**Sistem Manajemen Layanan Web Berbasis Platform as a Service (PaaS) dengan API Openstack**

Ida Bagus Rathu Eka Surya Wibawa1, I Dewa Made Bayu Atmaja Darmawan2, I Komang Ari Mogi3

123 Program Studi Tekhnik Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Udayana

Jl. Kampus Bukit Jimbaran (80361), Badung, Bali, Indonesia

1guzekawibawa@gmail.com

2 dewabayu@cs.unud.ac.id

3 arimogi@cs.unud.ac.id

***Abstrak***

*Cloud computing merupakan teknologi yang saat ini mulai berkembang dalam banyak aktivitas teknologi informasi. Cloud computing merupakan model komputasi yang semua sumber daya yang ada dalam layanan cloud dijalankan dengan media jaringan internet. Dengan adanya cloud computing memudahkan para developer dalam melakukan komputasi tanpa harus melakukan instalasi aplikasi pada komputer, developer hanya perlu mengaksesnya melalui internet. Cloud computing memiliki beberapa fasilitas yang dapat dipilih oleh developer sesuai kebutuhan developer seperti Infrastructure as a Service(IaaS), Platform as a Service(PaaS). Serta Software as a Service(SaaS).*

*Pengembangan Platform as a Service sebagai salah satu teknologi cloud computing yang dapat digunakan oleh pengembang aplikasi untuk mengembangkan aplikasi yang akan dibuat tanpa perlu menyediakan infrastruktur, database, framework aplikasi dan lain sebagainya serta bersifat dinamis. Dalam pengembangan layanan cloud yang dikelola oleh seorang sistem administrator atau developer, tugas menginstalasi dan menkonfigurasi sistem pada server maupun software aplikasi dilakukan dengan otomatis dengan menggunakan platform otomatisasi sistem linux untuk efisiensi waktu dan memanagemen penggunanya dengan lebih mudah.*

***Kata kunci : Cloud Computing, Platform as a Service, automation, ansible, API, Openstack***

1. **Pendahuluan**

Pada era teknologi seperti sekarang ini, perkembangan teknologi sangat meningkat pesat, dimana para pengembang aplikasi saat ini dengan mudah membangun sebuah web server yang yang dapat dikerjakan secara bersama-sama dengan menggunakan komputasi awan ( Cloud Computing). Sebelum teknologi cloud computing ditemukan pengembang aplikasi berbasis web lebih sering mengembangkan aplikasi web secara lokal di perangkat komputer. Membuat client dari pengembang aplikasi web tidak dapat memonitoring atau menilai sejauh mana aplikasi web yang di buat oleh pengembang aplikasi web telah selesai, menyebabkan kurangnya dalam memanajemen aplikasi web. Serta pada proses perilisan aplikasi web pun membutuhkan waktu yang tidak sebentar, banyak hal yang butuh diperhitungkan seperti penyimpanan data yang besar, ketersediaan domain, data processing dan masih banyak lainnya.

Menurut Ricky W. Griffin menyatakan bahwa manajemen adalah sebuah proses pengorganisasian, pengkoordinasian, perencanaan, dan pengontrolan sumber daya agar dapat mencapai sasaran (goals) secara efisien dan efektif. Efektif sendiri sebuah tujuan mampu dicapai sesuai dengan apa yang telah direncanakan. Perlunya memanajemen aplikasi web yang dibuat untuk memudahkan pengembang aplikasi beserta tim pengembang lainnya dapat saling meninjau kebutuhan aplikasi web yang di buat secara bersama – sama walau berada di lokasi yang berbeda.

Konsep komputasi awan ini sudah banyak menarik minat industri digital maupun pendidikan. Solusi berbasis cloud sepertinya menjadi kunci bagi organisasi teknologi informasi yang memiliki permasalahan dengan keterbatasan anggaran (Teng & Magoules,2010). Komputasi awan merupakan paradigma yang baru dalam komputasi terdistribusi, menyajikan banyak konsep teknologi, ide, serta arsitektur yang disajikan secara service-oriented. Pada pemanfaatanya komputasi awan mengubah cara bagaimana layanan informasi disediakan maupun di sebarkan. Dengan memanfaatkan teknologi komputasi awan dalam memanajemen aplikasi web, developer / pengembang aplikasi web bekerja secara efektif dan efisien dari segi waktu maupun perangkat. Saat ini sudah banyak penyedia layanan cloud computing untuk para pengembang aplikasi seperti Google Cloud Platform, Amazon Web Services, Alibaba Cloud Server, bahkan untuk di Indonesia ada Biznet GIO cloud.

