

Jurnal Informatika Terpadu

Vol. 5 No. 1 Tahun 2019



Jurnal Informatika Terpadu

Jurnal Informatika Terpadu jurnal ilmiah yang berasal dari mahasiswa/i S-1 pada prodi Teknik Informatika dan Sistem Informasi di STT Terpadu Nurul Fikri sebagai salah satu syarat untuk meraih gelar Sarjana (S-1).

Ketua Penyunting (*Editor-in-chief*)

Sirojul Munir, S.Si., M.Kom.
Teknik Informatika
STT Terpadu Nurul Fikri

Dewan Penyunting (*Editorial Board Member*)

Hilmy Abidzar Tawakal, S.T., M.Kom.
Teknik Informatika
STT Terpadu Nurul Fikri

Tiffany Nabarian, S.Kom., M.T.I.
Teknik Informatika
STT Terpadu Nurul Fikri

Suhendi, S.T., M.MSI.
Sistem Informasi
STT Terpadu Nurul Fikri

Zaki Imaduddin, S.T., M.Kom.
Teknik Informatika
STT Terpadu Nurul Fikri

Ahmad Rio Ardiansyah, S.Si., M.Si.
Teknik Informatika
STT Terpadu Nurul Fikri

Mitra Bestari (*Reviewer*)

Drs. Rusmanto, M.M.
Sistem Informasi
STT Terpadu Nurul Fikri

Yekti Wirani, S.T., M.T.I.
Sistem Informasi
STT Terpadu Nurul Fikri

Henry Saptono, S.Si, M.Kom.
Teknik Informatika
STT Terpadu Nurul Fikri

Dr. Lukman Rosyidi, S.T., M.M., M.T.
Teknik Informatika
STT Terpadu Nurul Fikri

Amalia Rahmah, S.T., MT.
Sistem Informasi
STT Terpadu Nurul Fikri

Misna Asqia, M.Kom.
Sistem Informasi
STT Terpadu Nurul Fikri

Nurul Janah, S.IIP, M.Hum.
Sistem Informasi
STT Terpadu Nurul Fikri

Penyunting Pelaksana (*Assistant Editors*)

Muh Syaiful Romadhon, S.Kom.
Sistem Informasi
STT Terpadu Nurul Fikri

Fasyikhatun Maidah, S.Kom.
Sistem Informasi
STT Terpadu Nurul Fikri

Jurnal Informatika Terpadu (*Journal of Integrated Informatics*) telah terindeks oleh Google Scholar. Tanggungjawab isi artikel berada di penulis bukan pada penerbit atau editor.

Diterbitkan oleh:

LPPM STT Terpadu Nurul Fikri

Alamat Redaksi dan Distribusi:

Kampus B STT Terpadu Nurul Fikri lantai 3
Jl. Lenteng Agung Raya 20, Jakarta Selatan, DKI Jakarta, 12640
Telp. 021 – 786 3191

Email: lppm@nurulfikri.ac.id

Website: <https://journal.nurulfikri.ac.id/index.php/jit/> dan lppm.nurulfikri.ac.id

Daftar Isi

Analisis dan Implementasi Sistem Informasi Pengelolaan SDM berbasis OrangeHRM di SDIT AL QUDWAH	01
Afifah Sausan, Rusmanto	
Implementasi <i>Network Monitoring System</i> terintegrasi dengan <i>Ticketing System</i> menggunakan Nagios dan osTicket	06
Debi Onas Saputra, Henry Saptano	
Pengujian dan Implementasi Efektivitas Sistem Virtualisasi berbasis Docker <i>Platform</i>	18
Husni Mubarak, Henry Saptano	
Rancang Bangun <i>Website Customer Relationship Management (CRM)</i> pada Modul Akuntansi Studi Kasus CV Esindo Multi Tata	24
Muhammad Rizky Arinugraha Putra, Sirojul Munir	
Rancang Bangun Sistem <i>Monitoring Budget</i> pada <i>Event Organizer</i> berbasis <i>Website</i> menggunakan <i>MVC Framework</i>	30
Muhammad Tarmizi, Sirojul Munir	



ANALISIS DAN IMPLEMENTASI SISTEM INFORMASI PENGELOLAAN SDM BERBASIS ORANGEHRM DI SDIT AL QUDWAH

Afifah Sausan¹, Rusmanto²

^{1,2}Sistem Informasi, Sekolah Tinggi Teknologi Terpadu Nurul Fikri
Jakarta Selatan, DKI Jakarta, Indonesia 12640
afifahsausan96@gmail.com, rusmanto@gmail.com

Abstract

Human Resource Information System (HRIS) is an information system to support the activities of managers in the human resource function. This function was formerly known as the function of the personnel department, which is now renaming the HR function to show that people in the organization are an essential economic resource (Jogiyanto HM, 2005: 249). This study aims to implement an HR Information System using OrangeHRM to document data, especially on employee performance appraisals at SDIT Al-Qudwah, knowing the results of implementing HR Information Systems using OrangeHRM regarding employee performance appraisal SDIT Al-Qudwah to reduce paper use and direct meetings. With the teacher being assessed and knowing the value of the benefits of OrangeHRM for employee performance appraisal at SDIT Al-Qudwah. The stages in this research are a preliminary study, analysis and design, implementation, testing, and concluding. The results of this study are the implementation of OrangeHRM in the HR department of SDIT Al-Qudwah can document data, especially teacher performance appraisal data in the Performance and PIM modules, OrangeHRM can reduce the process of meeting directly with the teacher in question in the performance appraisal process with the number of answers agreeing 64%, reducing paper usage, reducing teacher performance appraisal process time and OrangeHRM is very useful in the teacher performance appraisal process at SDIT Al-Qudwah.

Keywords: *Teacher Performance Assessment, Human Resource Information System, OrangeHRM, SDIT Al-Qudwah*

Abstrak

Human Resource Information System (HRIS) adalah sistem informasi untuk mendukung kegiatan-kegiatan manajer di fungsi sumber daya manusia. Fungsi ini dulunya bernama fungsi departemen personalia, yang sekarang diubah namanya menjadi fungsi SDM untuk menunjukkan bahwa manusia di dalam organisasi adalah sumber daya ekonomis yang penting (Jogiyanto HM, 2005:249). Penelitian ini bertujuan untuk mengimplementasikan Sistem Informasi SDM menggunakan OrangeHRM untuk mendokumentasikan data khususnya pada penilaian kinerja karyawan di SDIT Al- Qudwah, mengetahui hasil dari implementasi Sistem Informasi SDM menggunakan OrangeHRM dalam hal penilaian kinerja karyawan di SDIT Al-Qudwah untuk mengurangi penggunaan kertas dan pertemuan langsung dengan guru yang dinilai, dan mengetahui nilai manfaat dari OrangeHRM untuk penilaian kinerja karyawan di SDIT Al-Qudwah. Tahapan dalam penelitian ini adalah studi pendahuluan, analisis dan perancangan, implementasi, uji coba, dan penarikan kesimpulan. Hasil dari penelitian ini adalah Implementasi OrangeHRM di bagian SDM SDIT Al-Qudwah dapat mendokumentasikan data khususnya data penilaian kinerja guru di dalam modul Performance dan PIM, OrangeHRM dapat mengurangi proses pertemuan langsung dengan guru yang bersangkutan pada proses penilaian kinerja dengan jumlah jawaban setuju 64%, mengurangi penggunaan kertas, mengurangi waktu proses penilaian kinerja guru dan OrangeHRM sangat bermanfaat pada proses penilaian kinerja guru di SDIT Al-Qudwah.

Kata Kunci: *Penilaian Kinerja Guru, Human Resource Information System, OrangeHRM, SDIT Al-Qudwah*

1. PENDAHULUAN

Human Resource Information System (HRIS) adalah sistem informasi untuk mendukung kegiatan- kegiatan manajer di fungsi sumber daya manusia. Fungsi ini dulunya bernama fungsi departemen personalia, yang sekarang diubah namanya menjadi fungsi SDM untuk menunjukkan bahwa manusia di dalam organisasi adalah sumber daya ekonomis yang penting [1]. Salah satu lembaga pendidikan yang memiliki bidang manajemen SDM adalah SDIT Al- Qudwah. Manajemen SDM adalah suatu proses menangani berbagai masalah pada ruang lingkup karyawan, pegawai, buruh, manajer dan tenaga kerja lainnya untuk dapat menunjang aktivitas organisasi atau perusahaan demi mencapai tujuan yang telah ditentukan [2]. Berdasarkan informasi yang telah diberikan oleh Manajer SDM SDIT Al-Qudwah saat ini bagian SDM SDIT Al-Qudwah masih menggunakan metode manual dalam pengelolaannya dan belum menggunakan sistem informasi.

Penilaian kinerja merupakan evaluasi atau penilaian formal terhadap prestasi atau kinerja karyawan. Penilaian tersebut dapat dilakukan secara informal, misal manajer menegur kesalahan karyawan atau memuji karyawan apabila berhasil menyelesaikan suatu pekerjaan dengan baik. Informasi informal seperti ini mempunyai keuntungan karena karyawan dapat memperoleh umpan balik dengan cepat, langsung setelah karyawan melakukan kesalahan atau berhasil menjalankan tugas [3]. Sampai saat ini, metode yang dilakukan oleh bagian SDM untuk menilai kinerja karyawan masih menggunakan rapor manual yang berbentuk buku. Sehingga memungkinkan sesuatu yang tidak diinginkan terjadi seperti rusak, hilang, tercecer, dan lain sebagainya terhadap penilaian kinerja tersebut karena masih berbentuk buku dan tidak memiliki dokumentasi yang baik. Selain itu, Manajer SDM juga harus bertemu langsung dengan guru yang dinilai untuk melakukan pengecekan sesuai dengan penilaiannya. Hal ini cukup menyulitkan Manajer SDM untuk membagi waktu bertemu dengan setiap guru yang dinilai.

Salah satu cara agar penilaian kinerja karyawan di SDIT Al-Qudwah dapat terlaksana dan terdokumentasi dengan baik yaitu dengan cara mengimplementasikan sistem informasi pada bidang SDM. Sistem Informasi SDM adalah sistem terintegrasi yang menyediakan informasi yang digunakan dalam pembuatan keputusan sumber daya manusia. Dengan sistem informasi sumber daya manusia, catatan waktu pegawai dimasukkan ke dalam sistem, dan pengurangan yang sesuai da penyesuaian karyawan lainnya akan tercermin dalam cek gaji akhir [4]. Dengan implementasi tersebut, semua data baik data pribadi guru dan data penilaian terdokumentasi dalam satu sistem. Dengan demikian bagian SDM tidak membutuhkan kertas ataupun buku rapor lagi dan memiliki dokumentasi yang

baik dalam penilaian kinerja karyawan serta dapat memangkas waktu untuk bertemu langsung dengan setiap guru yang dinilai.

Implementasi sistem informasi yang digunakan adalah OrangeHRM. OrangeHRM merupakan solusi *open source* yang ditemukan, dimana solusi ini sangat relevan dengan kebutuhan pengelolaan HR (*Human Resource*) atau sumber daya manusia. OrangeHRM adalah generasi terbaru sistem pengelolaan SDM (Sumber Daya Manusia) berbasis *web*. OrangeHRM akan membantu dalam mengelola aset terpenting perusahaan, yaitu sumber daya manusia [5].

Rumusan masalah dari penelitian ini adalah “Bagaimana OrangeHRM dapat mendokumentasikan data khususnya pada penilaian kinerja karyawan di SDIT Al-Qudwah?”, “Apakah OrangeHRM dapat mengurangi proses pertemuan langsung dengan guru yang dinilai untuk melakukan pengecekan sesuai penilaiannya pada penilaian kinerja karyawan di SDIT Al-Qudwah?” dan “Bagaimana nilai manfaat dari OrangeHRM untuk penilaian kinerja karyawan di SDIT Al-Qudwah?”. Setelah penelitian ini dilakukan diharapkan dapat mengurangi penggunaan kertas pada penilaian kinerja karyawan di bagian SDM SDIT Al-Qudwah, penilaian kinerja karyawan terdokumentasi dengan baik dalam Sistem Informasi SDM, data karyawan terintegrasi dengan baik dan mengurangi proses penilaian kinerja melalui pertemuan langsung dengan guru yang dinilai untuk pengecekan penilaian.

2. METODE PENELITIAN

Penelitian ini dilakukan dengan menggunakan metode diagram alir yang terdiri dari studi pendahuluan berupa wawancara dengan manajer SDM SDIT Al-Qudwah, analisis hasil wawancara kemudian melakukan perancangan sistem sesuai hasil analisis. Selanjutnya implementasi sistem.

Tahapan selanjutnya adalah uji coba dengan menggunakan metode *Black Box*. Kemudian tahap diakhiri dengan penarikan kesimpulan dan rekomendasi dari penelitian yang sudah dilakukan agar dapat melakukan penelitian yang lebih baik dalam implementasi Sistem Informasi Sumber Daya Manusia di SDIT Al-Qudwah.

Penelitian ini termasuk ke dalam jenis penelitian implementasi sistem. Sistem yang diimplementasikan yaitu Sistem Informasi SDM dengan menggunakan OrangeHRM. Metode penelitian ini adalah kuantitatif deskriptif. Penelitian dilakukan di SDIT Al-Qudwah.

3. HASIL PENELITIAN

Pada bab ini akan dibahas mengenai analisis hasil studi lapangan, analisis masalah, analisis usulan perbaikan, dan perancangan yang terdiri dari SRS (*Software Requirement Spesification*) dan *Use Case Diagram*.

3.1 Analisis Hasil Studi Lapangan

Pada bagian ini akan dibahas hasil dari studi lapangan proses penilaian kinerja yang terjadi pada saat ini di SDIT Al-Qudwah. Selain itu sub bab ini juga membahas kebutuhan SDIT Al- Qudwah terhadap sistem informasi SDM untuk menangani proses penilaian kinerja gurunya. Selama ini proses penilaian kinerja karyawan di SDIT Al-Qudwah menggunakan cara manual yang disebut raport guru. Setiap semester Manajer SDM SDIT Al-Qudwah melakukan supervisi (monitoring dan evaluasi) bagi para karyawannya dan dari supervisi tersebut menghasilkan nilai yang akan dijadikan rekomendasi kepada yayasan misalnya dari satu guru yang telah disupervisi dan mendapat nilai baik maka gajinya akan dinaikkan. Selain itu, Manajer SDM SDIT Al-Qudwah dapat memberikan pelatihan yang tepat bagi setiap guru.

Supervisi dilakukan oleh Manajer SDM SDIT Al-Qudwah dengan merujuk kepada KPI yang sudah ditentukan. KPI (*Key Performance Indicator*) terbagi menjadi 4 kompetensi yaitu kompetensi profesional, kompetensi pedagogi, kompetensi pribadi dan kompetensi sosial. Dari hasil wawancara dengan Manajer SDM SDIT Al-Qudwah proses penilaian diawali dengan melihat presensi setiap guru sehingga mendapatkan total presensi dan kemudian dari total presensi tersebut dilanjutkan dengan penilaian kinerja terhadap 4 kompetensi yang menjadi KPI (*Key Performance Indicator*). Misalkan setiap kompetensi maksimal memiliki nilai 10 berarti nilai yang harus dipenuhi oleh guru maksimal 40.

Penilaian tersebut dilakukan oleh Kepala Unit kemudian dilaporkan ke Manajer SDM sehingga dapat mengetahui total gaji yang didapatkan bulan tersebut, prestasi yang dihasilkan dan pelatihan yang dibutuhkan oleh guru. Selain penilaian yang dilakukan oleh Kepala Unit dan Manajer SDM, siswa juga menilai melalui observasi selama proses belajar mengajar di kelas. Observasi yang dilakukan siswa hanya tambahan penilaian saja. Observasi tersebut bertujuan untuk menemukan sesuatu yang dilakukan oleh guru ketika di kelas.

Setelah dilakukan penilaian oleh Manajer SDM, guru akan mendapatkan jumlah nilai dengan kesimpulan sebagai berikut : tidak sesuai, sesuai sebagian dan sesuai seluruhnya. Kemudian diberikan komentar atau rekomendasi terhadap RPP (Rencana Pelaksanaan Pembelajaran) secara umum. Sepuluh bagian penilaian tersebut terbagi menjadi 4 KPI (*Key Performance*

Indicator) yang sudah disebutkan sebelumnya. Setelah dilakukan penilaian kemudian dilakukan pengecekan langsung kepada guru yang dinilai.

3.2 Analisis Masalah

Berdasarkan hasil analisis proses penilaian kinerja yang ada di SDIT Al-Qudwah, masih terdapat beberapa permasalahan di antaranya sebagai berikut.

1. Tidak adanya dokumentasi yang baik untuk penilaian kinerja guru dan *database* guru karena masih menggunakan kertas;
2. Memerlukan waktu untuk tatap muka dengan guru yang bersangkutan ketika proses penilaian agar guru yang bersangkutan dapat mengecek langsung;
3. Proses penilaian tidak sesuai dengan waktu yang telah ditentukan karena membutuhkan waktu cukup lama ketika proses penilaian, sehingga yang seharusnya dilakukan tiga kali penilaian menjadi hanya satu kali penilaian.

3.3 Analisis Usulan Perbaikan

Usulan pemecahan masalah terhadap pengelolaan SDM di SDIT Al-Qudwah, anatar lain:

1. Menggunakan OrangeHRM dalam mengelola SDM di bagian SDM SDIT Al-Qudwah dengan modul *PIM, ESS, Time* dan *Performance*.
2. Melakukan presensi pada modul *Time* agar terdokumentasi dengan baik.
3. Melakukan penilaian dengan modul *Performance*.
4. Menyimpan data personal guru pada modul *PIM*.

3.4 Perancangan

Perancangan dilakukan setelah penulis selesai melakukan analisis masalah tersebut. Perancangan yang dilakukan terdiri dari SRS (*Software Requirement Spesification*) diantaranya kebutuhan fungsional, kebutuhan non fungsional dan kebutuhan data. Kemudian *Use case Diagram* dan *Activity Diagram*.

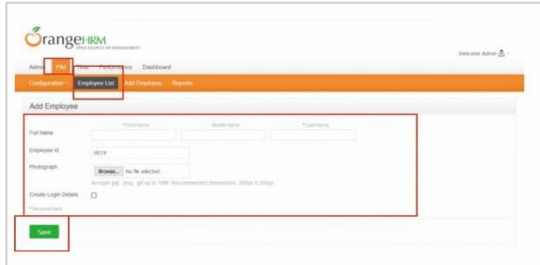
Berdasarkan kebutuhan fungsional terdiri dari dua aktor yang terlibat dari penelitian ini yaitu Manajer SDM yang juga sebagai admin dapat melakukan *login*, mengatur *role user*, melakukan penilaian dan membuat data personal guru. Kemudian actor selanjutnya yaitu Guru yang dapat melakukan *login*, melakukan presensi dan melakukan penilaian terhadap dirinya dan melihat penilaian yang dilakukan oleh Manajer SDM SDIT Al- Qudwah.

Selanjutnya terdapat *Use Case* utama dari aplikasi OrangeHRM yang sesuai dengan SRS yang telah ditentukan di perancangan sebelumnya. *Use Case* utama tersebut merupakan acuan untuk merancang *Activity Diagram* yang merupakan rincian aktivitas dari setiap *Use case*.

4. IMPLEMENTASI

Berikut merupakan hasil implementasi OrangeHRM di SDIT Al-Qudwah. Implementasi tersebut memiliki tiga fitur utama yaitu sebagai berikut.

