**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.

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.

Pada Penelitian sebelumnya, Nishant Kumar Singh dkk, menjelaskan pada perkembangan cloud computing pada sisi infrastruktur menuntut penyebaran layanan yang cepat, pada waktu yang bersamaan kebutuhan penyediaan secara otomatis ikut berperan dalam pengembangan aplikasi melalui komputasi awan. Maka dari itu, dalam Tugas Akhir ini penelitian dilakukan dengan membangun sebuah Platform website berbasis cloud untuk pengembang aplikasi web dalam membangun sebuah website secara otomatasasi untuk meningkatkan efisiensi kinerja tim secara bersamaan mampu menajemen kebutuhan pengembangan aplikasi yang dibuat.

1. **Metode Penelitian**
2. **Pengumpulan Data**
3. **Arsitektur Kerja Sistem**



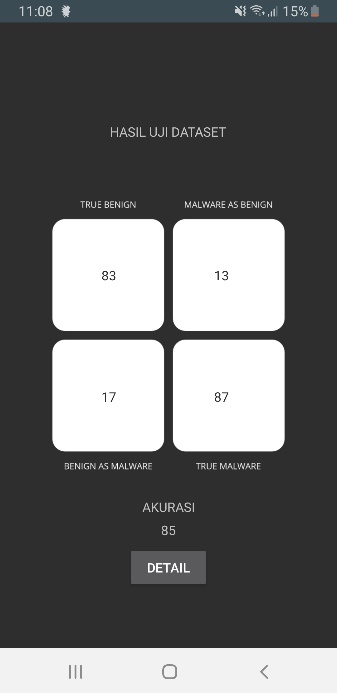
Gambar 1. Rancangan Implementasi Sistem

1. **Alur Proses**
2. **Desain Antarmuka Sistem**



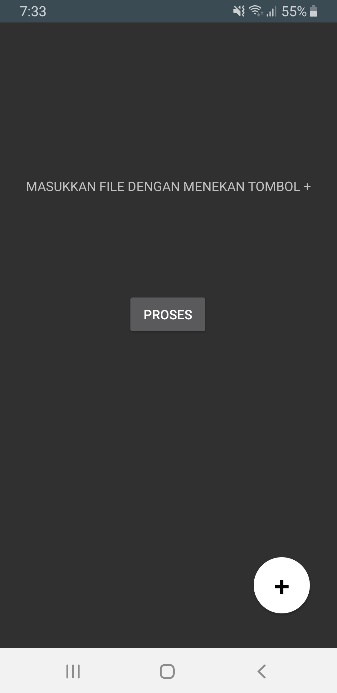
Gambar 2. Tampilan Menu Utama

Pada Menu utama terdapat dua pilihan menu, jika user memilih menu uji dataset maka akan tampil seperti pada gambar 3. Sedangkan jika user memilih menu deteksi, maka akan tampil seperti gambar 4.



Gambar 3. Tampilan Menu Uji Dataset

Menu uji dataset menampilkan hasil dari klasifikasi berupa True Malware, Malware as Benign, True Benign, dan Benign as Malware serta akurasi dan tombol detail yang berisikan detail dari hasil klasifikasi.



Gambar 4. Tampilan Menu Deteksi

Menu deteksi berisi tombol + untuk memilih file APK kemudian tombol proses untuk melakukan proses klasifikasi yang nantinya akan menghasilkan keluaran berupa teks hasil klasifikasi apakah malware tersebut malware atau benign.

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]. |