1. **Metode Penelitian**

Bagian ini akan menjelaskan mengenai langkah – langkah yang akan dilakukan dalam merancang sistem manajemen layanan web berbasis Platform as a Service (PaaS) dengan API openstack

1. **Pengumpulan Data**

Pada tahap ini dilakukan analisis kebutuhan sistem, analisis kebutuhan sistem meliputi data yang digunakan, pembelajaran dari referensi yang sudah ada dan perangkat yang digunakan baik perangkat lunak maupun perangkat keras:

Tahapan pertama yang dilakukan dalam penelitian ini adalah mengidentifikasi permasalahan. Tahap ini merupakan tahap yang paling penting dalam penelitian karena jalannya penelitian didasarkan atas permasalahan yang terjadi. Setelah menentukan masalah yang terjadi, tahapan yang diperlukan selanjutnya adalah menentukan rumusan masalah dan tujuan yang ingin dicapai dalam penenlitian. Pada penelitian ini identifikasi permasalahan dilakukan dengan menggunakan teknik observasi, dari teknik ini maka akan dapat diketahui mengenai keluhan – keluhan yang ada di lapangan.

Tahap kedua yang dilakukan dalam metodelogi penelitian ini adalah studi literatur. Studi literatur dilakukan dengan mengambil literatur – literatur pendukung dari jurnal – jurnal ilmiah, baik jurnal dalam negeri ataupun jurnal luar negeri dan dari beberapa buku. Dalam studi literatur ini, penulis mencari sumber terkait permasalahan – permasalahan yang perlu menjadi perbaikan dalam penelitian selanjutnya..

1. **Arsitektur Kerja Sistem**



Gambar 1. Rancangan Implementasi Sistem

1. **Alur Proses**
2. Web dan database

Database digunakan untuk menyimpan informasi pengguna serta informasi pada VPS yang dimiliki oleh setiap sysadmin.

1. Web dan framework flask REST API

Flask REST API digunakan untuk menghubungkan web front-end dengan back-end. Dimana back-end dibuat dengan menggunakan framework flask dari python, selanjutnya dibuat API tersendiri agar web front-end dapat mengirim dan menerima informasi serta konfigurasi yang dilakukan yang nantinya akan diproses oleh back-end server.

1. Database dan ansible API

Hubungan database dengan ansible API akan mengambil informasi developer berupa username, email dan password yang nantinya akan dimasukan pada konfigurasi dalam VPS yang selanjutnya sebagai super amin pada VPS yang dibuat.

1. framework flask REST API dan asible API

Hubungan framework flask REST API dengan asible API. Ketika user melakukan konfigurasi pada web utama akan dikirim melalui REST API flask selanjutnya konfigurasi tersebut akan digunakan oleh API ansible untuk mengkonfigurasi VPS yang dibuat.

1. framework flask dan openstack API

Hubungan framework flask dan openstack API adalah pada flask dikonfigurasi untuk dapat terhubung dengan server devstack, dimana server openstack yang mengelola dalam membuat sebuah VPS.

1. openstack API dan server

API openstack digunakan untuk terhubung dengan server back-end yang dibuat dengan framework flask. Untuk dapat mengembangkan aplikasi openstack dari bahasa pemrograman yang berbeda.

1. Ansible API dan Virtual Private Server

hubungan ansible API dengan VPS adalah ansible akan mengirimkan konfigurasi pada VPS melalui SSH yang yang dibuat. Selanjutnya pada VPS akan melakukan konfigurasi yang diperlukan untuk membangun sebuah web aplikasi secara otomatis. Developer hanya perlu memilih keperluan yang ada menu web front-end.