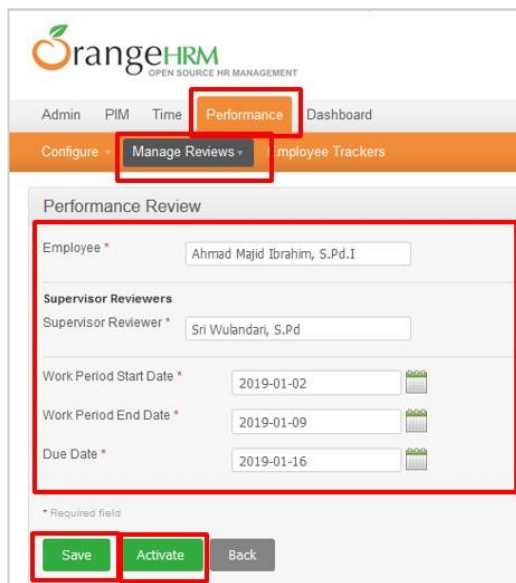
1. Memasukkan data personal guru ke dalam OrangeHRM dengan modul PIM.



Gambar 1. Memasukkan data personal guru ke dalam OrangeHRM dengan modul PIM

Pada tampilan ini, manajer SDM dapat memasukkan data personal guru SDIT Al-Qudwah. Setelah data tersebut dimasukkan, maka bagian SDM SDIT Al-Qudwah memiliki *database* data guru yang baik karena terdokumentasi dalam OrangeHRM.

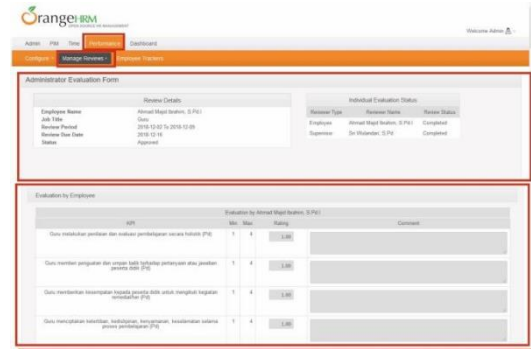
2. Melakukan penjadwalan penilaian guru



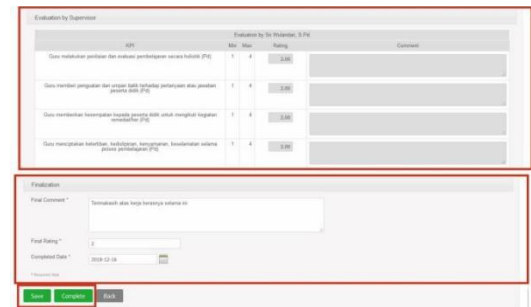
Gambar 2. Melakukan penjadwalan penilaian

Sebelum melakukan penilaian, Manajer SDM harus membuat jadwal penilaian guru. Kemudian setelah jadwalnya, maka Manajer SDM dapat melakukan penilaian selama masa penilaian.

3. Melakukan Penilaian



Gambar 3. Melakukan Penilaian (1)



Gambar 4. Melakukan Penilaian (2)

Pada tampilan ini, Manajer SDM SDIT Al-Qudwah dapat melakukan penilaian yang sesuai dengan KPI yang telah dimasukkan ke dalam OrangeHRM. Hasil akhirnya berupa *Final Rating*, komentar dan pelatihan yang dibutuhkan serta prestasi apa yang didapatkan.

5. DISKUSI DAN PEMBAHASAN

Berdasarkan dari hasil implementasi yang telah dilakukan, maka terdapat beberapa hal yang perlu diperhatikan ketika akan memasukkan data personal guru ke dalam OrangeHRM yaitu admin harus memasukkan data personal dengan selengkap-lengkapny sesuai aplikasi OrangeHRM. Selain itu, seringnya lupa untuk mengisi bagian *Report-to*. Hal itu menyebabkan menghambatnya ketika proses penilaian guru. Karena *Report-to* itu biasanya tidak diperhatikan oleh admin karena terfokus dengan data personal lainnya seperti *Personal Detail*, *Contact Detail*, *Job* dan *Salary*.

Selain itu ketika admin memberikan *username* dan *password* dalam modul Admin kepada setiap guru, hendaknya disesuaikan dengan keinginan guru agar tidak terjadi hal yang tidak diinginkan seperti lupa *username* guru karena tidak dicatat terlebih dahulu. Tetapi akan berbeda halnya ketika guru yang menentukan *username* sendiri maka akan menjadi tanggung jawab bagi setiap guru tersebut.

Pada saat melakukan penilaian pada modul *Performance*, harus dibuat jadwal penilaian terlebih dahulu oleh admin kepada guru yang akan dinilai. Karena jika tidak

diberikan jadwal penilaian, maka tidak akan dapat dilakukan penilaian oleh admin.

6. KESIMPULAN DAN SARAN

Penelitian yang telah dilakukan menghasilkan kesimpulan bahwa implementasi OrangeHRM di bagian SDM SDIT Al-Qudwah dapat mendokumentasikan data penilaian kinerja guru dengan modul *Performance* dan data personal guru dengan modul PIM, OrangeHRM dapat mengurangi pertemuan langsung dengan guru yang bersangkutan dalam proses penilaian kinerja guru dan OrangeHRM bermanfaat dalam proses penilaian kinerja guru di SDIT Al-Qudwah.

Berdasarkan penelitian yang telah dilakukan penulis, terdapat beberapa saran untuk penelitian selanjutnya yaitu memperbanyak target karyawan yang akan dikelola menggunakan OrangeHRM seperti Tata Usaha dan *Office Boy*. Selain itu diharapkan pada penelitian selanjutnya, implementasi sistem informasi SDM menggunakan OrangeHRM lebih luas lagi, tidak hanya membatasi pada modul *Performance*, *Time*, *Admin* dan PIM saja tapi juga *Recruitment* dan *Leave*.

DAFTAR PUSTAKA

- [1] Rasmanto, "Analisis Implementasi Sistem Informasi Manajemen SDM OrangeHRM, Studi Kasus Bagian Kepegawaian STT NF," Jurnal Informatika Terpadu, 2015.
- [2] Amirullah, "Pengantar Manajemen," Jakarta: Mitra Wacana Media, 2015.
- [3] I. K. Rachmawati, "Manajemen Sumber Daya Manusia," Yogyakarta: C.V Andi OFFSET, 2008.
- [4] Sedarmayanti, "Manajemen Sumber Daya Manusia Reformasi Birokrasi dan Manajemen Pegawai Negeri Sipil," Edisi Revisi, Bandung: PT Refika Aditama, 2017.
- [5] J. Luhukay, "Analisis Penerapan Aplikasi OrangeHRM pada Departemen *Human Resource Development* (Studi Kasus: Departemen HRD Bakrie *Microfinance* Indonesia)," 2012.



IMPLEMENTASI *NETWORK MONITORING SYSTEM* TERINTEGRASI DENGAN *TICKETING SYSTEM* MENGGUNAKAN NAGIOS DAN OSTICKET

Debi Onas Saputra¹, Henry Saptono²

^{1,2}Teknik Informatika, Sekolah Tinggi Teknologi Terpadu Nurul Fikri
Jakarta Selatan, DKI Jakarta, Indonesia 12640
saputraonas@gmail.com, henry@nurulfikri.ac.id

Abstract

This research proposes integrating a network monitoring system based on Nagios with a ticketing system based on ticket os. In this study, there are steps to integrate the Nagios-based monitoring system with the ticketing os-based ticketing system and provide convenience for network managers in carrying out network maintenance with a network monitoring system supported by an automatic ticketing system monitoring system that detects infrastructure failures in the network. There are four tests, namely testing the up and down status of the host, then testing the warning and critical service status. The results obtained from testing the integration of the Nagios-based network monitoring system with the osTicket-based ticketing system, namely utilizing the alerting feature on Nagios and the web API feature on osTicket showed effective results. This is evidenced by the automatic formation of tickets on the ticket when the host status is down and when the service status is warning or critical.

Keywords: *Network monitoring system, Alerting, Python, Nagios, osTicket*

Abstrak

Penelitian ini diajukan untuk mengintegrasikan *network monitoring system* berbasis Nagios dengan *ticketing system* berbasis osTicket. Pada penelitian ini terdapat langkah-langkah dalam mengintegrasikan *monitoring system* berbasis Nagios dengan *ticketing system* berbasis osTicket dan memberikan kemudahan bagi pengelola jaringan dalam melakukan pemeliharaan jaringan dengan adanya *network monitoring system* yang didukung dengan *ticketing system* otomatis ketika *network monitoring system* mendeteksi kegagalan infrastruktur dalam jaringan. Terdapat 4 pengujian yaitu, pengujian *host status up* dan *down*, kemudian pengujian *service status warning* dan *critical*. Hasil yang didapat dari pengujian pada integrasi *network monitoring system* berbasis Nagios dengan *system ticketing* berbasis osTicket ini yaitu memanfaatkan fitur alerting pada nagios dan fitur web API yang ada pada osTicket menunjukkan hasil yang efektif. Hal ini dibuktikan dari terbentuknya tiket secara otomatis pada osTicket pada saat status host down dan ketika status *service warning* atau *critical*.

Kata kunci: *Network monitoring system, Alerting, Python, Nagios, osTicket*

1. PENDAHULUAN

Network Monitoring System adalah sebuah sistem yang memiliki fungsi untuk mengamati atau memonitor sistem jaringan komputer yang sedang berjalan dan memungkinkan deteksi kesalahan pada jaringan. Monitoring adalah proses rutin pengumpulan data dan pengukuran kemajuan atas objektif program atau memantau perubahan, yang fokus pada proses dan keluaran.

Network Monitoring System dilakukan dengan menggunakan sebuah komputer yang dijadikan sebagai

server yang dihubungkan menggunakan jaringan lokal. Pada umumnya sebuah monitoring jaringan mengharuskan administrator untuk terus berada didepan layar agar mengetahui setiap perubahan yang terjadi hal ini kurang efisien. Untuk mengatasi masalah tersebut akan dilakukan sebuah monitoring yang menggunakan perangkat lunak yaitu Nagios.

Sebelumnya *system* monitoring berbasis Nagios hanya melakukan metode alerting melalui email. Setelah itu *system* monitoring tersebut bisa dikembangkan pada metode alerting. Oleh karena itu peneliti mengembangkan

system monitoring berbasis Nagios yang terintegrasi pada *system ticketing* berbasis osTicket dengan menerapkan alerting pada ticketing berbasis osTicket. Supaya pengguna bisa memanfaatkan *system ticketing* pada osTicket untuk memonitoring *system*.

Oleh karena itu dapat memberikan kemudahan bagi pengelola jaringan dalam melakukan pemeliharaan dengan adanya *network monitoring system* yang didukung dengan *system ticketing* otomatis ketika *network monitoring system* mendeteksi kendala infrastruktur dalam jaringan. Oleh karena itu dengan adanya sistem ticketing atau IT *helpdesk* dapat memudahkan semua penggunaan perangkat IT melaporkan setiap permasalahan yang dihadapi. Sehingga, dapat menghasilkan solusi yang tepat dalam mengatur sumber daya yang ada.

Nagios dapat memberi informasi tentang kondisi komponen jaringan secara *realtime* dengan mengirimkan pesan kepada administrator jaringan melalui osTicket. Sehingga osTicket ini berfungsi untuk mengetahui notifikasi dan permasalahan yang terjadi pada sebuah *monitoring* tersebut.

1.1 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan diatas, rumusan masalah dari penelitian ini adalah:

1. Proses apakah yang dapat digunakan atau diterapkan dalam mengintegrasikan *network monitoring system* berbasis Nagios dengan ticketing system osTicket?
2. Bagaimanakah cara integrasi *network monitoring system* berbasis Nagios dengan *ticketing system* berbasis osTicket?
3. Bagaimanakah efektifitas dari integrasi *network monitoring system* berbasis Nagios dengan *ticketing system* berbasis osTicket?

1.2 Tujuan penelitian

1. Mengimplementasikan *network monitoring system* berbasis Nagios dan *ticketing system* berbasis osTicket.
2. Merancang dan membuat program untuk mengintegrasikan *network monitoring system* berbasis Nagios dengan *system ticketing* berbasis osTicket.
3. Menyediakan mekanisme alerting dan notifikasi pada *network monitoring* Nagios yang dapat menciptakan ticket secara otomatis pada ticket berbasis OS Ticket.

1.3 Manfaat Penelitian

1. Memberikan kemudahan bagi pengelola jaringan dalam melakukan pemeliharaan jaringan dengan adanya *network monitoring system* yang didukung

dengan *ticketing system* otomatis ketika *network monitoring system* mendeteksi kegagalan infrastruktur dalam jaringan.

2. Menghasilkan dokumen tugas akhir yang dapat menjadi sebuah alternatif referensi bagi siapa pun yang ingin mempelajari dan menerapkan sebuah *monitoring system* berbasis Nagios yang terintegrasi dengan *ticketing system* berbasis osTicket.

1.4 Batasan Masalah

Batasan masalah dalam penelitian ini adalah:

1. Implementasi *network monitoring system* dan *ticketing system* dilakukan pada lingkungan percobaan.
2. Mekanisme peringatan (*alerting*) dari *network monitoring system* hanya berfokus pada pembuatan ticket otomatis di osTicket.
3. Hal hal yang dimonitoring oleh *network monitoring system* hanya yang berkaitan dengan status dan performa sistem.
4. Penelitian tidak mengembangkan atau memodifikasi dashboard monitoring system Nagios maupun ticketing system pada osTicket.

2. LANDASAN TEORI

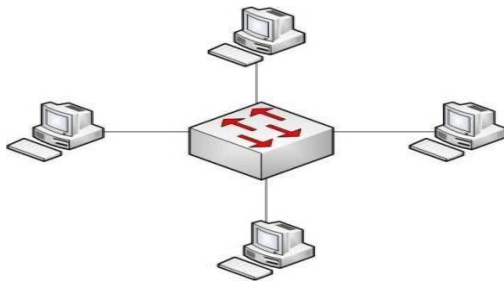
2.1 Jaringan Komputer

Jaringan komputer merupakan sekelompok komputer otonom yang saling berhubungan satu sama lain, dengan menggunakan satu protokol komunikasi sehingga seluruh komputer yang saling terhubung tersebut dapat berbagi informasi, program, sumber daya dan juga dapat saling menggunakan perangkat keras lainnya secara bersamaan, seperti printer, harddisk dan lain sebagainya. Jaringan komputer merupakan kumpulan sejumlah terminal komunikasi yang tersebar di berbagai tempat yang terdiri lebih dari satu komputer yang saling berhubungan.

Ada dua bagian dalam sebuah jaringan komputer yaitu pihak yang meminta layanan pada jaringan di sebut klien (*client*) sedangkan yang memberikan layanan pada jaringan komputer disebut peladen (*server*).

2.2 LAN (*Lokal Area Network*)

LAN adalah singkatan dari *Lokal Area Network*. Jenis jaringan LAN ini sangat sering kita temui di warnet-warnet, kampus, sekolah ataupun perkantoran yang membutuhkan hubungan atau koneksi antara dua komputer atau lebih dalam suatu ruangan. Jaringan LAN juga merupakan jaringan yang sangat di pengaruhi oleh topologi jaringannya.

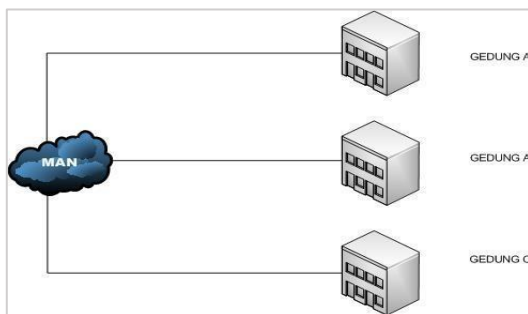


Gambar 1. LAN

2.3 MAN (Metropolitan Area Network)

MAN singkatan dari *Metropolitan Area Network*. Jenis jaringan komputer MAN ini adalah suatu jaringan komputer dalam suatu kota dengan transfer data berkecepatan tinggi yang menghubungkan suatu lokasi seperti sekolah, kampus, perkantoran dan pemerintahan.

Sebenarnya jaringan MAN ini adalah gabungan dari beberapa jaringan LAN. Jangkauan dari jaringan MAN ini bisa mencapai 10 - 50 kilometer.



Gambar 2. MAN

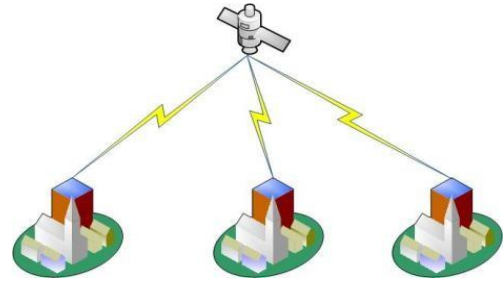
2.4 WAN (Wide Area Network)

WAN singkatan dari *Wide Area Network*. WAN adalah jenis jaringan komputer yang mencakup area yang cukup besar. contohnya adalah jaringan yang menghubungkan suatu wilayah atau suatu negara dengan negara lainnya [6]. Adapun beberapa karakteristik dari jaringan WAN, diantaranya sebagai berikut ini:

- Biasanya WAN digunakan untuk menghubungkan perangkat-perangkat yang tidak dapat dihubungkan melalui jaringan LAN dan jaringan MAN. Oleh karena itu jaringan WAN digunakan untuk menghubungkan jaringan yang sangat luas.
- WAN memiliki cakupan area yang sangat luas. Biasanya pada jaringan WAN akan melibatkan operator telekomunikasi. Tujuannya menggunakan operator telokomunikasi yaitu supaya perangkat-perangkat yang ada dalam jaringan WAN dapat saling berkomunikasi satu sama lain.
- Menggunakan koneksi serial dari berbagai macam jenis untuk dapat mengakses *bandwith* dalam lokasi yang berjauhan atau luas.
- Dapat melakukan pertukaran paket data maupun

frame antar *router* atau *switch* dan jaringan LAN yang sudah dibangun.

- Bekerja pada layer fisik dan pada layer data link dari layer OSI.



Gambar 3. WAN

2.5 TCP/IP

TCP/IP (*Transmission Control Protokol/Internet Protokol*) adalah sekumpulan protocol yang terdapat di dalam jaringan komputer yang digunakan untuk berkomunikasi atau bertukar data antar komputer. TCP/IP merupakan protokol standar pada jaringan internet yang menghubungkan banyak komputer yang berbeda jenis mesin maupun sistem operasi agar dapat berinteraksi satu sama lain. TCP/IP secara umum berfungsi untuk memilih rute terbaik transmisi data, memilih rute alternative jika suatu rute tidak dapat digunakan, mengatur, dan mengirimkan paket-paket pengiriman data dan lain-lain. TCP/IP merupakan protokol yang memungkinkan sistem di seluruh dunia berkomunikasi pada jaringan tunggal yang disebut *Internet*.

2.6 Network Monitoring System

Network Monitoring System adalah sistem yang memiliki fungsi untuk mengamati atau memonitor sistem jaringan komputer yang sedang berjalan dan memungkinkan deteksi dini terjadinya kesalahan pada jaringan. *Network Monitoring System* dilakukan dengan menggunakan sebuah komputer yang dijadikan sebagai *server* yang dihubungkan menggunakan jaringan *local*. *Network Monitoring System* adalah sistem ekstra atau kumpulan sistem yang memiliki tugas mengamati/memonitor sistem-sistem terhadap kemungkinan terjadinya masalah-masalah pada sistem tersebut untuk dapat dideteksi secara dini. Suatu *network monitoring system* dapat secara periodik menghubungi suatu server untuk menjamin adanya respon dari server, jika tidak ada respon maka monitoring sistem kemudian mengirimkan pesan atau notifikasi ke administrator.

