukup lama dalam pemenuhan layanan dimaksud.

Adapun proses layanan *cloud server* saat ini, adalah sebagai berikut:

- 1 Pada langkah awal pengguna mengajukan *request* melalui *e-mail* atau melalui portal permintaan untuk mendapatkan tiket, kemudian Tim Service Desk memeriksa berkas kelengkapan administratif, apabila lengkap, maka permintaan tersebut akan dilakukan pemeriksaan terkait spesifikasi server, namun apabila tidak lengkap, maka permintaan tersebut akan ditolak.
- 2 Saat dilakukan pemeriksaan terkait spesifikasi server, dilakukan identifikasi apakah permintaan melebihi standar spesifikasi atau tidak. Apabila melebihi standar, maka diperlukan persetujuan *Change Manager* untuk pemenuhan layanan, apabila sesuai standar, maka Tim Service Desk akan mendelegasikan tiket kepada PIC terkait, yaitu Tim Jaringan.
- 3 Tim Jaringan mengalokasikan *Internet Protocol* (IP) Address yang dapat digunakan oleh VM.
- 4 Tim Pengelola VM melakukan pembuatan VM sesuai dengan kriteria (sistem operasi dan spesifikasi) yang diinginkan oleh *pengguna*.
- 5 Tim Pengelola VM melakukan *release* VM.
- 6 Tim Pengelola VM melakukan *power on* VM sehingga dapat VM dimaksud dapat diakses oleh *pengguna*.

B. Perancangan Siklus Layanan

Sebagaimana pada gambar 3 diketahui bahwa proses pemberian layanan *cloud server* dilakukan dengan banyak langkah dan membutuhkan ketersediaan waktu serta sinergi yang baik antar tim (Tim Service Desk, Tim Jaringan, dan Tim Pengelola VM). Berikut adalah *flowchart* proses layanan *cloud server* yang penulis usulkan. Pada *flowchart* dimaksud terdapat beberapa proses yang dapat di-*substitusi* menjadi proses yang terotomatisasi sehingga dapat dihilangkan pada proses ini. Namun, untuk menjaga *good governance* proses tersebut dibutuhkan persetujuan oleh *Direct Manager* (atasan langsung) atau Admin IT Pusat, hal ini dibutuhkan untuk menghindari adanya pembuatan VM untuk kepentingan di luar kedinasan



**Gambar 4. Flowchart Layanan Cloud Server yang diusulkan**

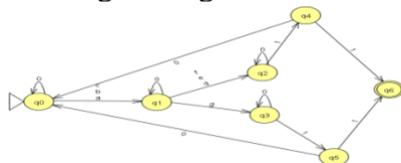
Pada gambar 4 menggambarkan bagaimana proses layanan *cloud server* yang dilakukan secara *self-service* oleh pengguna, dengan meminimalisasikan proses manual yaitu peran Tim Jaringan dan Tim Pengelola VM, diharapkan

dengan menggunakan *flowchart* ini dapat menghemat waktu pembuatan VM dan meminimalisasikan potensi kesalahan dalam membuat VM. Adapun prosesnya adalah sebagai berikut:

1. Pada langkah awal pengguna login portal layanan *cloud*.
2. Pengguna dapat memilih salah satu jenis sistem operasi VM, yaitu Windows Server, Ubuntu Server, dan Centos Server.
3. Pengguna dapat memilih salah satu jenis spesifikasi server, yaitu *small*, *medium*, dan *large*.
4. Pengguna yang memilih jenis spesifikasi antara *small*, *medium*, dan *large* perlu mendapatkan persetujuan *Direct Manager* untuk melanjutkan *state* berikutnya. Sedangkan pengguna yang memilih jenis spesifikasi *custom* perlu mendapatkan persetujuan Admin IT Pusat.
5. Setelah mendapatkan persetujuan, Vm dapat *release* dan dapat diakses oleh pengguna secara langsung menggunakan *protocol Remote Desktop Protocol (RDP)* maupun *Secure Shell (SSH)*.

Setelah menentukan perancangan siklus layanan, maka diperlukan untuk dilakukan perencanaan diagram state berupa FSA. Pada diagram tersebut akan diterangkan secara rinci alur dari perancangan sistem yang akan didesain, sehingga akan memberikan gambaran sistem yang lengkap dan komprehensif.