1. server dan Virtual Private Server

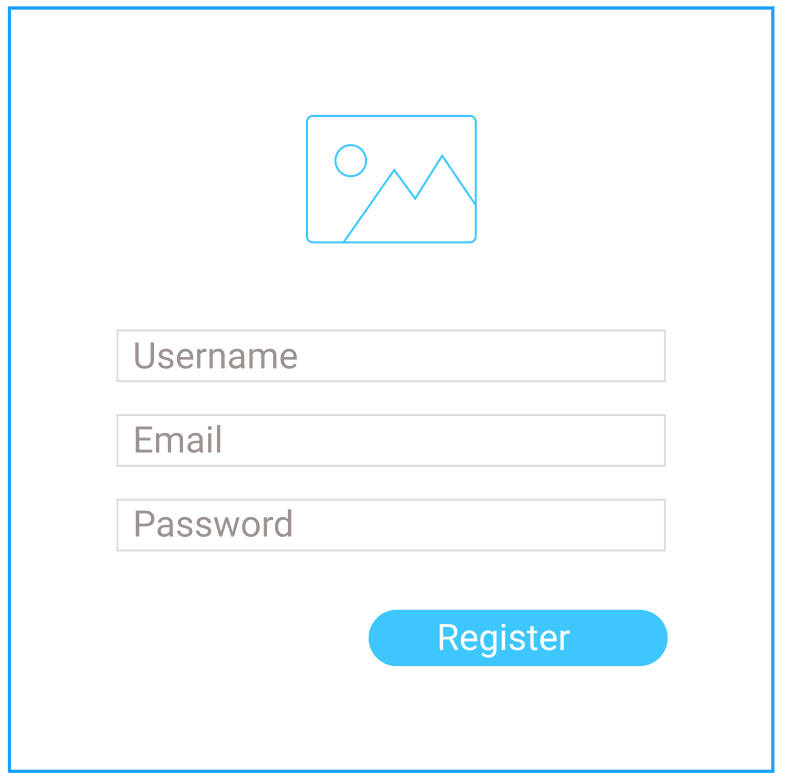
hubungan antara server dan VPS(Virtual Private Server) adalah server mengelola dan memberikan resource pada VPS serta mengatur konektivitas setiap VPS yang dimiliki

1. **Desain Antarmuka Sistem**



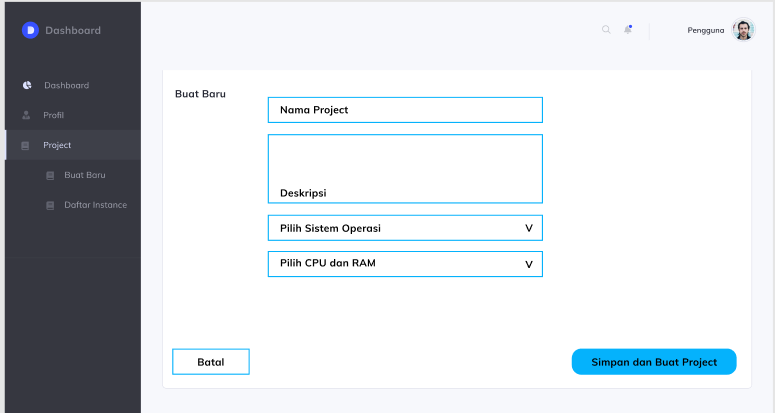
Gambar 2. Rancangan Landing Page

Gambar 2 merupakan rancangan landing page yang berisi informasi dari sistem cloud yang di kembangkan, serta berisi informasi layanan dan fitur yang diberikan kepada pengembang aplikasi yang akan menggunakan sistem cloud.



Gambar 3. Rancangan form pendaftaran

Pada gambar 3 merupakan rancangan sistem untuk dapat masuk ke dalam sistem cloud, pengembang aplikasi web harus melakukan pendaftaran, ini dimaksudkan untuk mempermudah pengelolaan sistem virtual server cloud.



Gambar 4. Rancangan membuat instance server baru

Pada gambar 4 merupakan rancangan tampilan dalam membangun sebuah virtual server cloud, pengembang aplikasi web hanya perlu memilih kebutuhan server yang diinginkan sesuai kebutuhan aplikasi yang dibuat seperti sistem operasi, jumlah unit processing, dan kapasitas penyimpanan.

1. **Hasil dan Diskusi**

Pada penelitian ini pengujian dilakukan dengan menggunakan K-Fold Cross Validation. Pengujian dilakukan sebanyak lima kali dan dicari rata-rata dari akurasi hasil pengujian tersebut.

Tabel 1. Tabel Hasil Pengujian

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | p1 | p2 | p3 | p4 | p5 | Rata – rata Akurasi |
| k1 | 84 | 81 | 80 | 83 | 82 | 82 |
| k3 | 86 | 86 | 87 | 85 | 85 | 85,8 |
| k5 | 85 | 87 | 86 | 86 | 85 | 85,8 |
| k7 | 85 | 84 | 83 | 84 | 86 | 84,4 |
| k9 | 83 | 83 | 84 | 84 | 82 | 83,2 |
| k11 | 84 | 83 | 82 | 83 | 85 | 83,4 |
| k13 | 84 | 86 | 83 | 84 | 84 | 84,2 |
| k15 | 85 | 86 | 83 | 84 | 84 | 84,4 |
| k17 | 87 | 85 | 85 | 85 | 87 | 85,8 |
| k19 | 85 | 86 | 85 | 83 | 86 | 85 |