2.7 Server

Server adalah komputer yang hanya memberikan layanan bagi komputer lain [9]. *Server* didukung oleh perangkat lunak yang memadai untuk mengontrol akses terhadap jaringan dan sumber daya yang terdapat di dalamnya. Tugas utama *server* adalah melayani beberapa komputer

client.

2.8 Client

Client adalah komputer yang hanya bisa menggunakan resource yang disediakan *server* sesuai dengan otoritas yang diberikan oleh administrator. Aplikasi yang dijalankan pada sisi *client* bisa saja merupakan *resource* yang tersedia di server atau aplikasi yang diinstall di sisi *client* namun hanya bisa di jalankan setelah terkoneksi ke sever.

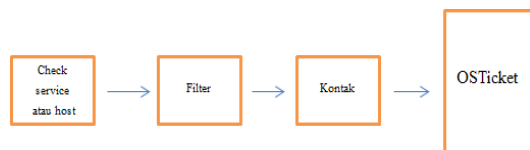
2.9 SNMP (Simple Network Management Protocol)

Simple Network Management Protocol merupakan *protocol* aplikasi yang mampu menjalankan tugas untuk memonitoring kondisi jaringan. SNMP akan mempengaruhi proses monitoring dan manajemen jaringan karena dengan menggunakan SNMP akan dapat diketahui informasi tentang kondisi perangkat jaringan yang diamati.

SNMP adalah sebuah *protocol Application layer* pada standard 7 OSI *layer* dan merupakan bagian dari *protocol TCP/IP* yang banyak digunakan untuk mengatur pertukaran *database* informasi yang menyangkut sistem manajemen dari sebuah perangkat jaringan.

2.10 Alert

Alert (peringatan) adalah notifikasi laporan yang dikirimkan Nagios kepada kontak administrator. *Alert* dapat dikirim melalui osTicket. *Alert* dapat dikirim ketika nagios mendeteksi perubahan status pada suatu *host* atau servis, maka Nagios akan memberitahukan melalui web *interface* dan *log file* serta mengirimkan notifikasi berupa *ticketing system* pada osTicket kealamat administrator. Filter pesan merupakan aturan-aturan yang dibuat dalam konfigurasi agar aktivitas *monitoring* dan notifikasi yang dikirim ke administrator yang diinginkan.



Gambar 4. Alert

2.11 Nagios

Nagios adalah aplikasi monitoring yang dapat digunakan untuk monitoring sistem komputer, monitoring jaringan dan monitoring infrastruktur jaringan. Nagios diciptakan oleh Ethan Galstad. Nagios berbasis *open source* yang dapat dijalankan pada sistem operasi Linux. Sistem monitoring Nagios memonitor seluruh infrastruktur IT untuk memastikan sistem, aplikasi, layanan, dan proses bisnis yang berfungsi dengan baik. Jika jaringan mengalami masalah, Nagios dapat langsung memberikan alert kepada teknisi jaringan sehingga memungkinkan

teknisi untuk mengatasi masalah tersebut lebih awal [4]. Adapun fungsi dari Nagios:

- Memantau layanan jaringan (SMTP, POP3, HTTP, NNTP, PING, dll).
- Memantau *resource server* (*Processor load, Disk usage, memory, dll*).
- Kemudahan untuk membuat *plugin*.
- Pengecekan layanan secara paralel.
- Dengan implementasi “*parents*” dari suatu *host*, maka bisa dengan mudah dibedakan *host* yang benar-benar *down*, dan *host* yang terkena dampaknya.

2.12 Helpdesk Ticketing System

Ticketing System adalah sebuah karcis gangguan atau disebut laporan masalah adalah mekanisme yang digunakan dalam suatu organisasi untuk melacak deteksi, pelaporan, dan resolusi dari beberapa jenis masalah. Trouble sistem *ticketing* berasal dari manufaktur sebagai dasar sistem pelaporan kertas, sekarang kebanyakan berbasis web dan terkait dengan hubungan manajemen pelanggan (CRM) lingkungan, seperti *call center* atau *e-business* [8]. Adapun *helpdesk* bertujuan untuk memudahkan semua penggunaan perangkat IT dalam ruang lingkup perusahaan maupun lembaga instansi pemerintah yang melaporkan setiap permasalahan yang dihadapi. Sehingga, dapat menghasilkan solusi yang tepat dalam mengatur sumber daya yang ada.

2.13 osTicket

osTicket adalah aplikasi *open source* yang digunakan untuk *ticketing system* gratis. osTicket bisa menjadi solusi murah untuk sebuah perusahaan menengah ke bawah untuk mengarsipkan kegiatan-kegiatan atau mungkin pendataan *problem-problem* yang terjadi pada perusahaan [9].

Aplikasi osTicket memiliki beberapa kelebihan dibandingkan aplikasi sistem pengelolaan tiket yang lain, beberapa kelebihan yang dimiliki osTicket adalah sebagai berikut :

- osTicket merupakan aplikasi yang berbasis *web-based platform* sehingga dapat diakses secara luas.
- Memiliki fitur-fitur diantaranya: *auto responder* tiket, email yang terintegrasi, dilengkapi dengan *role-based access* untuk manajemen *user, ticket assignment, alert* dan *notice* untuk tiket yang dikelola, *ticket filtering, email templates, service level agreements* dan *knowledge base management*
- Mendukung pemanfaatan komunikasi data dan informasi menggunakan layanan API.

2.14 Apache Web Server

Apache adalah sebuah *server web* yang dapat dijalankan di banyak system operasi (Unix, BSD, Linux, Microsoft Windows dan *platform* lainnya) yang berguna untuk

melayani dan memfungsikan situs web. Protokol yang digunakan untuk melayani fasilitas web/www ini menggunakan HTTP. Apache memiliki fitur-fitur canggih seperti pesan kesalahan yang dapat diatur autentikasi berbasis basis data dan lainnya. Apache juga didukung oleh sejumlah antarmuka pengguna berbasis grafik yang memungkinkan penanganan server menjadi mudah [5].

2.15 Python

Python adalah bahasa pemrograman yang bersifat *open source*. Bahasa pemrograman ini dioptimalisasikan untuk *software quality, developer productivity, program portability, dan component integration*. Python telah digunakan untuk mengembangkan berbagai macam perangkat lunak, seperti *internet scripting, systems programming, user interfaces, product customization, numeric programming* dll. Python saat ini telah menduduki posisi 4 atau 5 bahasa pemrograman paling sering digunakan di seluruh dunia. Bahasa pemrograman Python memiliki beberapa fitur yang dapat digunakan oleh pengembang perangkat lunak.

Berikut adalah beberapa fitur yang ada pada bahasa pemrograman Python :

1. Multi Paradigm Design
2. Open Source
3. Simplicity
4. Library Support
5. Portability
6. Extendable

3. ANALISIS DAN PERANCANGAN

3.1 Analisis Kebutuhan Hardware

Di dalam hardware ini menggunakan 2 virtual dengan spesifikasi sebagai berikut :

- Processor : Dual Core , 1,2 Ghz
- RAM : 1 GB
- Harddisk : 20 GB

Dalam penelitian ini, peneliti menggunakan laptop dengan spesifikasi sebagai berikut:

- Processor : Intel® Core i5-7200U
CPU @2.50GHz
- RAM : 8.00 GB
- Hardisk : 1 TB

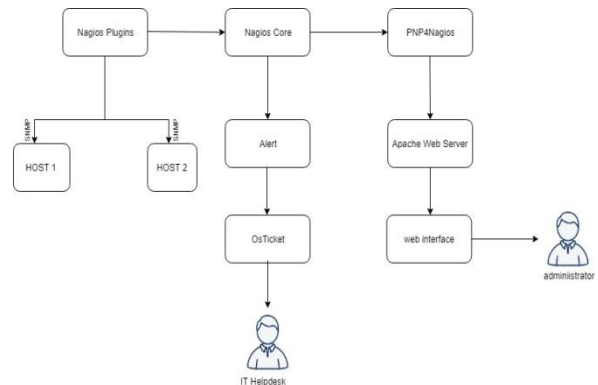
3.2 Analisis Kebutuhan Software

Dalam penelitian analisis kinerja HDFS sebagai infrastruktur pembangunan big data, peneliti menggunakan perangkat lunak (*software*) yang akan digunakan:

1. Platform Monitoring : Nagios 4.3.4
2. Apache web server 2.4.37

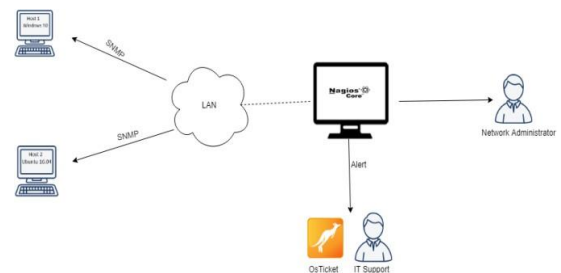
3. SNMP package
4. Platform Ticketing : osTicket
5. Apache web server 2.4.37
6. PHP 7.0
7. Mysql-server
8. Virtual Machine : Virtualbox dan VMWare
9. Web browser Chrome
10. Sistem Operasi : Linux Ubuntu 16.04
11. Bahasa Pemrograman : Python 2.7.123

3.3 Perancangan Arsitektur Sistem Logic



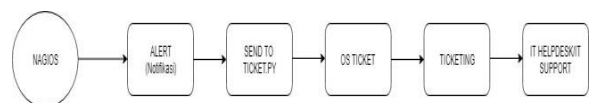
Gambar 4. Perancangan Arsitektur Sistem Logic

3.4 Perancangan Arsitektur Sistem Fisik



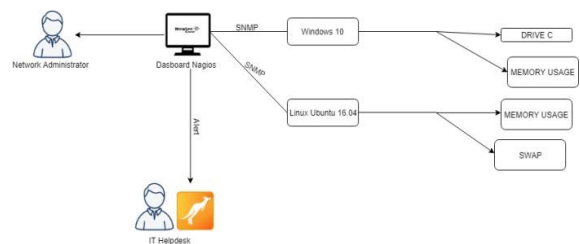
Gambar 5. Perancangan Arsitektur Sistem Fisik

3.5 Perancangan Alerting



Gambar 6. Perancangan Alerting

3.6 Rancangan Pengujian



Gambar 7. Rancangan Pengujian

4. IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi

1. Instalasi Nagios dengan *men-download* pada *website*-nya. Kemudian ekstrak dengan perintah:

```
$tar xzf nagios-4.3.4.tar.gz
```

2. Setelah mengekstrak, kemudian masuk ke direktori Nagios dan jalankan *configure* skrip untuk menentukan pengguna dan grup pada, dengan ketik perintah:

```
./configure--with-nagiosgroup=nagios--with-command-group=nagcmd
```

```
$make all
```

3. Kemudian untuk mengeluarkan perintah eksternal melalui antarmuka *web* ke Nagios, maka tambahkan pengguna server *web* yaitu *www-data* ke grup *nagcmd*:

```
$sudo usermod -G nagcmd www-data
```

4. *Men-download plugins* pada *website* Nagios kemudian ekstrak file *download*-nya

```
$tar xzf nrpe-3.2.1.tar.gz
```

5. Kemudian *configure* *nrpe* dengan perintah *configure*

```
./configure
```

6. Kemudian pasang dan install *Check_nrpe* dengan perintah *make*:

```
$ make check_nrpe
```

```
$ sudo make install-plugin
```

7. Kemudian instalasi *Apache web server*, dengan ketik perintah

```
$sudo apt-get install apache2
```

8. Selanjutnya konfigurasi *Apache* untuk melayani antarmuka pengguna Nagios. Aktifkan modul penulisan ulang dan *cgi* *Apache* dengan perintah *a2enmod*

```
$ sudo a2enmod rewrite
```

```
$ sudo a2enmod cgi
```

9. Selanjutnya membuat *user* dan *password* untuk *login* ke *web interface* Nagios. Dengan menjalankan perintah seperti di bawah ini

```
$sudo htpasswd -c
```

```
/usr/local/nagios/etc/htpasswd.users
```

10. Setelah itu membuat tautan untuk *nagios.conf* ke direktori yang didukung situs *nagios*

```
$sudo ln -s /etc/apache2/sites-available/nagios.conf
```

```
/etc/apache2/sites-enabled/
```

11. Setelah itu masuk ke direktori *nagios.conf* dan ubah seperti dibawah ini :

```
# SSLRequireSSL (Sebelum)
SSLRequireSSL (Sesudah)
```

```
Order allow,deny (sebelum) Allow
from all (sebelum)
```

```
#Order allow,deny (sesudah)
```

```
#Allow from all (sesudah)
```

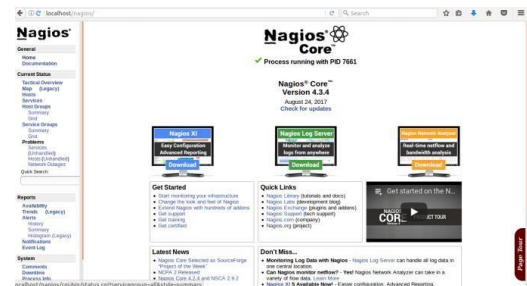
```
#Order deny,allow (sebelum) #Deny
from all (sebelum)
```

```
#Allow from 127.0.0.1 (sebelum)
```

```
Order deny,allow (sesudah) Deny
from all (sesudah)
```

```
Allow from 127.0.0.1 (sesudah)
```

12. Kemudian *restart* *nagios* dan membuka browser *localhost/nagios/* seperti gambar di bawah ini



Gambar 8. Restart Nagios

13. Kemudian Instalasi *osticket* dengan *men-download* pada *website*-nya. Kemudian ekstrak filenya seperti printer dibawah ini

```
$unzip osticket-v1.11.0-rc-1.zip
```

14. Kemudian mengubah *permission* direktori *osticket-v1.11.0-rc1* menjadi sebagai berikut

```
$ sudo chmod 755 -R osticket-
v1.11.0-rc1/
```

15. Selanjutnya, membuat *virtual host* untuk konfigurasi dengan perintah seperti dibawah ini

```
$ sudo nano/etc/apache2/sites-
available/000-default.conf
```

```
ServerAdmin webmaster@localhost
```

```

DocumentRoot /var/www/html
ServerName localhost

Alias /tiket
"/var/www/html/osticket/"

<Directory
/var/www/html/osticket/> Options
FollowSymLinks AllowOverride All

Order allow,deny allow from all

</Directory>

<Directory /var/www/html> Options
FollowSymLinks AllowOverride All

Order allow,deny

```

16. Setelah itu lakukan konfigurasi SSL-nya. Kemudian masukkan perintah seperti dibawah ini untuk masuk ke file yang ingin dikonfigurasi

```

$ sudo nano

/etc/apache2/sites
available/default-ssl.conf

<VirtualHost _default_:443>
ServerAdmin webmaster@localhost
DocumentRoot /var/www/html/osticket-
v1.11.0-rc1/

ServerName localhost

ServerAlias http://localhost/

Alias /localhost
"/var/www/html/osticket/osticket-
v1.11.0-rc1/"

<Directory
/var/www/html/osticket/> Options
FollowSymLinks

AllowOverride All

Order allow,deny

allow from all

</Directory>

```

17. Selanjutnya menghidupkan konfigurasi yang sudah dibuat tadi untuk hak akses dengan perintah dibawah ini:

```

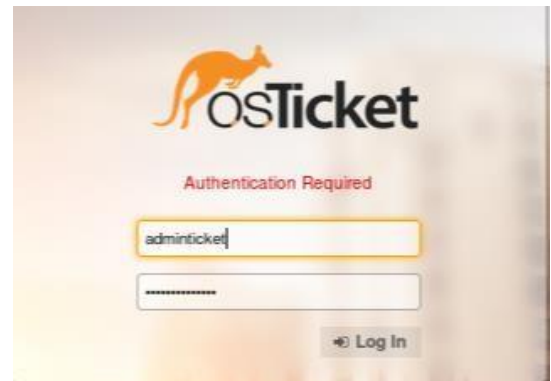
$a2ensite 000-default.conf

$a2ensite default-ssl.conf

```

18. Kemudian *restart* Apache dan membuka browser osTicket localhost/tiket/upload/scp/login.php.

berikut hasil tampilan dashboard pada osTicket.



Gambar 9. Tampilan Dashboard osTicket

19. Membuat API key osTicket. kemudian masuk ke admin panel, selanjutnya pilih *add API key*.

APIKey	IP Address	Status	Date Added	Last Updated
FD20114693E79F76069B310B3FE72C7D	192.168.32.7	Active	2/27/19	2/27/19, 2:01 PM

Gambar 10. API Key

20. Kemudian membuat kode program Python untuk mengintegrasikan *network monitoring system* berbasis Nagios dengan *system ticketing* berbasis osTicket. Pada sebelumnya peneliti mendapatkan referensi bentuk kode program pada *website* <http://sirlagz.net/2016/08/10/integrating-zabbix-with-osticket/> kemudian kode program tersebut dimodifikasi sesuai kebutuhan penelitian dalam mengintegrasikan *network monitoring system* berbasis Nagios dengan *ticketing system* berbasis osTicket.

```

url =

'http://192.168.32.5/tiket/api/ticket
s.json'

api_key =
'FD20114693E79F76069B310B3FE72C7D'

title = " "

def parse_args():

    parser =
    argparse.ArgumentParser(description
    =title)

        parser.add_argument('-
o', '-- object_type', nargs='?',
required=True)

        parser.add_argument('--
contact', nargs='?',
required=True)

```

```

        parser.add_argument('--
notificationtype', nargs='?')
        parser.add_argument('--
hoststate', nargs='?')
        parser.add_argument('--
hostname', nargs='?')
        parser.add_argument('--
hostaddress', nargs='?')

        parser.add_argument('--
servicestate', nargs='?')
        parser.add_argument('--
servicedesc', nargs='?')
        parser.add_argument('--
output', nargs='?')
args = parser.parse_args()

    return
args argsx =
parse_args()

def send_ticket(title,
message): priority = 1

if argsx.object_type == 'host':
    title = title +
argsx.object_type + " of " +
argsx.hostname + " " +
argsx.hoststate

    if argsx.hoststate
== "UP":
        priority = 2

    if argsx.hoststate
==
        priority = 4

    title = title +

"DOWN":

    else:

argsx.object_type + " of " +
argsx.hostname + " " +
argsx.servicestate

        if argsx.servicestate
==

"OK":
        priority = 2

"WARNING":
    if argsx.servicestate ==
        priority = 3

"Critical":
        priority = 4

    payload = {
        "autorespond": False,
        "source": "API",
        "name": "Debi Onas",
        "email":
"saputraonas@gmail.com",
        "subject": title,
        "message": message
        "priority": priority
    }

    header = {
        'Content-Type':
'application/json', 'X-API-Key': api_key
    }

    r =
requests.post(url, json=payload, header
s=header)

def host_notification(args):
    state = ''

    return '%s (%s) state is %s -
%s'%( args.hostname,
args.hostaddress,
args.hoststate,
args.output
)

def service_notification(args):
    state = ''

    return '%s (%s) state is %s -
%s '

%(args.hostname, args.servicedesc,
args.servicestate, args.output)

if __name__ == '__main__':
    args = parse_args()

```



```

if args.object_type ==
    'host':
        message =

host_notification(args)

elif args.object_type ==
    'service': message
=

service_notification(args)

send_ticket(title, message)
    
```

21. Membuka direktori command.cfg pada nagios untuk mendefinisikan argument service dan host.

```

define command {

    command_name notify-host-
by- osticket

    command_line

/usr/lib/nagios/plugins/check_ser
ver.py --object_type host --

contact "$CONTACTPAGER$" --
notificationtype
"$NOTIFICATIONTYPE$" --
hoststate "$HOSTSTATE$" --
hostname "$HOSTNAME$" -
hostaddress

"$HOSTADDRESS$" --output
"$HOSTOUTPUT$"

}
    
```

```

define command {

    command_name notify-
service- by-osticket

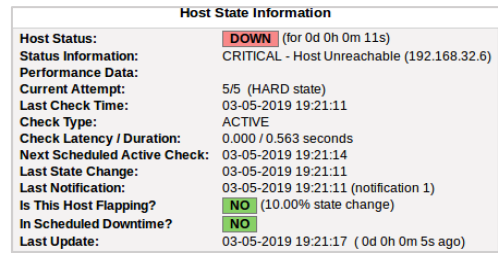
    command_line

/usr/lib/nagios/plugins/check_s
er ver.py --object_type service
-- contact "$CONTACTPAGER$" --
notificationtype
"$NOTIFICATIONTYPE$" --

servicestate "$SERVICESTATE$"
-- hostname "$HOSTNAME$" --
servicedesc "$SERVICEDESC$" --
output "$SERVICEOUTPUT$"
    
```

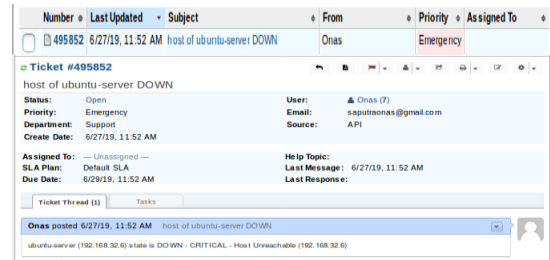
4.2 Pengujian Host

a. Mematikan host Ubuntu dengan cara mematikan panel koneksi network. Maka hasil Nagios seperti dibawah ini.