C. Perancangan Diagram State



**Gambar 5. Diagram State**

Diagram FSA menggambarkan sebuah model mesin yang dapat memperoleh input dan memberikan output dengan jumlah state yang berhingga serta dapat berpindah dari satu state ke state berikutnya sesuai input yang diperoleh dari fungsi-fungsi transisinya (Erni, et al., 2020). Berdasarkan Gambar Perancangan Diagram State tersebut diketahui bahwa:

$$Q = \{q_0, q_1, q_2, q_3, q_4, q_5, q_6\}$$

$$\Sigma = \{a, b, c, d, e, f, g, 0, 1\}$$

$$S = \{q_0\}$$

$$F = \{q_6\}$$

**Tabel 2. Tabel Himpunan State**

State	Deskripsi
q0	Masuk ke portal layanan <i>cloud server</i>
q1	Menampilkan sistem operasi yang dipilih
q2	Menampilkan spesifikasi <i>small</i> atau <i>medium</i> atau <i>large</i>
q3	Menampilkan spesifikasi <i>custom</i>
q4	Menampilkan persetujuan <i>Direct Manager</i>
q5	Menampilkan persetujuan Admin IT Pusat
q6	VM <i>release</i>

**Tabel 3. Tabel Keterangan Transisi**

Transisi	Deskripsi
a	Pilih Windows Server
b	Pilih Ubuntu Server
c	pilih Centos Server
d	Pilih spesifikasi <i>small</i>
e	Pilih spesifikasi <i>medium</i>
f	Pilih spesifikasi <i>large</i>
g	Pilih spesifikasi <i>custom</i>
0	Tidak lanjut
1	Lanjut

Pada gambar 5, *state* q0 diuraikan bahwa pengguna login ke portal layanan *cloud* untuk dapat memilih salah satu jenis sistem operasi VM secara mandiri, yaitu Windows Server, Ubuntu Server,

dan Centos Server. Pada *state* q1 menampilkan jenis sistem operasi yang dipilih, kemudian pengguna dapat memilih salah satu jenis spesifikasi, yaitu *small, medium, large*, atau *custom*. Pada pengguna yang memilih spesifikasi antara *small, medium*, atau *large* akan berlanjut ke *state* q4 untuk mendapatkan persetujuan *Direct Manager*.

Pada pengguna yang memilih spesifikasi *custom* akan berlanjut ke *state* q5 untuk mendapatkan persetujuan dari Admin IT Pusat. Apabila mendapatkan persetujuan, maka akan berlanjut ke *state* q6 yaitu VM akan *release* secara otomatis. Fungsi transisi dapat dijelaskan pada tabel 4, berikut ini:

**Tabel 4. Tabel Transisi**

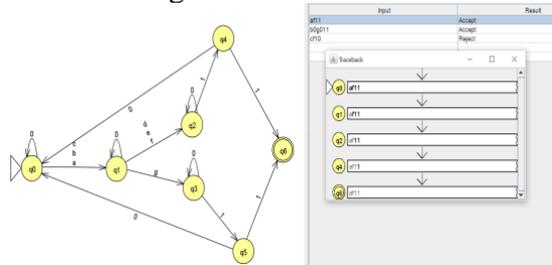
$\delta$	a	b	c	d	e	f	g	0	1
q0	q1	q1	q	-	-	-	-	q0	-
q1	-	-	q	q2	q2	q	q1	-	-
q2	-	-	-	-	-	-	-	q2	q
q3	-	-	-	-	-	-	-	q3	q
q4	-	-	-	-	-	-	-	q0	q
q5	-	-	-	-	-	-	-	q0	q
q6	-	-	-	-	-	-	-	-	-

Pengujian dilakukan untuk menguji apakah formula yang dihasilkan telah memenuhi kondisi dari FSA atau tidak. Pengujian input *string* pada diagram FSA dilakukan dengan menggunakan *software* JFLAP versi 7.1 (Rodger & Finley, 2006). Sistem akan mengikuti alur dari proses pemilihan sistem operasi dan spesifikasi yang telah dipilih oleh pengguna. FSA digunakan untuk membaca simbol masukan yang diberikan dari *state* awal sampai berakhirnya proses sehingga dapat dikenali oleh portal layanan *cloud*.