Gambar 5. Grafik Rata-rata Akurasi

Dari lima kali pengujian yang dilakukan, parameter K sangat mempengaruhi akurasi dari aplikasi. Gambar 5 memperlihatkan grafik dari rata – rata akurasi dimana rata – rata akurasi dari setiap k yaitu sebagai berikut : k = 1 sebesar 82 %, k = 3 sebesar 85,8 %, k = 5 sebesar 85,8 %, k = 7 sebesar 84,4 %, k = 9 sebesar 83,2 %, k = 11 sebesar 83,4 %, k = 13 sebesar 84,2 %, k = 15 sebesar 84,4 %, k = 17 sebesar 85,8 %, k = 19 sebesar 85 %. Dari grafik diatas nilai K dengan rata – rata akurasi tertinggi terdapat pada k = 3, k = 5, dan k = 17, jadi disimpulkan nilai k terbaik yang dapat digunakan dalam sistem “Pendeteksian Malware Android dengan Pendekatan Permissions dan API Call menggunakan K-Nearest Neighbor (KNN)” yaitu k = 3 dikarenakan semakin sedikit nilai k maka semakin cepat proses klasifikasi berlangsung.

1. **Kesimpulan**

Berdasarkan penelitian yang telah dilakukan, dapat diambil beberapa kesimpulan sebagai berikut :

1. Untuk mendeteksi malware dengan menggunakan K Nearest Neighbor, tahap pertama yang dilalui adalah preprocessing yaitu pencarian fitur berupa Permissions dan API Calls yang nantinya akan digunakan dalam proses klasifikasi. Selanjutnya proses klasifikasi dimana pertama – tama kita harus menentukan jarak dari setiap dataset menggunakan Binary Jackard Distance karena bentuk fitur yang digunakan berupa binary 1 dan 0, kemudian menentukan kelas sesuai dengan jumlah K yang diinputkan.
2. Berdasarkan hasil pengujian sebanyak lima kali, didapatkan rata – rata akurasi yaitu k = 1 sebesar 82 %, k = 3 sebesar 85,8 %, k = 5 sebesar 85,8 %, k = 7 sebesar 84,4 %, k = 9 sebesar 83,2 %, k = 11 sebesar 83,4 %, k = 13 sebesar 84,2 %, k = 15 sebesar 84,4 %, k = 17 sebesar 85,8 %, k = 19 sebesar 85 %.
3. Dari rata – rata pengujian tertinggi sebesar 85,8 % dapat disimpulkan nilai k terbaik yang dapat digunakan dalam sistem “Pendeteksian Malware Android dengan Pendekatan Permissions dan API Call menggunakan K-Nearest Neighbor (KNN)” yaitu k = 3 dikarenakan proses klasifikasi dengan k = 3 lebih cepat dibandingkan dengan k = 5 dan k = 17.

**Referensi**

|  |  |
| --- | --- |
| [1] | M. Blazek, "Google Android APK application package file format description," 14 Februari 2017. [Online]. Available: https://www.file-extensions.org/article/android-apk-file-format-description. [Accessed 1 Maret 2018]. |
| [2] | N. Elenkov, Android Security Internals, San Francisco: William Pollock, 2015. |
| [3] | F. Gorunescu, Data Mining Concepts, Models, and Techniques, Verlag Berlin Heidelberg : Springer, 2011. |
| [4] | J. Han, M. Kamber and J. Pei, Data Mining Concepts and Techniques third edition, United States of America: Morgan Kaufmann, 2012. |
| [5] | N. Minihane, F. Moreno, E. Peterson, R. Samani, C. Schmugar, D. Sommer and B. Sun, "McAfee Labs Threat Report December 2017," McAfee, 2017. |
| [6] | F. A. Narudin, A. Feizollah, N. B. Anuar and A. Gani, "Evaluation of machine learning classifiers for mobile malware," 2014. |
| [7] | J. Oberheide and C. Miller, "Dissecting the Android Bouncer," *SummerCon 2012,* 2012. |
| [8] | E. Prasetyo, Data Mining Mengolah Data Menjadi Informasi Menggunakan MATLAB, Yogyakarta: ANDI Yogyakarta, 2014. |
| [9] | M. Qiao, A. H. Sung and Q. Liu, "Merging Permission and API Features for Android Malware Detection," *IIAI International Congress on Advanced Applied Informatics,* pp. 566-571, 2016. |
| [10] | A. F. Ridlo and A. Karima, "Analisis Implementasi Metode Klasifikasi Bayes untuk Deteksi Malware Android," pp. 1-11, 2016. |
| [11] | M. Sikorski and A. Honig, Practical Malware Analysis, San Francisco: William Pollock, 2012. |
| [12] | K. Teknomo, "Jaccard's Coefficient," 2015. [Online]. Available: https://people.revoledu.com/kardi/tutorial/Similarity/Jaccard.html. [Accessed 20 November 2018]. |