Gambar 11. Host State Information

b. Melihat pesan notifikasi pada osTicket bahwa host Ubuntu down.



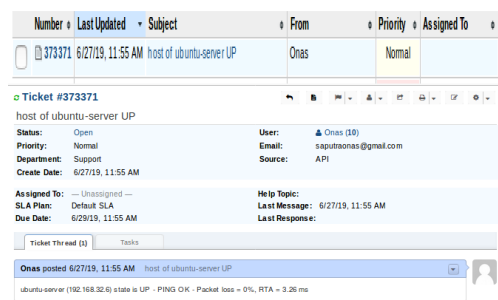
Gambar 12. Melihat Pesan Notifikasi pada osTicket

c. Kemudian menghidupkan kembali panel koneksi network pada host Ubuntu, kemudian terdapat perubahan pada nagios bahwa host Ubuntu berstatus up.



Gambar 13. Menghidupkan panel koneksi network

d. Kemudian melihat notifikasi pesan pada osTicket bahwa host Ubuntu mengalami perubahan status up



Gambar 14. Melihat Notifikasi Pesan

4.3 Hasil Pengujian Host

Setelah melakukan pengujian monitoring status host, peneliti membuat hasil pengujian dalam bentuk tabel untuk keberhasilan yang telah di ujikan sebanyak 4 kali. Hal ini menyatakan bahwa efektifitas keberhasilan sebuah notifikasi pada host Ubuntu, akan tetapi script python

tersebut bisa diujikan juga untuk jenis sistem operasi lain selain sistem operasi Ubuntu. Karena pada penelitian ini peneliti hanya melakukan salah satu *host* yaitu *host* Ubuntu. Berikut peneliti memaparkan keberhasilan terbentuknya notifikasi tiket secara otomatis pada osTicket untuk *host* Ubuntu.

Tabel 1. Hasil Pengujian Host

Host	Pengujian	Status host	Notifikasi
Ubuntu 16	Mematikan koneksi <i>network</i> pada <i>host</i> Ubuntu	Host Down	Tiket Terbentuk
Ubuntu 16	Menghidupkan koneksi <i>network</i> pada <i>host</i> Ubuntu	Host Up	Tiket Terbentuk
Ubuntu 16	Mematikan koneksi <i>network</i> pada <i>host</i> Ubuntu	Host Down	Tiket Terbentuk
Ubuntu 16	Menghidupkan koneksi <i>network</i> pada <i>host</i> Ubuntu	Host Up	Tiket Terbentuk

4.4 Pengujian Service

a. Melakukan perubahan ukuran *threshold* memory pada *command.cfg* di nagios. Pada sebelumnya -w 80-c 90 menjadi -w 1 -c 2. Untuk -w dalam artian maksimum nilai persen dalam status warning jika dia dari 1%-2% dia berstatus warning dan -c itu jika dia melebihi 2% maka berstatus critical. Setelah itu simpan perubahan.

```
define command {
    command_name    check_mem_linux
    command_line    $USER1$/nagios_plugins/check_smp_storage.pl -H $HOSTADDRESS$ -c $ARGC1$ -n "Memory" -w 1 -c 2 -f
}
```

Gambar 15. Perubahan Ukuran *Threshold* Memory

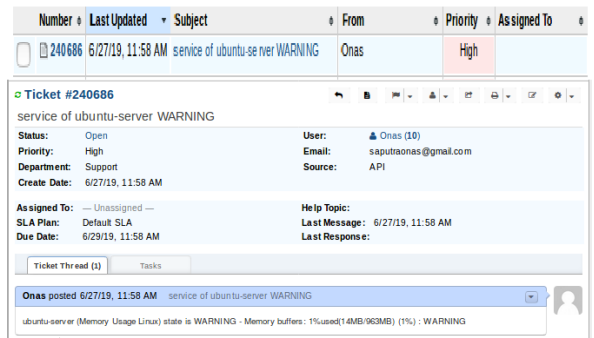
b. Kemudian melihat *dashboard* Nagios untuk melihat hasil sttus perubahan pada *service*, bahwa *service* berstatus *warning*.



Gambar 16. Melihat *Dashboard* Nagios

c. Kemudian melihat hasil notifikasi berupa tiket pada osTicket untuk melihat pesan dan membuka isi pesan

bahwa *memory* berstatus *warning*.



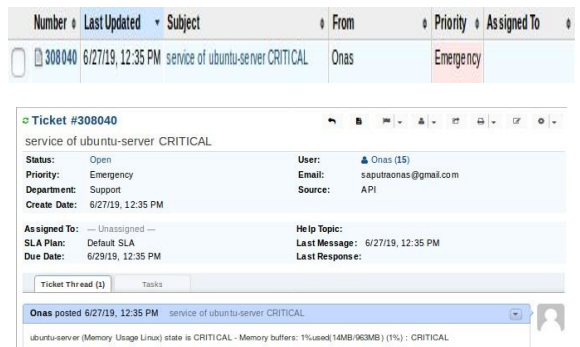
Gambar 17. Melihat Hasil Notifikasi

d. Kemudian peneliti menguji lagi, menjalankan program seperti Firefox, Libre Office di *host* Ubuntu tersebut. Maka hasilnya di Nagios *host* Ubuntu status berubah menjadi *critical*.



Gambar 18. Nagios *Host* Ubuntu Berubah menjadi *Critical*

e. Selanjutnya melihat hasil notifikasi berupa tiket pada osTicket untuk melihat pesan dan membuka pesan bahwa *memory* berstatus *critical*.



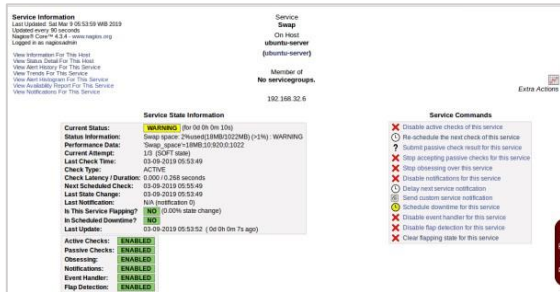
Gambar 19. Melihat Hasil Notifikasi berupa Tiket

f. Kemudian peneliti mengubah ukuran *threshold* swap pada *command.cfg* di nagios. Pada sebelumnya -w 80 - c 90 menjadi -w 1 -c 90. Untuk -w dalam artian maksimum nilai persen dalam status warning jika dia dari 1%-89% dia berstatus warning dan -c itu jika dia melebihi 90% maka berstatus critical. Setelah itu simpan perubahan.

```
define command {
    command_name check_swap_linux
    command_line SUSEB15/nagios_plugins/check_smp_storage.pl -H $HOSTADDRESS$ -C $ARG1$ -n $Swap -w 11 -c 90 -f
}
```

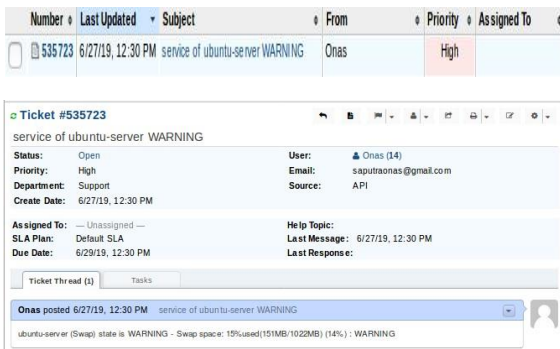
Gambar 20. Mengubah Ukuran *Threshold Swap* pada *Command*

- g. Setelah itu melihat dashboard nagios pada perubahan swap di nagios.



Gambar 21. Melihat Dashboard Nagios pada Perubahan Swap

- h. Kemudian mengecek notifikasi pada osticket. Untuk melihat pesan dan isi pesan notifikasi tersebut.



Gambar 22. Mengecek Notifikasi pada OS Ticket

- i. Kemudian peneliti mengubah ukuran *threshold* ukuran *swap* pada Nagios untuk mengubah status menjadi *critical*.

```
define command {
    command_name check_swap_linux
    command_line SUSEB15/nagios_plugins/check_smp_storage.pl -H $HOSTADDRESS$ -C $ARG1$ -n $Swap -w 0 -c 1 -f
}
```

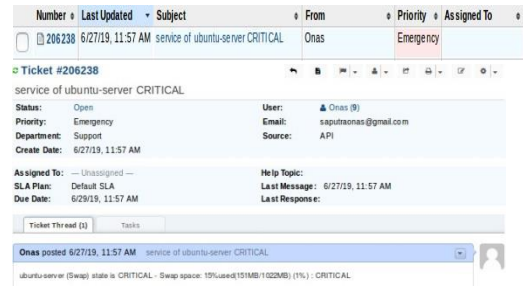
Gambar 23. Mengubah Ukuran *Threshold* Ukuran *Swap* pada Nagios

- j. Kemudian melihat hasil perubahan pada Nagios.



Gambar 24. Melihat Hasil Perubahan pada Nagios

- k. Kemudian mengecek notifikasi berupa tiket pada osTicket untuk melihat pesan dan membuka isi pesan bahwa *swap* berstatus *crit*



Gambar 25. Melihat dan Membuka Isi Pesan berstatus Critical

4.5 Hasil Pengujian *Service*

Setelah melakukan pengujian monitoring *service*, peneliti melakukan tahap pengujian 2 *service* yaitu *ram* dan *swap*. Kemudian peneliti membuat hasil pengujian dalam bentuk tabel sebanyak 4 kali pengujian ketika *service* mendapatkan kendala problem yaitu *warning* dan *critical*. Hal tersebut menyatakan efektifitas keberhasilan bahwa pada saat terjadinya problem di *service* maka menciptakan tiket secara otomatis pada osTicket. Berikut peneliti memaparkan keberhasilan terbentuknya notifikasi tiket secara otomatis pada osTicket untuk *service*.

Tabel 2. Hasil Pengujian *Service*

Host	Pengujian	Status	Notifikasi
Ubuntu	Mengubah Ukuran <i>Threshold</i> atau ukuran maksimum <i>memory</i>	<i>Warning</i>	Tiket Terbentuk
Ubuntu	Menjalankan program seperti Firefox, Libre Office pada host Ubuntu	<i>Critical</i>	Tiket Terbentuk
Ubuntu	Menjalankan program seperti Firefox, Libre Office pada <i>host</i> Ubuntu	<i>Warning</i>	Tiket Terbentuk
Ubuntu	Mengubah Ukuran <i>Threshold</i> atau ukuran Maksimum <i>Swap</i>	<i>Critical</i>	Tiket terbentuk

5 KESIMPULAN DAN SARAN

5.1 Kesimpulan

- a. Berdasarkan fitur *alerting* atau notifikasi yang terdapat pada *network monitoring system* berbasis Nagios, peneliti memanfaatkan fitur tersebut sebagai

metode atau proses yang digunakan dalam mengintegrasikan *network monitoring system* berbasis Nagios dengan *ticketing system* berbasis osTicket.

- b. Dengan adanya fitur *alerting* pada Nagios, peneliti membuat *script* dengan bahasa pemrograman Python yang melakukan akses ke *web API* dari osTicket *system*. Akses ke *web API* osTicket ini akan melakukan pembuatan tiket baru yang menunjukkan sebuah notifikasi yang terjadi pada nagios saat terjadi *host down* dan *service warning* dan *critical*.
- c. Berdasarkan hasil perancangan, penerapan, dan pengujian yang sudah dilakukan oleh peneliti, proses integrasi *network monitoring system* berbasis nagios dengan *ticketing system* berbasis osTicket, yang memanfaatkan fitur *alerting* pada nagios dan fitur *web API* yang ada pada osTicket menunjukkan hasil yang efektif. Hal ini dibuktikan dari terbentuknya tiket secara otomatis pada osTicket pada saat status *host down* dan ketika status *service warning* atau *critical*.

5.2 Saran

1. Penelitian berikutnya dapat mengimplementasikan kedalam sistem yang real atau tidak dalam lingkungan virtualisasi.
2. Melakukan peningkatan atau optimasi kode program Python yang dibuat oleh peneliti untuk pembuat tiket otomatis pada osTicket pada saat terjadinya perubahan status down menjadi status *up* dan dari status *service warning* atau *critical* menjadi status ok akan memproses terjadinya *close* tiket.
3. Dalam penelitian ini, peneliti tidak menerapkan sisi keamanan sistem monitoring pada Nagios maupun sistem *ticketing* pada osTicket. oleh karena itu penelitian selanjutnya dapat menyertakan pertimbangan sisi keamanan seperti penerapan HTTPS.

DAFTAR PUSTAKA

- [1] M. D. Haryanto, I. Riadi, "Analisis dan Optimalisasi Jaringan Menggunakan Teknik *Load Balancing*," Yogyakarta, 2014.
- [2] H. Amnur, et al., "Perancangan dan Implementasi *Network Monitoring System* Menggunakan Nagios dengan Email dan SMS *Alert*," Politeknik Negeri Padang, 2014.
- [3] H. Saptono, "*Network Monitoring System* dengan Nagios," 2008.
- [4] M. E. "Implementasi Nagios untuk Merancang *System Monitoring Client* Jaringan Komputer menggunakan Sms dan Email," Jakarta: Universitas Islam Syarif Hidayatullah, 2009.
- [5] J. P. Jumri, "Perancangan Sistem Monitoring Konsultasi Bimbingan Akademik Mahasiswa dengan Notifikasi *Realtime* berbasis *SMS Gateway*," Universitas Tanjungpura, 2012.
- [6] M. R. Zaini, "Implementasi Protokol SNMP untuk Jaringan," Univesitas Andalas, 2013.
- [7] Qoyyimah, "Rancang Bangun *Helpdesk Ticketing System* Studi Kasus PT Primus Indojaya," Jakarta: Universitas Syarif Hidayatullah, 2011.
- [8] M. S. Daulay, "Pengantar Jaringan Komputer," Yogyakarta: Andi, 2005.
- [9] M. Lutz, "*Programming Python*," *Fourth Edition ed.*, Sebastopo: O'Reilly Media, Inc., 2010.



PENGUJIAN DAN IMPLEMENTASI EFEKTIVITAS SISTEM VIRTUALISASI BERBASIS DOCKER PLATFORM

Husni Mubarak¹, Henry Saptono²

^{1,2}Teknik Informatika, Sekolah Tinggi Teknologi Terpadu Nurul Fikri
Jakarta Selatan, DKI Jakarta, Indonesia 12640

husnimubarak295@gmail.com, henry@nurulfikri.co.id

Abstract

The increasing number of application development needs require various and different services with different environments, so Docker-based virtualization is needed so that applications can be run on various platforms so that it can make it easier for application developers and system administrators to build and run applications that have been implemented designed, namely by running the required services in a container. This scientific writing intends to analyze and design a Docker-based virtualization platform installed on the Ubuntu 17.10 Linux operating system, including the process of installing, configuring, and running the Docker application. Docker is an open project that works to help developers or system administrators build, package, and run applications anywhere in a container. The architecture uses the client-server method, the docker client, sends requests to the Docker daemon to build, distribute and run Docker containers. Both the Docker client and the Docker daemon communicate via a socket using the Restful API. Docker provides tools and platforms to manage the lifecycle of containers where developers and system administrators can develop applications and their supporting components using containers. A container is a container into a unit to distribute and test applications ready to be installed.

Keywords: Docker, Client Server, Container, Docker Daemon, Socket, Restful API

Abstrak

Semakin banyaknya kebutuhan pengembangan aplikasi yang membutuhkan servis yang beragam dan berbeda-beda dengan lingkungan yang berbeda juga, maka dibutuhkan adanya virtualisasi dengan berbasis Docker agar aplikasi dapat dijalankan di berbagai *platform* yang berbeda sehingga dapat memudahkan pengembang aplikasi dan sistem administrator dalam membangun dan menjalankan aplikasi yang telah dirancang, yaitu dengan menjalankan servis yang dibutuhkan dalam sebuah kontainer. Penulisan ilmiah ini bermaksud menganalisis dan merancang sebuah *platform* virtualisasi berbasis Docker yang dipasang di operasi *system* Linux Ubuntu 17.10, meliputi proses instalasi, konfigurasi, dan menjalankan aplikasi Docker. Docker adalah sebuah projek terbuka yang berfungsi membantu cara kerja untuk *developer* atau sistem administrator untuk membangun, mengemas dan menjalankan aplikasi dimanapun dalam sebuah kontainer. arsitektur menggunakan metode klien *server*, Docker klien, mengirimkan permintaan ke docker daemon untuk membangun, mendistribusikan dan menjalankan kontainer Docker. Keduanya Docker klien dan Docker *daemon* berkomunikasi via *socket* menggunakan *Restful* API. Docker menyediakan alat dan *platform* untuk mengelola siklus hidup pada pada kontainer yaitu developer dan sistem administrator dapat mengembangkan aplikasi dan komponen pendukungnya menggunakan kontainer. Kontainer adalah sebuah wadah menjadi unit untuk mendistribusikan dan menjalankan uji coba aplikasi yang akan siap dipasang.

Kata kunci: Docker, Client Server, Kontainer, Docker Daemon, Socket, Restful API

1. PENDAHULUAN

Di antara banyaknya masalah di dunia perkembangan aplikasi, ada satu masalah klasik yang sering ditemui oleh pengembang aplikasi dan sistem administrator. Masalah tersebut yaitu perbedaan *platform* sistem operasi yang digunakan para pengembang aplikasi dan sistem administrator, maka diperlukanya rancangan arsitektur *platform* virtualisasi yang lebih baik dan terdistribusi. Ada

yang menggunakan MacOS, Linux ataupun Windows. Variasi *platform* di antara pengguna linux pun akan beragam, baik distro yang digunakan, maupun versi perangkat lunak yang terpasang di dalamnya. Intinya adalah tidak semua pengembang aplikasi menggunakan lingkungan kerja yang sama persis. Teknologi virtualisasi menawarkan perluasan penggunaan pada infrastruktur teknologi informasi menjadi lebih tinggi dengan biaya yang

lebih rendah karena virtualisasi memanfaatkan sumber daya komputer secara optimal sehingga dapat digunakan untuk berbagai kebutuhan, setiap kali penambahan aplikasi atau fitur baru maka diperlukan juga menambahkan mesin komputer beserta sistem operasi, sehingga biaya untuk lisensi perangkat lunak makin bertambah dengan virtualisasi penambahan komputer tidak diperlukan karena sebuah server dapat digunakan secara Bersama-sama dengan berbagai aplikasi yang akan diperlukan. Saat ini teknologi virtualisasi kontainer semakin populer terutama docker di mana dapat memasang aplikasi kedalam sebuah kontainer.