**Tabel 4. Contoh pengujian masukan *string* dan keluaran akhir pada aplikasi JFLAP**

Masukan <i>string</i>	Keluaran akhir
<b>af11</b>	VM windows server dan spesifikasi <i>large</i>
<b>b0g011</b>	VM Ubuntu server dan spesifikasi <i>custom</i>
<b>cf10</b>	<i>Reject</i> , dikarenakan q0 bukan <i>state</i> akhir

Pada Gambar 6 menunjukkan proses input *string* “af11”: pemilihan VM oleh pengguna berupa sistem operasi windows server dan spesifikasi *small* yang mendapat persetujuan oleh *Direct Manager*. Fungsi transisinya adalah sebagai berikut:



**Gambar 6. Pengujian FSA pada aplikasi JFLAP**

Gambar menunjukkan proses pengujian FSA dengan contoh input *string* “af11” pada mesin, sehingga proses yang terjadi adalah sebagai berikut:

- diawali dari *state* q0
- membaca input a, melakukan transisi dari q0 ke q1
- membaca input f, melakukan transisi dari q1 ke q2
- membaca input 1, melakukan transisi dari q2 ke q4
- membaca input 1, melakukan transisi dari q4 ke q6
- input *string* “af11” diterima (*accepted*) karena berakhir di q6 dengan memberikan keluaran berupa VM dengan sistem operasi Windows Server dengan spesifikasi *large*.

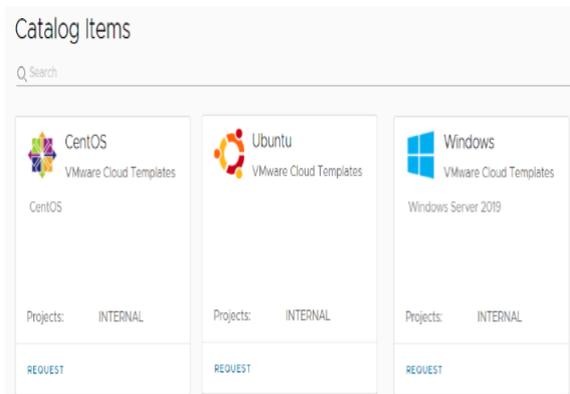
Demikian juga dengan pengujian FSA yang menggunakan *string*

sebagaimana pada tabel 4 menunjukkan hasil yang konsisten.

#### D. Desain Perancangan Sistem

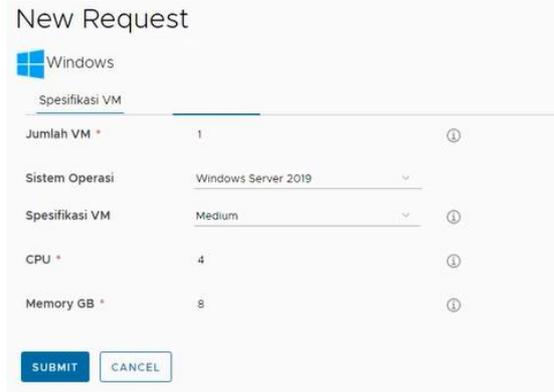
Pada desain perancangan sistem dibutuhkan sebuah sistem informasi yang dapat memungkinkan pengguna dalam melakukan pemilihan sistem operasi, spesifikasi VM, dan sistem persetujuan *Direct Manager* dan Admin IT Pusat. Dalam hal ini penulis menggunakan *software* vRealize Automation untuk membantu membuat desain perancangan sistem dan membuat *platform* yang terotomatisasi agar mudah dipahami dan digunakan oleh pengguna (Choudhari & Sasankar, 2021).

Setelah pengguna mendapatkan tiket layanan permintaan *cloud server*, pengguna dapat membuka portal layanan *cloud server*. Tampilan portal layanan *cloud server* yang penulis usulkan adalah sebagai berikut:



**Gambar 7. Portal layanan *cloud server***

Gambar 7 merupakan tampilan awal pada portal layanan *cloud server*, pengguna dapat memilih sistem operasi apa yang ingin digunakan. Terdapat 3 (tiga) pilihan, yaitu Centos Server, Ubuntu Server, dan Windows Server.



**Gambar 8. Pilih spesifikasi VM**

Gambar 8 merupakan tampilan menu pengguna untuk melakukan pemilihan spesifikasi VM, pengguna dapat memilih 4 (empat) jenis spesifikasi, yaitu *small*, *medium*, *large* dan *custom*.



**Gambar 9. Persetujuan dari *Direct Manager* atau Admin IT Pusat**

Gambar 9 merupakan tampilan menu yang diakses oleh *Direct Manager* dan Admin IT Pusat untuk melakukan persetujuan atas VM yang di-*create* oleh pengguna. *Direct Manager* dapat melakukan *review* untuk permintaan VM dengan spesifikasi *small*, *medium*, dan *large*. Sedangkan Admin IT Pusat dapat melakukan *review* untuk permintaan VM dengan spesifikasi *custom*.