Docker adalah sebuah proyek terbuka yang menyediakan *platform* terbuka untuk pengembang aplikasi mau pun sistem administrator untuk dapat membangun, mengemas, dan menjalankan aplikasi di mana pun sebagai sebuah wadah kontainer yang ringan. docker awal mulanya dikembangkan oleh *Solomon hykes* sebagai proyek internal di *dotcloud*, sebuah perusahaan PAAS. Sistem kontainer docker melakukan pembungkusan pada bagian-bagian perangkat lunak dalam berkas sistem secara lengkap yang berisi semua kebutuhan yang diperlukan dalam menjalankan kode, *runtime*, sistem alat dan sistem perpustakaan yang dapat dijalankan pada *server*, sehingga perangkat lunak dapat berjalan sama meskipun dalam lingkungan yang berbeda. Kontainer berjalan pada satu komputer tunggal yang dapat dipasang berbagai sistem operasi dalam *kernel* yang sama dapat mengurangi penggunaan RAM dibangun dengan berkas sistem yang berlapis sehingga penggunaan memori menjadi lebih efisien.

1.1 Rumusan Masalah

Berdasarkan penjelasan di atas pada bagian latar belakang, maka penulis merumuskan berbagai masalah yang akan dianalisis dengan penelitian yang akan penulis yaitu:

1. Bagaimana rancangan dan arsitektur *platform* virtualisasi berbasis Docker?
2. Bagaimana keefektifan rancangan arsitektur Docker sebagai *platform* virtualisasi berbasis kontainer.
3. Bagaimana membuktikan kompatibilitas kontainer Docker untuk menjalankan aplikasi yang telah berjalan di atas *platform* sistem operasi yang berbeda.
4. Bagaimana docker dapat digunakan untuk membantu proses kontrol versi sebuah aplikasi.

1.2 Tujuan dan Manfaat Penelitian

1.2.1 Tujuan

Tujuan yang ingin dicapai dalam penulisan tugas akhir ini adalah sebagai berikut:

1. Merancang arsitektur *platform* virtualisasi berbasis docker yang mudah digunakan dan cepat dalam implementasi serta lebih terdistribusi.

2. Menguji efektifitas rancangan arsitektur docker sebagai platform virtualisasi berbasis kontainer.
3. Menguji kompatibilitas Docker sebagai platform virtualisasi untuk mengatasi permasalahan terkait ketergantungan lintas *platform*.
4. Melakukan pengujian kontrol versi untuk aplikasi.

1.2.2 Manfaat Penelitian

Dalam penelitian ini manfaat yang didapatkan adalah sebagai berikut:

1. Memahami cara kerja dan penggunaan docker sebagai salah satu *platform* berbasis kontainer.
2. Menghasilkan karya tugas akhir di bidang virtualisasi berbasis Docker yang dapat menjadi salah satu rujukan karya tulis dan tugas akhir yang bermanfaat bagi dunia akademis dan dunia industri.

1.3 Batasan Masalah

Dalam penelitian ini berikut beberapa batasan masalah di antaranya:

1. Perangkat lunak Docker dipasang pada sistem operasi Ubuntu versi 17.10.
2. Manajemen sistem Docker menggunakan alat yang telah tersedia (*built in*) pada perangkat lunak docker dan alat manajemen *rancher server*.
3. Pada uji kompatibilitas Docker untuk mengatasi permasalahan terkait ketergantungan lintas *platform* hanya menguji sebuah aplikasi web yaitu joomla.
4. Dalam penelitian ini tidak dilakukan pengujian keamanan pada sistem docker.

2. METODE PENELITIAN

Dalam tulisan ini penulis menggunakan metode kualitatif deskriptif dan eksperimental dengan menggunakan perangkat lunak virtualisasi dan membuat simulasi yang disesuaikan dengan kasus aslinya. Adapun teknik pengumpulan datanya dilakukan dengan dua unsur pengumpulan data:

1. Studi pustaka data diperoleh dari tulisan-tulisan dan panduan di internet yang berkaitan dengan masalah yang penulis bahas.
2. Simulasi dilakukan dengan percobaan pembuatan pemodelan dengan berdasarkan panduan yang berkaitan dengan masalah yang dibahas.

3. KEBUTUHAN PERANGKAT LUNAK DAN PERANGKAT KERAS

3.1 Kebutuhan Perangkat Lunak

Tabel 1. Kebutuhan Perangkat Lunak

Sistem Operasi/ Aplikasi	Versi	User Interface	Guest/ Host	Penggunaan
Ubuntu	17.10 Mate	Desktop	Host	Sebagai Host utama dan juga sebagai client.
Docker	1.13.1	CLI	Host	Melakukan pull image dan Menjalankan Kontainer
Portainer	-	CLI	Guest/ Host	Uji Kompabilitas terhadap Perbedaan Lingkungan Host
Centos	7	CLI	Guset/ Host	Uji Kompabilitas terhadap Perbedaan Lingkungan Host/ Distro
Virtualbox	-	Desktop	-	Menjalankan Ubuntu 18.04
NFS Server	-	CLI	Host	Backup data dan sharing File
Joomla, moodle, wordpress	-	Web Application	-	Aplikasi yang akan dijalankan dalam bentuk kontainer

3.2 Kebutuhan Perangkat Lunak

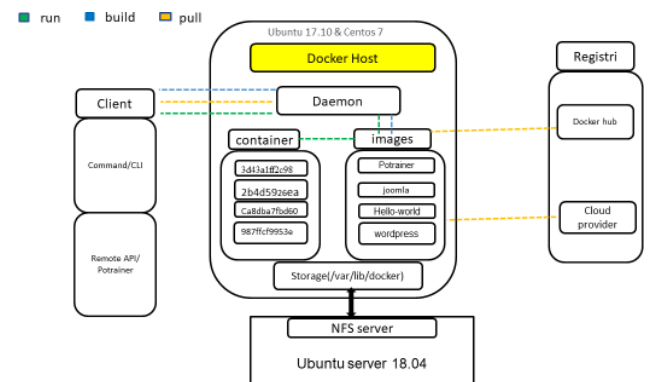
Standard requirement untuk Ubuntu Desktop Edition

1. Processor : 2 GHz *dual core processor*.
2. RAM : 2 GB RAM (*system memory*).
3. Harddisk : 25 GB of *hard-drive space*.
4. VGA : kemampuan resolusi layar 1024x768.
5. CD/DVD atau port USB untuk media instalasi.
6. Koneksi internet.

4. PERANCANGAN SISTEM

4.1 Perancangan Arsitektur Sistem

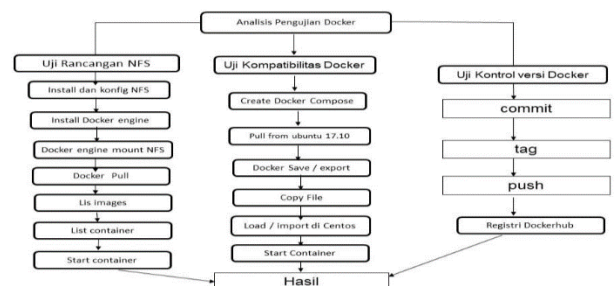
Dalam penelitian ini peneliti akan melakukan rancangan terhadap arsitektur sistem. Arsitektur yang dirancang menggunakan pendekatan rancangan sistem terdistribusi, dimana dilakukan pemisahan antara media penyimpanan (penyimpanan untuk *images* dan file sistem Docker *container*) dari docker engine itu sendiri. Didalam arsitektur umum penyimpanan ini berada dalam satu *host* yang sama dengan Docker *engine*. Namun didalam rancangan arsitektur yang akan dibangun ini, teknik pemisahan media penyimpanan (penyimpanan untuk *images* dan file sistem docker *container*) menggunakan solusi *network file system* (NFS). Ilustrasi mengenai rancangan arsitektur sistem yang akan dibangun di dalam penelitian ini dapat dilihat pada gambar.



Gambar 1. Perancangan Arsitektur Sistem

4.2 Perancangan Pengujian

Pada penelitian ini tiga fokus yang diteliti yaitu penelitian pada efektifitas, uji kontrol versi, dan kompatibilitas docker sebagai *platform* virtualisasi dengan melakukan uji coba secara langsung menggunakan laptop peneliti, kemudian menjalankan docker dan mengakses aplikasi yang telah terpasang. Berikut adalah bagan pengujian yang akan coba penulis rancang.



Gambar 2. Perancangan Pengujian

5. IMPLEMENTASI DAN PENGUJIAN

5.1 Pemasangan NFS-server dan NFS-client

Pada pengujian ini peneliti akan melakukan pemasangan NFS-server pada ubuntu server 18.04 dan NFS-client pada Ubuntu 17.04, dan melakukan remote direktori *file system* Docker yaitu */var/lib/docker* dengan melakukan *mounting* pada *client*, dengan masing-masing IP sebagai berikut:

```
Ubuntu server 18.04 (host server):
192.168.2.xxx Ubuntu 17.10
(client): 192.168.2.xxx
Centos 7 (client): 192.168.2.xxx
```

- a. Langkah pertama yaitu melakukan pemasangan komponen NFS-server dan NFS-client, Pada host server ubuntu 18.04 akan dipasang NFS-kernel-server yang berfungsi untuk melakukan berbagi direktori dengan menggunakan perintah apt pada terminal:

```
Description: Ubuntu 18.04 LTS
Release: 18.04
Codename: bionic
root@almubarok:/home/almubarok# apt-get install nfs-kernel-server
```

Gambar 3. Pemasangan Komponen NFS-server

- b. Kemudian pada client ubuntu 17.10 memasang NFS-common dan pada centos NFS-utils yang berfungsi untuk melakukan remote folder yang berada pada server. Dengan melakukan perintah:

```
root@almubarok:/# sudo apt-get
install nfs-common
```

- c. Membuat direktori folder yang akan di-share yaitu pada home dengan nama folder gudang.
- d. Langkah selanjutnya adalah melakukan konfigurasi NFS-export pada host server dengan mengedit file exports pada direktori /etc/ menggunakan text editor nano, pada file exports tersebut ditambahkan kode yang berisi struktur direktori yang akan di share, alamat ip client yang mengakses, dan hak akses pada folder yang di share, berikut adalah contohnya:

```
root@almubarok:/# nano
```

Dan menambahkan kode di bawah ini:

```
GNU nano 2.9.3 /etc/exports
# /etc/exports: the access control list for filesystems which may be exported
# to NFS clients. See exports(5).
#
# Example for NFSv2 and NFSv3:
# /srv/homes hostname1(rw,sync,no_subtree_check) hostname2(ro,sync,no_subtree_check)
#
# Example for NFSv4:
# gss/krb51(rw,sync,fsid=0,crossmnt,no_subtree_check)
# /srv/nfs4 gss/krb51(rw,sync,no_subtree_check)
#
/home/gudang 192.168.2.0/24(rw,sync,fsid=0,no_root_squash,no_subtree_check)
```

Gambar 4. Konfigurasi /etc/export

Mengecek ulang apakah akses mounting sudah terdapat dalam list dengan perintah:

```
root@almubarok:/#
```

- e. Setelah selesai melakukan pengeditan kemudian di simpan dan jalankan perintah:

```
root@almubarok:/# systemctl
```

- f. Pastikan firewall dalam kondisi tidak aktif dengan menjalankan perintah:

```
root@almubarok:#sudo ufw status
Status : inactive
```

- g. Langkah selanjutnya mount point pada klien, setelah host server sudah dapat melakukan sharing. Maka salin terlebih dulu folder var/lib/docker pada klien ke folder mnt/temp, setelah disalin pada klien, matikan terlebih dahulu docker engine nya dengan perintah:

```
root@almubarok:/# systemctl stop
```

- h. Lakukan mounting pada klien ke NFS-server dengan perintah:

```
Sudo mount
192.168.2.80:/home/gudang
/var/lib/docker
```

- i. Dengan perintah di atas, folder yang telah di-share pada host server sudah dimounting pada client, dapat dicek dengan menjalankan perintah df -h.

```
root@almubarok /# df -h
Filesystem      Size  Used Avail Use% Mounted on
udev             1.8G   0 1.8G   0% /dev
tmpfs            365M   1.6M 363M  1% /run
/dev/sda7        62G  48G  13G  83% /
tmpfs            1.8G  47M  1.8G  3% /dev/shm
tmpfs            5.0M  4.0K 5.0M  1% /run/lock
tmpfs            1.8G   0 1.8G   0% /sys/fs/cgroup
/dev/loop0       87M   87M   0 100% /snap/core/4650
/dev/loop1       87M   87M   0 100% /snap/core/4880
/dev/loop6       109M  109M   0 100% /snap/core/4910/1015
/dev/loop4       7.7M   7.7M   0 100% /snap/pulsemixer/8
/dev/loop2       103M  103M   0 100% /snap/core/4910/1006
/dev/loop5       104M  104M   0 100% /snap/ao/10
/dev/loop7       8.7M   8.7M   0 100% /snap/pulsemixer/1
/dev/loop8       87M   87M   0 100% /snap/core/4917
/dev/loop10      8.0M   8.0M   0 100% /snap/pulsemixer/23
/dev/loop11      94M   94M   0 100% /snap/ao/8
/dev/loop3       102M  102M   0 100% /snap/core/4910/999
/dev/loop9       94M   94M   0 100% /snap/ao/9
tmpfs            365M  48K 365M  1% /run/user/1000
/dev/sda3        180G  83G  95G  48% /media/almubarok/0103F89146731590
192.168.2.80:/home/gudang/ 14G  8.2G  4.4G  66% /var/lib/docker
```

Gambar 5. Menampilkan Mounting Direktori

5.2 Pengujian NFS-server dan Docker

- a. Melakukan uji coba pull image pada klien dan menjalankan container serta mengecek apakah ada pertambah size memori pada direktori NFS-server, dengan menjalankan perintah berikut:

```
root@almubarok:/home/almubarok# docker pull hello-world
Using default tag: latest
latest: Pulling from library/hello-world
9db2ca6ccae0: Pull complete
Digest: sha256:4b8ff392a12ed9ea17784bd3c9a8b1fa3299cac44ca35a85c90c5e3c7afacd
Status: Downloaded newer image for hello-world:latest
root@almubarok:/home/almubarok# docker images
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
joomla          latest   78e01bf0893b  12 days ago  413 MB
hello-world     latest  2cb0d9787c4d  13 days ago  1.85 KB
mysql           5.6     97f0bd065c6a  3 weeks ago  256 MB
portainer/portainer latest  7af7abce5f    4 weeks ago  57 MB
```

Gambar 6. Uji Coba Pull Image

Pada gambar di atas telah diketahui bahwa images hello-world berhasil diunduh. Setelah pull berhasil kemudian jalankan kontainer hello-world misalkan dengan nama uhuy dengan perintah:

```
root@almubarok:# docker run -it
--name uhuy hello-world
```


Maka container akan menampilkan teks hello--world pada CLI sebagai berikut:

```
root@almubarak:/home/almubarak# docker run -it --name uhuy hello-world
Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
   (amd64)
3. The Docker daemon created a new container from that image which runs the
   executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it
   to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/engine/userguide/
```

Gambar 7. Tampilan Hello World pada CLI

b. Kemudian cek kembali, apakah file sistem pada NFS-server sudah ada penambahan size dengan perintah:

```
root@nfsserver:#du -sh /home/gudang
```

```
root@nfsserver:/home/nfsserver# du -sh /home/gudang/
3.6G /home/gudang/
root@nfsserver:/home/nfsserver# _
```

Gambar 8. Tampilan Hasil Pengecekan pada CLI

Dari gambar di atas dapat diketahui bahwa *image* yang telah di pull pada klien berhasil terpasang pada NFS-server pada ubuntu server 18.04 dengan bertambahnya size pada direktori */home/gudang*. Dengan demikian file system Docker engine sudah berhasil terpasang pada NFS-server.

5.3 Uji Kompatibilitas Docker terhadap Perbedaan Sistem Operasi

Docker *engine* yang telah terpasang pada sistem operasi tersebut, berikut adalah langkah-langkah pengujian kompatibilitas docker terhadap perbedaan sistem:

a. Melakukan *pull image* Wordpress pada Ubuntu 17.10 menggunakan docker compose dengan langkah-langkah sebagai berikut:

1. Membuat file docker compose dengan ekstensi, yml, dengan kode program.
2. Selanjutnya menjalankan perintah docker-compose up maka image wordpress otomatis diunduh seperti gambar di bawah ini:

```
root@almubarak:/home/wordpress# docker-compose up
Creating network "wordpress_default" with the default driver
Pulling wordpress (wordpress:latest)...
latest: Pulling from library/wordpress
be8881be8156: Pull complete
```

Gambar 9. Docker *compose-up*

Dari gambar diatas dapat diketahui bahwa pull image menggunakan Docker *compose* telah berhasil, dapat dibuktikan dengan perintah selanjutnya.

b. Mengecek apakah images sudah berhasil unduh

dengan perintah:

```
root@almubarak:/# docker images
root@almubarak:/# docker
ps -a
```

Jika kontainer dalam keadaan *exited* maka perlu menjalankan Docker *start*.

c. Save image dan exort kontainer wordpress pada ubuntu 17.10 save image wordpress dan mysql:5.7 dengan perintah:

```
root@almubarak:/home# sudo
docker save wordpress >
wordpress1.tar
root@almubarak:/home# sudo
docker save mysql:5.7 >
mysql1.tar
```

Kemudian cek pada direktori *home*, hasil dari *save image* yang berektensi *wordpress1.tar* dan *mysql1.tar*.salin yang file telah disave dan diexport kedalam folder home yang berada pada sistem operasi centos kemudian jalankan perintah load dan import:



Gambar 10. File Hasil *Save* dan *Export*

d. Kemudian melakukan load image dan cek hasilnya dengan list images.

```
[root@b207 sttnf]# docker load < /home/sttnf/Downloads/wordpress1.tar
[root@b207 sttnf]# docker load < /home/sttnf/Downloads/mysql1.tar
[root@b207 sttnf]# docker images
REPOSITORY          TAG             IMAGE ID        CREATED
VIRTUAL SIZE
wordpress           latest         75051e328ed2   4 days ago
408.1 MB
mysql                5.7            cdecc0a0b97    8 days ago
371.9 MB
[root@b207 sttnf]# █
```

Gambar 11. Pengecek Hasil *List Image*

Dari gambar diatas dapat diketahui bahwa image yang telah disave pada ubuntu berhasil di-load pada sistem operasi centos dan membuktikan bahwa Docker mampu mengatasi masalah perbedaan sistem operasi dalam hal menjalankan aplikasi (*conflicting dependencies*).

5.4 Uji Kontrol Versi Aplikasi pada Joomla

a) Mengecek image id yang akan dilakukan kontrol versi

```
root@almubarak:/# docker images
```

```
root@almubarak:/# docker ps
```

- b) Melakukan commit image joomla dengan perintah:

```
root@almubarok:# docker
commit 0319dc55ef03
almubarok95/joomla:07.18
```

- c) Melakukan docker tag untuk versioning pada images dengan perintah:

```
root@almubarok:# docker tag
78e01bfb093b
almubarok95/joomla:07.18
```

- d) Setelah melakukan tag lalu melakukan push ke docker hub, login terlebih dahulu pada akun *dockerhub*-nya dengan perintah docker login:

```
root@almubarok:# docker
login Username
(almubarok95):almubarok95
Password:
Login Succeeded
```

- e) Melakukan push image Joomla Docker dengan perintah docker push:

```
root@almubarok:# docker push
almubarok95/joomla:07.18
```

- f) Kemudian melakukan pengecekan dengan login pada situs *dockerhub* melalui browser.

6. KESIMPULAN DAN SARAN

6.1 Kesimpulan

Dengan hasil pengujian pada bab sebelumnya pengujian mendapatkan kesimpulan sebagai berikut:

1. Rancangan virtualisasi berbasis Docker yang diuji yaitu menggunakan NFS-*server* memisahkan antara file sistem Docker dengan Docker *engine* dimana Docker *engine* berada pada dua komputer klien dan file sistem Docker tersimpan pada komputer *server* yang telah dipasang NFS-*server*, file sistem Docker yang berada pada direktori `/var/lib/docker` di-*mounting* pada NFS *server* yang berada pada direktori `/home/gudang`.
2. Efektifitas rancangan *platform* virtualisasi berbasis docker menggunakan NFS-*server* berjalan dengan baik, serta dalam proses pengembangan aplikasi menjadi lebih terdistribusi, dapat mengatasi masalah

jika ada salah satu docker engine yang gagal aktif (*crash*).

3. Kompatibilitas Docker dalam mengatasi perbedaan pada sistem operasi sangat baik itu buktikan dengan melakukan *export* dan *import* kontainer Docker menggunakan sistem operasi ubuntu 17.10 dan centos6.
4. Uji kontrol versi aplikasi menggunakan Docker sangat efisien dan mudah dilakukan dengan waktu relatif lebih singkat.
5. Manajemen Docker menggunakan portainer memberikan kemudahan dalam hal pengelolaan pada setiap komponen-komponen docker yang terpasang.

6.2 Saran

Dalam penulisan ini masih banyak kekurangan dalam implementasinya, berikut adalah saran untuk penelitian selanjutnya:

1. Perlu adanya penelitian lebih lanjut mengenai bagaimana *high availability* pada NFS-*server* dalam menjalankan aplikasi berbasis kontainer.
2. Dalam rancangan ini peneliti menjalankan Docker pada jaringan lokal, Docker dapat dimanfaatkan ketahap *platform multi-cloud* sebagai wadah komputasi awan.

DAFTAR PUSTAKA

- [1] Goasguen, Sebastien, “*Docker Cookbook*,” United States of America: O’Reilly Media, Inc., 2015.
- [2] “Dasar Manajemen Docker menggunakan Portainer,” Cilsy Fiolotuion Indonesia, 2017.
- [3] R. Mckendrick, S. Gallagher, “*Mastering Docker*,” *second edition*, Birmingham: Packtpublishing Ltd., 2017.
- [4] V. Sugianto, R. Pratama, “*Virtualisasi Modern berbasis Docker*,” Bekasi: Penerbit PT Excellent Infotama Kreasindo, 2016.
- [5] Javacodegeeks, “*Docker Containerization Cookbook*,” Exelixis Media P.C, 2016.
- [6] “*Ubuntu is the numberone for container*,” ubuntu.com 2017, [Online]. Available: <https://www.ubuntu.com/containers>
- [7] James Turnbull, “*The Docker Book: Containerization is a the new Virtualization*,” Melbourne, Australia, 2017.



RANCANG BANGUN *WEBSITE CUSTOMER RELATIONSHIP MANAGEMENT* (CRM) PADA MODUL AKUNTANSI STUDI KASUS: CV ESINDO MULTI TATA

Muhammad Rizky Arinugraha Putra¹, Sirojul Munir²

^{1,2}Teknik Informatika, Sekolah Tinggi Teknologi Terpadu Nurul Fikri
Jakarta Selatan, DKI Jakarta, Indonesia 12640
rizkyarinugraha.putra@gmail.com , rojulman@gmail.com

Abstract

CV Esindo Multi Tata is a consulting service company in taxation, accounting, and payroll systems. In managing its customers, CV Esindo Multi Tata already has a system called Esi-CRM, a web-based application with four main modules, namely the marketing module, the customer module, the operational module, and the reporting module. In its development, CV Esindo Multi Tata wants to add an accounting module to automate the journal process for transactions contained in the customer and operational modules. In this study, the analysis and design of the accounting module in the Esi-CRM application was carried out using the waterfall software development method and UML diagrams. The results of testing the application prototype using black-box testing stated that the accounting module was running 100% well, and the results of the user acceptance test (UAT) had a feasibility value of 94.28%, or the application module was declared accepted by the user and could be used.

Keywords: Accounting, CRM, Waterfall, UML, Web Application

Abstrak

CV Esindo Multi Tata adalah perusahaan jasa konsultan bidang perpajakan, akuntansi dan sistem payroll. Dalam pengelolaan pelanggannya CV Esindo Multi Tata telah memiliki sistem bernama Esi-CRM, yaitu aplikasi berbasis web yang memiliki empat modul utama yaitu modul marketing, modul pelanggan, modul operasional dan modul pelaporan. Pada perkembangannya CV Esindo Multi Tata ingin menambahkan modul akuntansi untuk otomatisasi proses jurnal pada transaksi yang terdapat pada modul pelanggan dan operasional. Pada penelitian ini dilakukan analisis dan perancangan modul akuntansi pada aplikasi Esi-CRM dengan menggunakan metode pengembangan software waterfall dan diagram UML. Hasil pengujian prototype aplikasi menggunakan black box testing dinyatakan modul akuntansi sudah 100% berjalan dengan baik, dan hasil pengujian penerimaan user (UAT) memiliki nilai kelayakan sebesar 94.28% atau modul aplikasi dinyatakan diterima user dan dapat digunakan.

Kata kunci: Akuntansi, CRM, Waterfall, UML, Aplikasi Web

1. PENDAHULUAN

Sistem informasi akuntansi merupakan bagian dari sistem yang ada dalam sebuah organisasi atau perusahaan, penerapan sistem informasi akuntansi diharapkan dapat membantu proses administrasi keuangan, sehingga perusahaan tersebut dapat lebih baik dari kompetitor lainnya yang bergerak di bidang yang sama.

CV Esindo Multi Tata didirikan di Jakarta pada 1 Februari 2011, adalah perusahaan yang bergerak dalam bidang pelatihan dan pengembangan sumber daya manusia, dengan fokus pada bidang perpajakan, akuntansi, sumber daya manusia, teknologi informatika dan pengembangan software penghitungan penggajian/*payroll system* [1].

Saat ini CV Esindo Multi Tata sudah memiliki sistem informasi untuk pengelolaan hubungan pelanggan atau lebih dikenal sebagai *Customer Relationship Management* (CRM) berbasis web bernama Esi-CRM dan mempunyai beberapa modul diantaranya adalah: modul marketing, modul pelanggan, modul operasional dan modul laporan.

CV Esindo Multi Tata saat ini menginginkan adanya sebuah modul akuntansi untuk aplikasi Esi-CRM yang mampu membantu pengguna dalam melakukan proses pencatatan akuntansi secara otomatis, dikarenakan saat ini perhitungan akuntansi masih dilakukan dengan cara manual yaitu menggunakan *excel*.

Berdasarkan latar belakang tersebut, maka dalam tugas akhir yang berjudul “Analisis dan Perancangan Modul Akuntansi Pada Aplikasi *Customer Relationship Management (CRM)* Berbasis Web Studi Kasus: CV Esindo Multi Tata” Penulis akan melakukan analisis dan perancangan sebuah modul akuntansi yang dapat menunjang proses bisnis yang ada pada aplikasi Esi-CRM.

1.1 Rumusan Masalah

1. Fitur apa saja yang dibutuhkan pada perancangan modul akuntansi untuk aplikasi Esi-CRM?
2. Apakah modul akuntansi pada aplikasi Esi-CRM mempunyai manfaat bagi perusahaan?

1.2 Tujuan dan Manfaat Penelitian

Tujuan dari penelitian ini adalah:

1. Dapat membuat dan mengintegrasikan modul akuntansi dengan aplikasi Esi-CRM.
2. Mengetahui dan menerapkan fitur-fitur yang dibutuhkan pada modul akuntansi.

Adapun manfaat dari penelitian ini adalah:

1. Tersedianya modul akuntansi pada aplikasi Esi-CRM yang dapat digunakan oleh staf admin.
2. Memudahkan staf admin dalam melakukan pembuatan jurnal secara otomatis dari data *invoice* yang ada pada aplikasi Esi-CRM.

1.3 Batasan Masalah

1. Penelitian dilakukan pada aplikasi Esi-CRM yang sudah berjalan di CV Esindo Multi Tata.
2. Peneliti hanya akan membahas proses bisnis yang ada pada modul akuntansi.
3. Peneliti hanya akan membuat fitur-fitur pada modul akuntansi sesuai dengan kebutuhan CV Esindo Multi Tata.

2. TINJAUAN PUSTAKA

2.1 *Customer Relationship Management (CRM)*

Menurut Rainer dan Cegielski, CRM adalah strategi organisasi yang berfokus pada pelanggan dan berorientasi pelanggan. Yaitu, organisasi berkonsentrasi pada penilaian kebutuhan pelanggan akan produk dan jasa dan kemudian memberikan kualitas yang tinggi dan pelayanan responsif. CRM membangun hubungan pelanggan jangka panjang yang sifatnya jangka panjang yang menciptakan nilai bagi perusahaan serta untuk pelanggan [2].

Satzinger, Jackson, Burd menambahkan, CRM adalah sebuah sistem yang mendukung kegiatan pemasaran (*marketing*), penjualan (*sales*), dan operasi jasa (*service operations*) yang melibatkan *direct customer interactions* dan *indirect customer interactions* [3].

Esi-CRM adalah program CRM yang dikembangkan oleh CV Esindo Multi Tata untuk membantu kegiatan operasional kantor. Aplikasi ini dikembangkan menggunakan *Yii2 Framework* dimana di dalamnya terdiri dari beberapa modul seperti modul marketing, modul pelanggan, modul operasional dan modul laporan.

2.2 Sistem Akuntansi

Menurut Mahatmyo salah satu sistem yang berkaitan dengan perusahaan adalah sistem akuntansi. Pembahasan mengenai sistem akuntansi tidak kalah pentingnya dengan pembahasan bidang akuntansi yang lain. Tanpa sistem akuntansi yang memadai tidak mustahil kegiatan operasional perusahaan akan terganggu. Sistem akuntansi merupakan suatu rangkaian bukti transaksi, dokumen, catatan-catatan akuntansi dan laporan-laporan serta alat-alat prosedur, kebijakan, sumber daya manusia maupun sumber daya lain dalam suatu perusahaan yang di koordinasikan sedemikian rupa untuk mendukung dalam pencapaian apa yang menjadi tujuan perusahaan [4].

Menurut Shatu akuntansi ialah pengukuran, penjabaran, atau pemberian kepastian mengenai informasi yang akan membantu manajer, investor, otoritas pajak dan pembuat keputusan lain untuk membuat alokasi sumber daya keputusan di dalam perusahaan, organisasi, dan lembaga pemerintah [5].

Jadi dapat disimpulkan sistem akuntansi adalah serangkaian proses pencatatan, pengklasifikasian, pengikhtisaran dan mengelola menjadi laporan yang membantu memberikan informasi untuk pihak yang berkepentingan.

2.3 *Waterfall*

Menurut Pressman, metode *waterfall* adalah metode yang memiliki tahapan-tahapan, yang dibagi menjadi 5 tahapan. Berikut tahapan dari metode tersebut: [6]

1. *Communication*
2. *Planning*
3. *Modeling*
4. *Construction*
5. *Deployment*

2.4 *Unified Modelling Language (UML)*

Menurut Gooma, *Unified Modeling Language (UML)* adalah pemodelan grafis dan notasi untuk menggambarkan pemodelan berorientasi objek. UML digunakan untuk menggambarkan perangkat lunak, analisis dan model desain. Adapun jenis-jenis diagram UML yaitu: [7]

1. *Use Case Diagram*
2. *Class Diagram*
3. *Sequence Diagram*
4. *Collaboration Diagram*

5. *Activity Diagram*
6. *Component Diagram*
7. *Object Diagram*

2.5 Black Box Testing

Black box testing disebut juga sebagai *behavioral testing* atau *functional testing*. *Black box testing* memungkinkan tester untuk memperoleh hasil berupa kumpulan input yang berhasil menjalankan seluruh kebutuhan fungsional dari program [8].

2.6 User Acceptance Test (UAT)

UAT adalah software testing yang dijalankan pada calon pengguna sebelum aplikasi di rilis. Tujuan utama dari UAT adalah untuk meyakinkan bahwa sistem baru sudah sesuai dengan kebutuhan yang sudah ditentukan [9].

2.7 Skala Likert

Skala likert digunakan untuk mengukur sikap, pendapat, dan persepsi seseorang atau sekelompok orang tentang fenomena sosial. Dalam penelitian fenomena sosial ini telah ditetapkan secara spesifik oleh peneliti, yang selanjutnya disebut sebagai variabel penelitian. Dengan skala likert, maka variabel yang akan diukur dijabarkan menjadi indikator variabel [10].

3. METODE PENELITIAN

3.1 Metode Pengembangan

2.1 Metode pengumpulan data, instrumen penelitian, dan metode pengujian

Pada tahap pengembangan ini penulis menjadi 5 tahap, yaitu:

1. **Kebutuhan**
Dilakukan proses pengumpulan kebutuhan dari pengguna pada bagian keuangan CV Esindo Multi Tata, melalui proses wawancara. Pada tahap ini jika terdapat kebutuhan tambahan akan langsung masuk ke dalam tahap ini.
2. **Analisis**
Proses analisis terhadap kebutuhan pengguna yang telah dikumpulkan sebelumnya, tujuan dari tahap ini adalah memastikan bahwa seluruh kebutuhan pengguna telah terkumpul.
3. **Desain**
Pendefinisian kebutuhan pengguna tersebut ke dalam bentuk desain dengan menggunakan UML
4. **Implementasi**
Dimana pada fase ini dilakukan proses implementasi dari desain yang telah dibuat ke dalam kode program menggunakan *Yii2 Framework* dan *database MySQL* hingga siap untuk dilakukan pengujian.
5. **Pengujian**
Setelah pembuatan kode program selesai maka dilakukan proses pengujian. Pengujian yang

dilakukan menggunakan metode *black box* untuk menguji tiap fungsionalitas menu. Jika modul yang dibuat sudah melalui proses penerimaan, maka modul sudah siap untuk diintegrasikan.

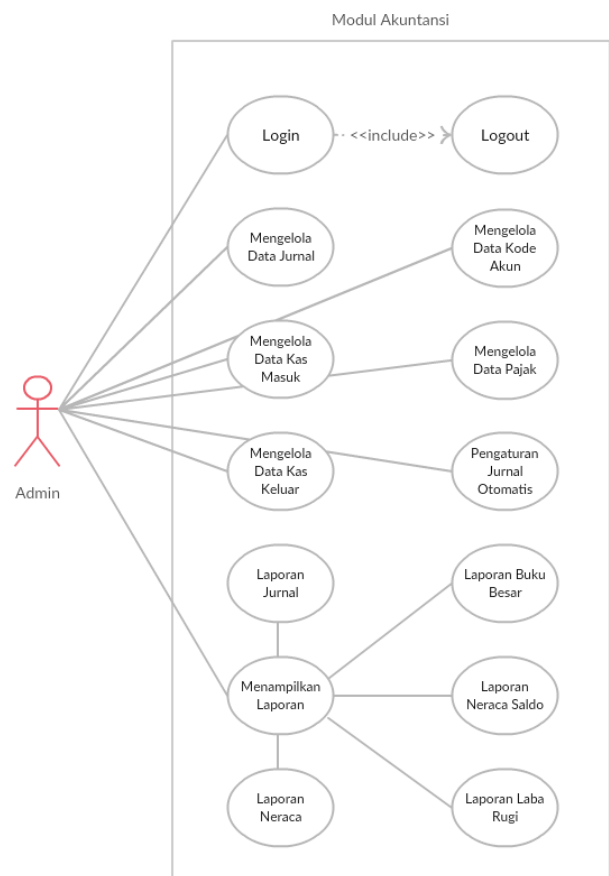
3.2 Metode Pengumpulan Data

Metode pengumpulan data digunakan untuk memperoleh informasi dan data sebagai acuan rancangan untuk menyelesaikan masalah yang penulis ambil. Adapun metode pengumpulan data yang digunakan dalam penelitian ini yaitu studi literatur dan wawancara.

4. ANALISIS DAN PERANCANGAN

4.1 Analisis Sistem

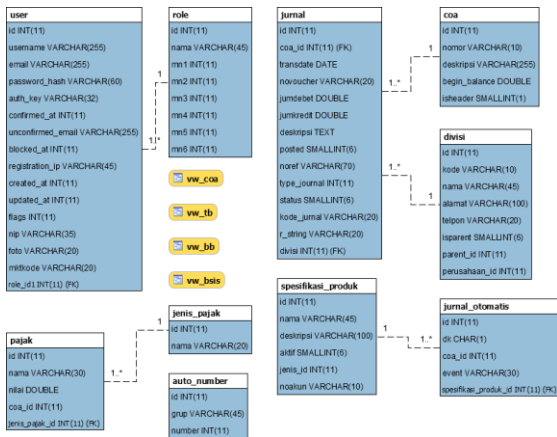
Pada penelitian ini dilakukan proses pengumpulan data kebutuhan pengguna dengan melakukan proses wawancara dengan beberapa orang di CV Esindo Multi Tata untuk mengetahui kebutuhan apa saja yang diperlukan untuk pengembangan modul akuntansi. Adapun hasil wawancara tersebut dapat disimpulkan sebagai berikut:



Gambar 1. Use Case

4.2 Perancangan Sistem

Pada penelitian ini dibuat domain model yang berfungsi untuk menunjukkan hubungan yang mungkin ada diantara objek-objek yang sangat kompleks, secara umum objek diagram berisi dengan objek dan link.

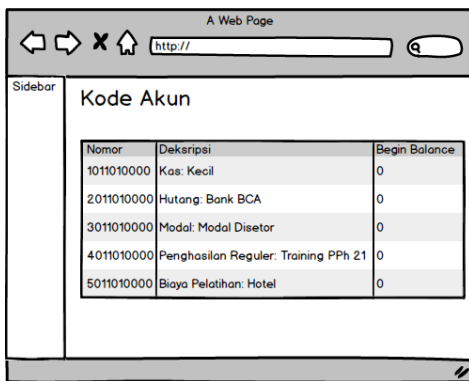


Gambar 2. Domain Model

4.3 Desain Antar Muka

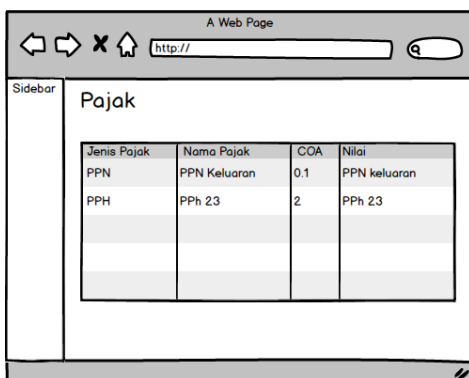
Rancangan *mockup* modul akuntansi dibuat untuk memberi gambaran terhadap modul akuntansi yang nantinya akan dibangun. Berikut ini beberapa pokok-pokok *mockup* modul akuntansi yang penulis buat:

1. Halaman Kode Akun
Merupakan *mockup* halaman kode akun, dimana admin dapat melihat, mencari, mengedit dan menghapus data kode akun yang telah dibuat.