Berdasarkan usulan kami di atas, pengguna akan lebih fleksibel dalam menggunakan layanan *cloud server* secara *self-service* tanpa mengorbankan sisi *Good-Governance*, yang dibuktikan dengan fitur persetujuan *Direct Manager* atau Admin IT Pusat, yang dibuktikan dengan Gambar 9.

## KESIMPULAN

FSA sangat sesuai untuk menggambarkan alur siklus layanan *cloud server*, dengan menggunakan 7 (tujuh) *state*, serta fungsi transisi yang berhasil menerima berbagai

kemungkinan *string input*. Penerapan konsep FSA pada alur siklus layanan *cloud server* ini digambarkan dengan cara memberikan sebuah inputan yang diberikan kepada mesin sebagai sebuah bahasa yang dapat dikenali oleh sistem. Kemudian mesin akan mengeluarkan *output* berupa VM sesuai dengan jenis sistem operasi dan spesifikasi VM sesuai pilihan pengguna. Dengan adanya alur dan desain perancangan sistem sebagaimana yang diusulkan, diharapkan dapat memudahkan dan menghemat waktu pengguna dalam memanfaatkan layanan *cloud server*, serta dapat mengurangi potensi kesalahan konfigurasi VM

#### DAFTAR PUSTAKA

- Ashari, A., Setiawan, H., Ilmu, J., Mipa, F., & Mada, U. G. (2011). *Cloud Computing: Solusi ICT?* 3(2), 336–345.
- Choudhari, N. V., & Sasankar, A. B. (2021, February). Architectural Vision of Cloud Computing in the Indian Government. In *2021 International Conference on Innovative Trends in Information Technology (ICITIIT)* (pp. 1-7). IEEE.
- Darmayunata, Y., Siswati, L., & Aryanto, A. (2021). Pekanbaru City Livestock Marketing With Android-Based Applications. *Journal of Applied Engineering and Technological Science (JAETS)*, 3(1), 40–52. <https://doi.org/10.37385/jaets.v3i1.292>
- Desvia, Y. F., Rosadi, R., Frieyadie, F., Haryanti, T., & Gata, W. (2021). Penerapan Finite State Automata Pada Vending Machine Parfum Laundry Pakaian. *BINA INSANI ICT JOURNAL*, 8(2), 103-112.
- Erni, E., Titiani, F., Putri, S. A., & Gata, W. (2020). Penerapan Konsep Finite State Automata Pada Aplikasi Simulasi Vending Machine Jamu Tradisional. *Jurnal Informatika*, 7(2), 141-147.
- Kurniawan, E. (2015). Penerapan Teknologi Cloud Computing Di Universitas Studi Kasus : Fakultas Teknologi Informasi UKDW. *Eksis*, 08(01), 29–36.
- Muhammad, R., Gata, W., Novitasari, H. B., Kurniawati, L., & Rahayu, S. (2022). Penerapan Finite State Automata Pada Desain Vending Machine Masker Dan Hand Sanitizer. *Jurnal Informasi dan Komputer*, 10(1), 21-28.
- Pusintek. (2020). *Service Catalog 2020* (p. 15).
- Rizki, M., Ghifari, A., Hui, W. L., Permata, E. G., Siregar, M. D., Umam, M. I. H., & Harpito, H. (2021). Determining Marketing Strategy At LPP TVRI Riau Using SWOT Analysis Method. *Journal of Applied Engineering and Technological Science (JAETS)*, 3(1), 10–18. <https://doi.org/10.37385/jaets.v3i1.276>
- Rodger, S. H., & Finley, T. W. (2006). *JFLAP: An Interactive Formal Languages and Automata Package*. 220. [http://books.google.co.uk/books?id=494hTkZ\\_gu4C](http://books.google.co.uk/books?id=494hTkZ_gu4C)
- Rumetna, M. S. (2018). Pemanfaatan Cloud Computing Pada Dunia Bisnis: Studi Literatur. *Jurnal Teknologi Informasi Dan Ilmu Komputer*, 5(3), 305. <https://doi.org/10.25126/jtiik.2018.53595>
- Sipser, M. (2006). *Introduction-To-The-Theory-Of-Computation-Second-Edition* (Second Edi). Thomson Course Technology.
- Utami, D. N. (2018). *Alasan Startup Lebih Suka Cloud*. <https://teknologi.bisnis.com/read/>

20180516/105/796097/alasan-  
startup-lebih-suka-cloud

Wildana, F. (2017). Implementasi Cloud Computing Di Beberapa Instansi Pemerintahan. *Masyarakat Telematika Dan Informasi: Jurnal Penelitian Teknologi Informasi dan Komunikasi*, 8(2), 97-108.