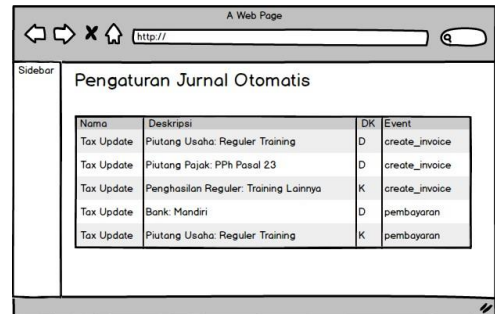
Gambar 3. Mockup List Data Kode Akun

2. Halaman Pajak
Merupakan *mockup* halaman pajak, dimana admin dapat melihat, mencari, mengedit dan menghapus data pajak yang telah dibuat.



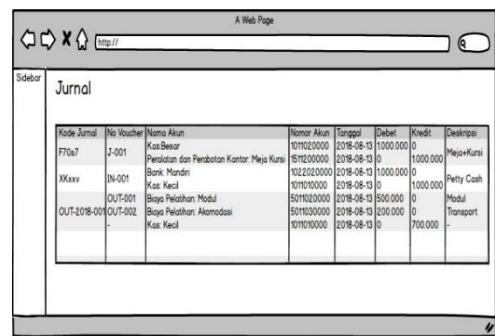
Gambar 4. Mockup List Data Pajak

3. Halaman Pengaturan Jurnal Otomatis
Merupakan *mockup* halaman pengaturan jurnal otomatis, dimana admin dapat melihat, mencari, mengedit dan menghapus data pengaturan jurnal otomatis yang telah dibuat.



Gambar 5. Mockup List Data Pengaturan Jurnal Otomatis

4. Halaman Jurnal
Merupakan *mockup* halaman pengaturan jurnal, dimana admin dapat melihat, mencari, mengedit dan menghapus data jurnal yang telah dibuat.



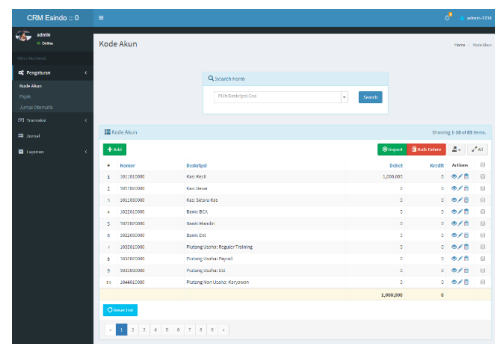
Gambar 6. Mockup List Data Jurnal

5. IMPLEMENTASI

5.1 Implementasi Antarmuka

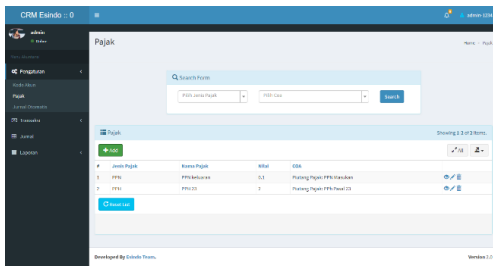
Pada tahap ini merupakan implementasi dari desain yang telah dibuat pada tahap sebelumnya. Berikut ini implementasi antar muka dari modul akuntansi yang sudah dibuat:

1. Halaman Kode Akun



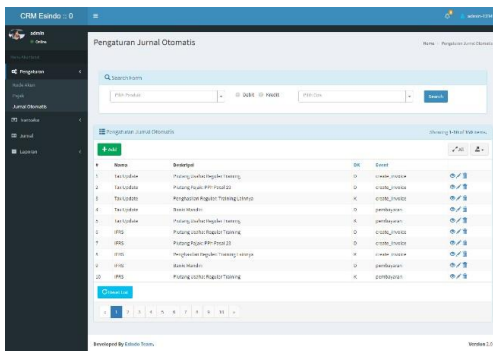
Gambar 7. Halaman List Data Kode Akun

2. Halaman Pajak



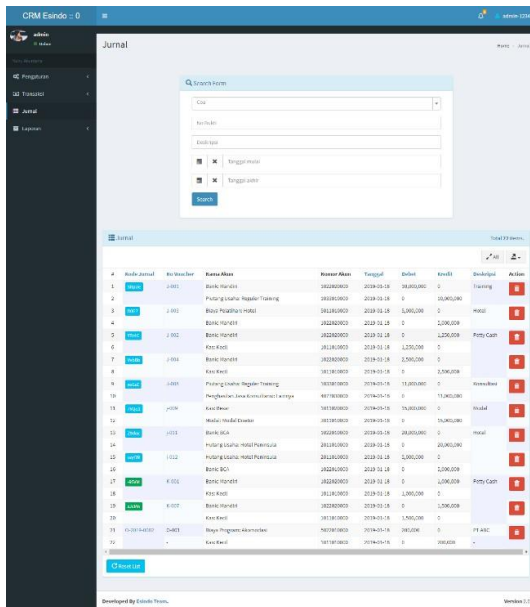
Gambar 8. Halaman List Data Pajak

3. Halaman Pengaturan Jurnal Otomatis



Gambar 9. Halaman List Data Pengaturan Jurnal Otomatis

4. Halaman Jurnal



Gambar 10. Halaman List Data Jurnal

5.2 Hasil Kuesioner

Tabel 1. Hasil Kuesioner

No	Pertanyaan	Nilai				
		STS	TS	CS	S	SS
1	Apakah fitur pada modul akuntansi sudah sesuai dengan kebutuhan?	0	0	0	1	6
2	Apakah modul akuntansi dapat memudahkan anda dalam pembuatan jurnal?	0	0	0	3	4
3	Apakah modul akuntansi dapat memudahkan anda dalam pembuatan laporan akuntansi?	0	0	0	2	5
4	Apakah modul akuntansi ini sudah terintegrasi dengan baik pada aplikasi Esi-CRM?	0	0	0	2	5
Jumlah		0	0	0	8	20
Jumlah skor observasi		0	0	0	32	100
Σ Skor		132				
Presentase (%)		94.28%				

Total skor observasi dari hasil kuesioner sejumlah 132 (94.28%) dari skor yang diharapkan yaitu 140 (100%).

6. KESIMPULAN

6.1 Kesimpulan

- Adapun fitur yang dibutuhkan pada perancangan modul akuntansi untuk aplikasi Esi-CRM yaitu Kode Akun, Pajak, Jurnal Otomatis, Jurnal, Kas, Laporan Jurnal, Laporan Neraca Saldo, Laporan Buku Besar, Laporan Neraca dan Laporan Laba Rugi.
- Setelah dilakukan pengujian fungsional modul akuntansi menggunakan *black box testing* dan UAT, dinyatakan modul akuntansi sudah 100% berjalan dengan baik dan dari hasil analisis data kuesioner menggunakan pengukuran skala likert untuk kelayakan modul akuntansi, didapat presentase kelayakan sebesar 94.28%. Maka dapat disimpulkan modul akuntansi pada aplikasi Esi-CRM sangat bermanfaat dalam pengerjaan akuntansi di CV Esindo Multi Tata.

6.2 Saran

Pada penelitian ini, penulis hanya berfokus kepada fitur-fitur utama yang dibutuhkan oleh CV Esindo Multi Tata. Maka dari itu penulis menyarankan untuk pengembangan selanjutnya agar dapat menambahkan fitur-fitur pendukung yaitu impor data pengaturan jurnal otomatis, impor data jurnal, neraca lajur dan juga dapat menerapkan saran yang ada pada hasil kuesioner.

DAFTAR PUSTAKA

- Esindo Multi Tata, "Tentang Kami," 26 Mei 2018. [Online]. Available: <http://esindo.net/tentang-esindo/>. [diakses Maret 5 2018]
- C. Rainer, "Introduction to Information Systems," John Wiley & Sons, Inc, 2013.
- J. W. J. R. B. B. S. D. Satzinger, "Systems Analysis and Design in a Changing World," Course Technology, 2009.
- A. Mahatmyo, "Sistem Informasi Akuntansi-Suatu Pengantar," Deepublish, 2015.

- [5] Y. P. Shatu, "Kuasai Detail Akuntansi Laba dan Rugi," Pustaka Ilmu Semesta, 2016.
- [6] R. S. Pressman, "Approach, Software Engineering: A Practitioner's," McGraw-Hill Education, 2010.
- [7] H. Gomaa, "Software Modeling and Design," Cambridge University Press, 2011.
- [8] B. M. Roger S. Pressman, "Software Engineering: A Practitioner's Approach," McGraw-Hill Education, 2014.
- [9] P. v. G. Brian Hambling, "User Acceptance Testing: A Step-by-step Guide," BCS Learning & Development Limited, 2013.
- [10] Sugiyono, "Metode Penelitian Pendidikan," Bandung: ALFABETA, 2011.



RANCANG BANGUN SISTEM *MONITORING BUDGET* PADA *EVENT ORGANIZER* BERBASIS *WEBSITE* MENGGUNAKAN *MVC FRAMEWORK*

Muhamad Tarmizi¹, Sirojul Munir²

^{1,2}Teknik Informatika, Sekolah Tinggi Teknologi Terpadu Nurul Fikri
Jakarta Selatan, DKI Jakarta, Indonesia 12640

muhamadtarmizi03@gmail.com, rojulman@gmail.com

Abstract

PT. INDI Notokreasi is a company engaged in the Integrated Marketing Communication Agency. Along with the development of information, PT. INDI Notokreasi requires an integrated information system in monitoring every work activity. The recording of work activities in INDI COMMUNICATION is still using the old method so that the data stored is still not arranged efficiently and is well integrated. Therefore, a Budget Event Organizer Monitoring application was built at PT. INDI Notokreasi website based. This application was built using Laravel PHP Framework and MYSQL database, while for system design, UML diagrams were used. The result of this research is a web-based application with several features, including project margin estimation, invoice scheduling, project expense input, upload payment notes, mark completed projects, mark completed invoices, edit project data, edit invoice data, delete the wrong project, delete a wrong invoice, view project details, view invoice details, view project expense details.

Keywords: *PT. INDI Notokreasi, Website, Monitoring Budget Event Organizer*

Abstrak

PT. INDI Notokreasi merupakan salah satu perusahaan yang bergerak di bidang *Integrated Marketing Communication Agency*. Seiring perkembangan informasi, PT. INDI Notokreasi memerlukan sistem informasi yang terintegrasi dalam memantau setiap aktifitas kerja. Pencatatan aktifitas kerja di INDI COMMUNICATION masih menggunakan cara lama, sehingga data-data yang tersimpan masih belum tersusun secara efisien dan terintegrasi dengan baik. Oleh karena itu, dibangun aplikasi *Monitoring Budget Event Organizer* pada PT. INDI Notokreasi berbasis website. Aplikasi ini dibangun menggunakan PHP Framework Laravel dan database MYSQL, sedangkan untuk perancangan sistem digunakan UML Diagram. Hasil penelitian ini adalah sebuah aplikasi berbasis web dengan beberapa fitur, antara lain: estimasi margin proyek, penjadwalan invoice, input pengeluaran proyek, unggah nota pembayaran, tandai proyek yang telah selesai, tandai invoice yang telah selesai, edit data proyek, edit data invoice, delete proyek yang salah, delete invoice yang salah, melihat detail proyek, melihat detail invoice, melihat detail pengeluaran proyek.

Kata Kunci: *PT. INDI Notokreasi, Website, Monitoring Budget Event Organizer*

1. PENDAHULUAN

Website menjadi salah satu media untuk mempermudah komunikasi antar sistem. Salah satu dasar dari pembangunan dan pengembangan sistem berbasis *website Monitoring Budget Event Organizer* adalah menggunakan *Framework*. *Framework* adalah kumpulan perintah atau fungsi dasar yang membentuk aturan-aturan tertentu dan saling berinteraksi satu sama lainnya sehingga dalam pembuatan aplikasi *website* harus mengikuti aturan dari *framework* tersebut. INDI COMMUNICATION merupakan salah satu perusahaan yang bergerak di bidang

Integrated Marketing Communication Agency. Seiring perkembangan informasi, INDI COMMUNICATION memerlukan sistem informasi dalam memantau setiap aktifitas kerja. Saat ini, pencatatan aktifitas kerja di INDI COMMUNICATION belum tersusun secara efisien dan belum terintegrasi dengan baik. Pengolahan data pengeluaran dalam bentuk berkas fisik yang dilakukan oleh divisi keuangan masih dapat menimbulkan indikasi rusak dan kehilangan. Dari latar belakang di atas, penulis melakukan penelitian pembangunan aplikasi *Monitoring Budget Event Organizer* berbasis *website*.

1.1 Rumusan Masalah

Dari uraian yang terdapat di latar belakang. Maka peneliti merumuskan masalah dari pengembangan aplikasi, yaitu:

1. Apakah dengan membangun aplikasi untuk pencatatan bagian keuangan dengan teknologi *framework* berbasis *website* dapat memberi kemudahan untuk memantau keuangan perusahaan?
2. Apakah membangun sistem perangkat lunak dengan teknologi *framework* berbasis *website* akan menjamin akses informasi lebih cepat dan akurat?

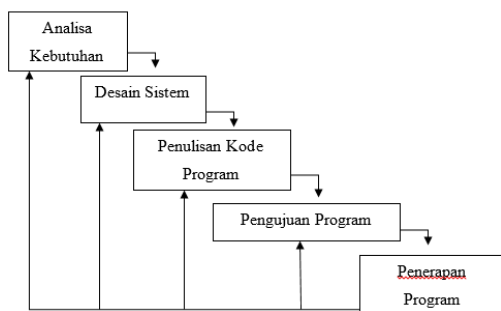
1.2 Tujuan Penelitian

Tujuan yang hendak dicapai dalam penelitian dan penyusunan tugas akhir ini adalah:

1. Membuat aplikasi yang dapat digunakan divisi *account* dan divisi keuangan untuk memudahkan pencatatan *budget* proyek.
2. Menjadi solusi untuk menyelesaikan permasalahan efektifitas perhitungan dan akurasi *budget* proyek di perusahaan INDI COMMUNICATION.

2. METODE PENELITIAN

Metode yang digunakan dalam penelitian ini yaitu dengan menerapkan metode *Waterfall*. Model SDLC (*Systems Development Life Cycle*) seperti air terjun ini (*waterfall*) juga sering disebut dengan sebutan model sekuensial linier (*sequential linear*) atau alur hidup klasik (*classic life cycle*). Metode *waterfall* juga merupakan model pengembangan sistem informasi yang sistematis, dan sekuensial [1]. Disebut dengan *waterfall* karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan.



Gambar 1. Model SDLC

3. TINJAUAN PUSTAKA

3.1 Framework

Framework dapat diartikan sebagai sebuah kerangka kerja yang digunakan untuk mempermudah penyelesaian sebuah pekerjaan. Jika dikaitkan dengan PHP (*Hypertext Preprocessor*) maka dapat diartikan sebagai suatu kerangka kerja yang telah terpola dengan tujuan memudahkan pengembang *web* dalam menyusun

rangkain proses dengan bahasa pemrograman PHP. Misalnya dalam membuat sebuah aplikasi *website*, pengembang harus menuliskan kode program secara berulang dengan kompleksitas tinggi. Hal tersebut tentunya akan sangat mengurangi tingkat efisien waktu serta menguras pikiran, bahkan kode yang dihasilkan mungkin saja kurang optimal dari segi penulisan serta proses yang dilakukan dalam program. Dengan adanya *Framework* akan dapat memudahkan proses, karena *framework* sudah memiliki kerangka kerja tersebut, sehingga tidak perlu menulis kode program dari awal [2].

3.2 Laravel

Laravel adalah sebuah *Framework* PHP yang dirilis di bawah lisensi MIT, dibangun dengan konsep MVC (*Model View Controller*). Laravel merupakan *framework* aplikasi *web* yang ekspresif dan sintaks yang elegan. Laravel juga merupakan sebuah *framework* dari kumpulan program kontrol, sistem migrasi yang ekspresif dan dukungan *tools* yang dibutuhkan dalam menguji aplikasi yang terintegrasi dengan beberapa aplikasi lainnya [3].

3.3 MVC

MVC (*Model View Controller*) adalah suatu metode yang digunakan untuk mengantisipasi tercampur aduknya kode program yang mengakibatkan jika adanya perubahan di tahap selanjutnya, maka tidak bisa diuraikan karena saling terkait dan berpengaruh ke sistem lain. MVC adalah sebuah pendekatan perangkat lunak yang memisahkan aplikasi logika dari presentasi (tampilan). MVC memisahkan aplikasi berdasarkan komponen komponen aplikasi, seperti: manipulasi data, *controller*, dan *user interface* [4].

3.4 Black Box Testing

Menurut Soetam Rizky, berpendapat bahwa "*Black box testing* adalah tipe *testing* dengan memperlakukan perangkat lunak yang tidak diketahui kinerja internalnya. Sehingga para *tester* memandang perangkat lunak seperti layaknya sebuah "kotak hitam" yang tidak penting dilihat isinya, tapi cukup dikenai proses *testing* di bagian luar" [5]. Berdasarkan beberapa pendapat yang dikemukakan diatas dapat ditarik kesimpulan bahwa metode pengujian *Black box* digunakan untuk menguji sistem dari segi *user* yang dititikberatkan pada pengujian kinerja, spesifikasi dan antarmuka sistem tersebut tanpa menguji kode program yang ada [6].

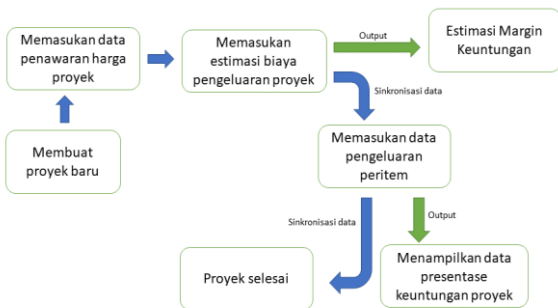
4. URAIAN PENELITIAN

4.1 Alur dan Sistem Kerja

Ketika akan mendapat proyek, tim *account* akan membuat harga penawaran yang di-*approve* oleh klien dengan nominal yang telah disepakati. Tim *account* membuat suatu penawaran tersebut dengan estimasi dari harga yang diberikan tim produksi, harga tersebut telah dilakukan

perbandingan dengan beberapa vendor, dengan permintaan sesuai kebutuhan klien. Harga yang diajukan ke klien sudah terdapat keuntungan perusahaan dalam jumlah presentase yang variatif. Seiring dengan berjalannya proyek, estimasi keuntungan perusahaan tersebut akan menjadi fluktuatif. Estimasi biaya yang sudah dirancang, sesuai dengan penerapan atau akan terjadi penambahan biaya di tengah perjalanan proyek. Ketika sistem pemantauan yang kurang baik, maka semakin lama akan semakin jauh dari yang direncanakan.

Tugas dari tim *account* adalah menjaga agar keuntungan perusahaan tetap dalam perencanaan awal. Maka, setiap pengeluaran pada tim produksi atau tim *account* langsung dicatat ke dalam sistem, tanpa harus menunggu tim keuangan yang merekap pada akhir proyek. Data yang telah dimasukkan akan ditotal dan secara otomatis akan tersinkronisasi dari data pengeluaran terbaru.



Gambar 2. Alur dan Sistem Kerja

4.2 User Requirement

Berdasarkan hasil wawancara dan observasi dapat disimpulkan beberapa kebutuhan *user* terhadap aplikasi sistem monitoring *budget* adalah sebagai berikut:

4.2.1 Kategori User Aplikasi

Pada tabel ini dijelaskan tentang kategori *user* dari aplikasi. Kategori *user* seperti *account*, *account* manajer, keuangan, dan produksi yang setiap kategori usernya memiliki hak akses berbeda-beda.

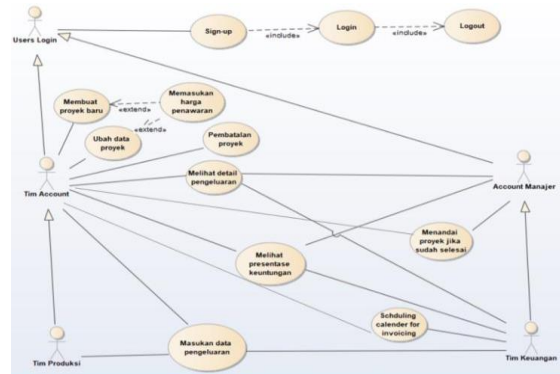
Tabel 1. Kategori User Aplikasi

Kode	Deskripsi
R.001	Membuat proyek baru
R.002	Ubah data proyek
R.003	Pembatalan proyek
R.004	Melihat presentase keuntungan
R.005	Melihat detail pengeluaran
R.006	Menandai proyek jika sudah selesai
R.007	Melakukan penjadwalan untuk invoicing

R.008	Memasukan data pengeluaran
-------	----------------------------

4.2.2 User Case Diagram

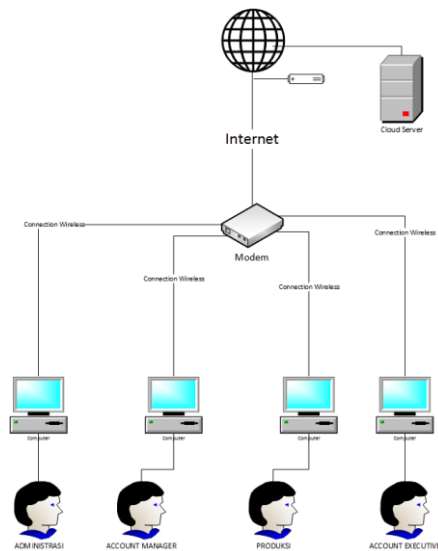
Pada gambar ini dijelaskan tentang *use case* diagram aplikasi *monitoring budget event organizer* berdasarkan kebutuhan dari hasil wawancara dan observasi kepada pihak terkait.



Gambar 3. Use Case Diagram

4.2.3 Deployment Diagram

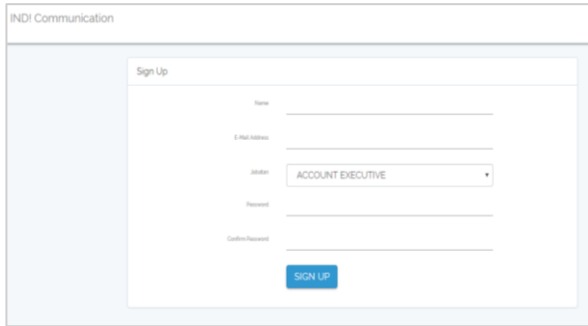
Pembangunan diagram ini menggambarkan arsitektur fisik dari perangkat keras dan perangkat lunak aplikasi. Aplikasi ini berbasis website, maka untuk mengaksesnya dengan memasukkan domain URL pada *web browser*.



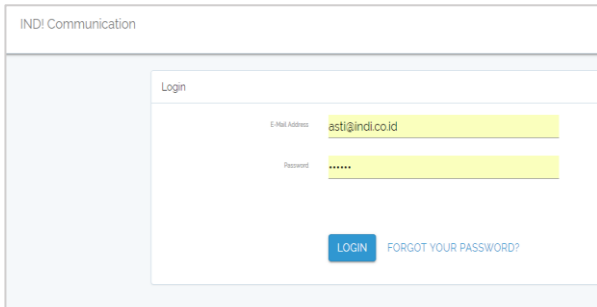
Gambar 4. Deployment Diagram

5. IMPLEMENTASI

Bagian ini akan menjelaskan mengenai tampilan dari aplikasi dan keterangan dari setiap menu. Aplikasi ini diuji melalui *Blackox Testing*, *User Acceptence Test* (UAT) dan wawancara dengan para pengguna.



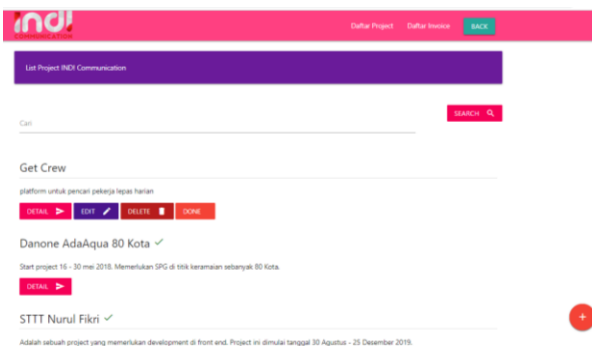
Gambar 5. Tampilan Sign Up



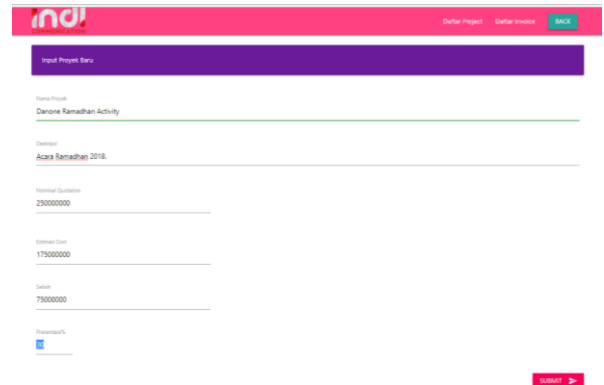
Gambar 6. Tampilan Login



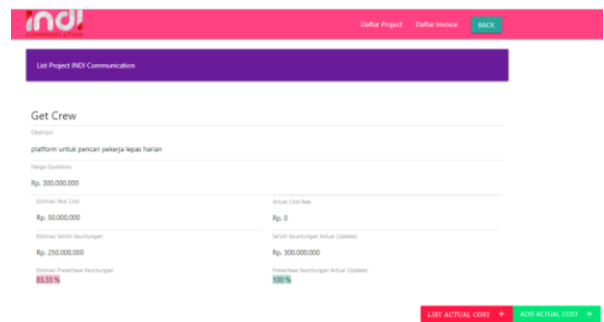
Gambar 7. Tampilan Home



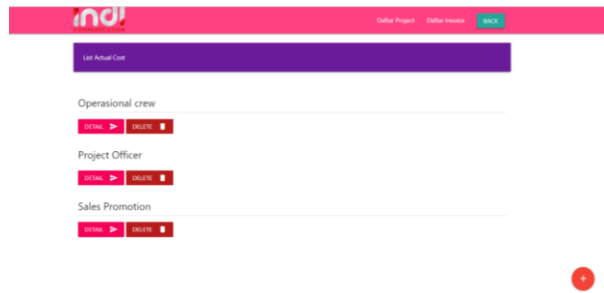
Gambar 8. Tampilan Daftar Proyek



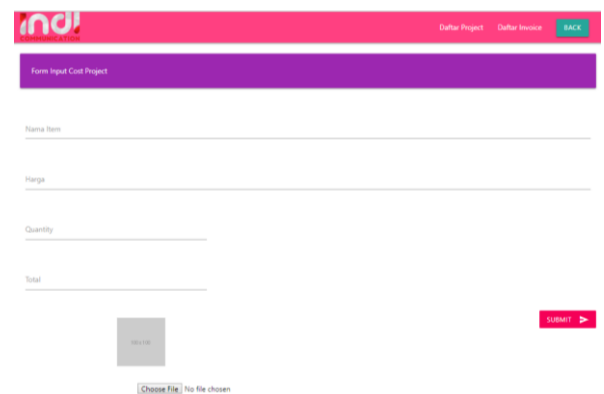
Gambar 9. Tampilan Form Masukan Proyek Baru



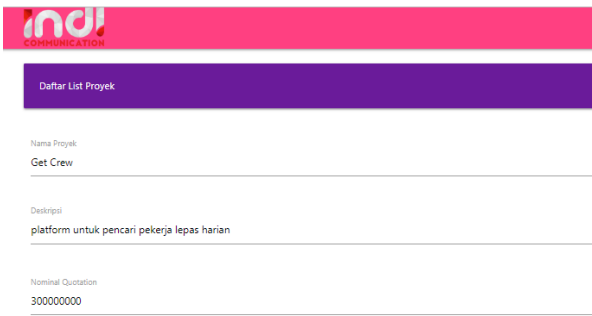
Gambar 10. Tampilan Detail Proyek



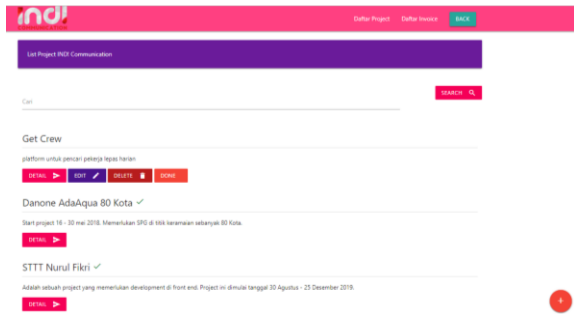
Gambar 11. Tampilan Daftar Pengeluaran Proyek



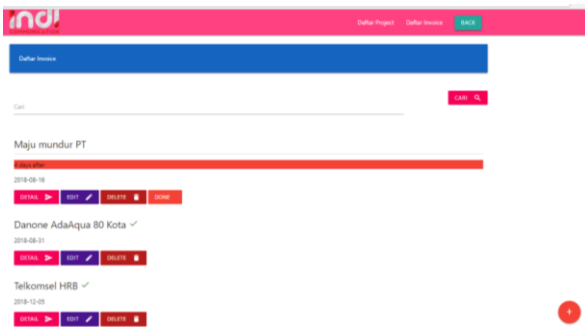
Gambar 12. Tampilan Form Pengeluaran Proyek



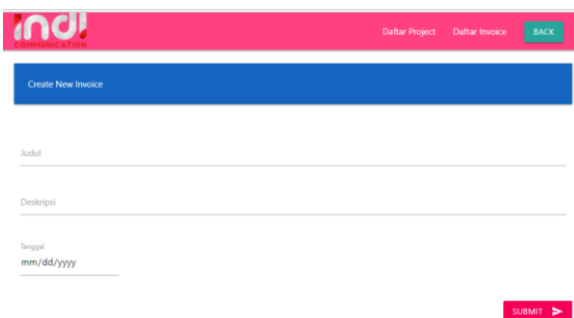
Gambar 13. Tampilan Form Edit Proyek



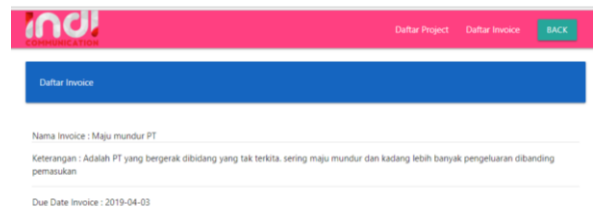
Gambar 14. Tampilan Menandai Proyek Sudah Selesai



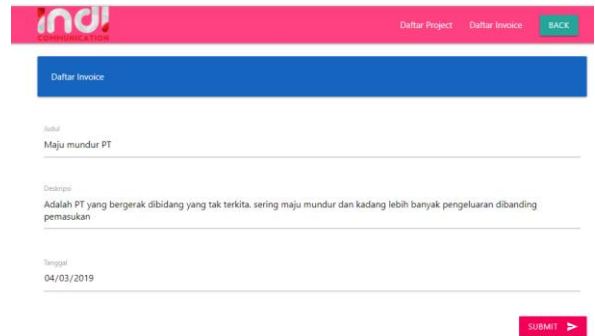
Gambar 15. Tampilan Daftar Invoice



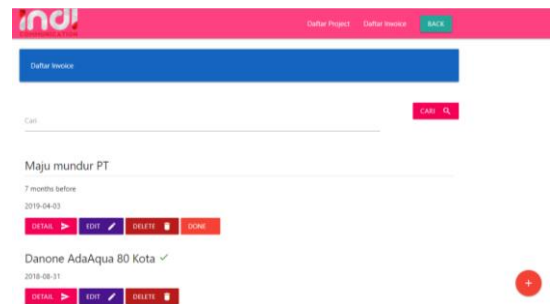
Gambar 16. Tampilan Form Membuat Invoice Baru



Gambar 17. Tampilan Detail Invoice

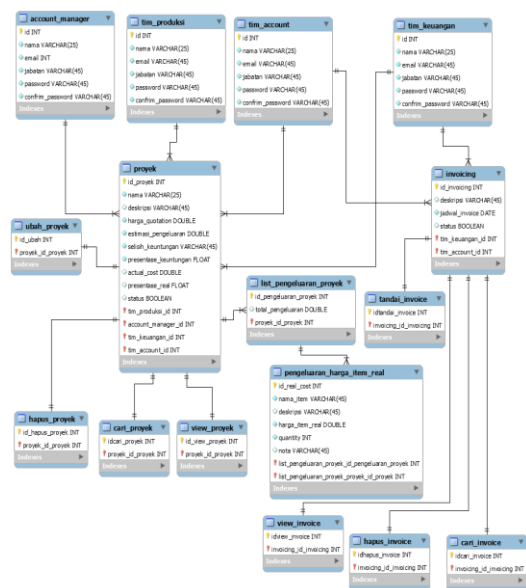


Gambar 18. Tampilan Form Edit Invoice



Gambar 19. Tampilan Menandai Invoice Sudah Selesai

5.1 Implementasi ERD



Gambar 20. ERD

5.2 Hasil *Blackbox Testing*

Setelah aplikasi selesai dibangun. Maka langkah selanjutnya adalah melakukan uji coba terhadap aplikasi tersebut. Dari total 27 kasus uji yang dilakukan. Dinyatakan 100% aplikasi telah berjalan atau dapat digunakan dengan baik.

5.3 Hasil *User Acceptance Test*

Berikut ini adalah hasil UAT yang telah dilakukan oleh user. Setiap user memiliki hak akses masing-masing. Dengan mekanisme setiap user mencoba aplikasi dengan akun mereka. Lalu disimpulkan dengan kasus uji di lapangan oleh *user* langsung, semua kasus uji aplikasi diterima tanpa penolakan. Hanya ada beberapa catatan untuk bahan pengembangan selanjutnya.

Tabel 2. Hasil UAT

Total User	Total Kasus Uji	Total Diterima	Total Diterima Dengan Catatan	Total Ditolak
4	80	67	13	0

5.4 Evaluasi Pengujian Aplikasi

Setelah melakukan wawancara dengan pengguna di lapangan mengenai aplikasi yang dibangun dalam penelitian ini, sebelum dan sesudah penerapan aplikasi. Maka, hasil wawancara tersebut dapat ditarik kesimpulan berupa akurasi perhitungan margin keuntungan proyek, dan efisiensi operasional. Terutama dalam pencarian data sebelumnya, dan nota pembayaran di divisi keuangan. Di bawah ini hasil wawancara dari sebelum, dan setelah penerapan aplikasi.

Tabel 3. Perbandingan Setelah Menggunakan Aplikasi

No	Nama Kasus	Sebelum Menggunakan Aplikasi	Sesudah Menggunakan
1	Sulit mencari nota pembayaran proyek	Membutuhkan waktu 30 menit – 60 menit	Kurang dari 10 menit
2	Lambat dalam perhitungan rekapitulasi margin keuntungan	Membutuhkan waktu minimal 45 menit	Kurang dari 5 menit
3	Kesulitan memperbarui presentase keuntungan ketika proyek sedang berlangsung	Tidak ada	Kurang dari 5 menit

4	Lambat dalam menemukan data proyek yang sudah lampau	Data tertimbun dalam berkas penyimpanan	Data tersimpan rapih dalam aplikasi
---	--	---	-------------------------------------

Setelah pengujian langsung kepada *user*. Maka, dapat disimpulkan dengan adanya aplikasi ini dapat memberi kemudahan dalam operasional perusahaan, penyajian yang lebih cepat dan akurat dibandingkan dengan sebelum menggunakan aplikasi.

6. KESIMPULAN DAN SARAN

6.1 Kesimpulan

Dari hasil penelitian yang dilakukan didapat kesimpulan sebagai berikut:

1. Aplikasi *monitoring budget event organizer* berbasis *web* dapat berfungsi dengan baik. Hal ini dinyatakan dari hasil pengujian fungsional *blackbox* telah berjalan 100%, dan hasil pengujian UAT dinyatakan 83% fitur aplikasi berjalan, dan 17% fitur diterima dengan catatan.
2. Aplikasi *monitoring budget event organizer* berbasis *web* dianggap memberi kemudahan untuk memantau keuangan di PT INDI Notokreasi, dibuktikan dengan akses informasi menjadi lebih cepat dari sebelumnya.

6.2 Saran

Pada dasarnya penelitian ini sudah cukup memenuhi kebutuhan pokok untuk menunjang dalam memonitoring budget di *event organizer*, tetapi akan lebih baik apabila UI dan UX nya juga diperhatikan untuk kemudahan *user*. Selain itu, berikut adalah beberapa saran untuk penyempurnaan aplikasi, diantaranya:

1. Adanya fitur *approval input* pengeluaran proyek yang dimiliki oleh *user manager*, untuk memantau apakah data yang diinput sudah sesuai dengan perencanaan.
2. Tidak hanya terdapat fitur *search*, tetapi ditambahkan dengan fitur *sort*. Untuk mensortir berdasarkan kategori yang diinginkan *user*.
3. Adanya *highlight* pembeda warna ketika keuntungan margin diatas, dan dibawah estimasi perencanaan.
4. Pengelompokan daftar proyek disimpan dalam kategori pertahun dan perbulan. Untuk kemudahan *tracking record* data.

DAFTAR PUSTAKA

- [1] Pressman, “Rekayasa Perangkat Lunak: Pendekatan Praktisi,” Buku Dua, 2002.
- [2] A. P. Basuki, “Membangun Web berbasis PHP dengan *Framework Codeigniter*,” 2007.
- [3] A. Rohman, “Mengenal *Framework Laravel*,” 2014.
- [4] Marlinda L, Hamid A, Ibnu A, “Rancangan Bangun Sistem Informasi Perpustakaan Berbasis Web menggunakan MVC (*Model View Controller*)”, 2016.
- [5] Soetam Rizky, “*Black Box Testing*,” 2011.
- [6] Irvan Nurseha, “Perancangan Sistem Informasi *Quality Control Stock Expired* di PT. Catur Sentosa Anugerah, 2014.

Jurnal Informatika Terpadu

Vol. 5 No. 1 Tahun 2019

Daftar Isi

- Analisis dan Implementasi Sistem Informasi Pengelolaan SDM berbasis OrangeHRM di SDIT AL QUDWAH** 01
Afifah Sausan, Rusmanto
- Implementasi *Network Monitoring System* terintegrasi dengan *Ticketing System* menggunakan Nagios dan osTicket** 06
Debi Onas Saputra, Henry Saptono
- Pengujian dan Implementasi Efektivitas Sistem Virtualisasi berbasis *Docker Platform*** 18
Husni Mubarak, Henry Saptono
- Rancang Bangun *Website Customer Relationship Management (CRM)* pada Modul Akuntansi Studi Kasus CV Esindo Multi Tata** 24
Muhammad Rizky Arinugraha Putra, Sirojul Munir
- Rancang Bangun Sistem *Monitoring Budget* pada *Event Organizer* berbasis *Website* menggunakan *MVC Framework*** 30
Muhammad Tarmizi, Sirojul Munir

Published by:

LPPM STT Terpadu Nurul Fikri

Jln. Raya Lenteng Agung, no. 20, Srengseng Sawah,
Jagakarsa, Jakarta Selatan, DKI Jakarta 12640

Telp. 021 - 786 3191

Email : lppm@nurulfikri.ac.id

Website : <https://journal.nurulfikri.ac.id/index.php/jit>

