

Aplikasi *Dashboard Administrator Server Nginx* Pada aCommerce

Firgi Surya Prasetya¹, Eko Subyantoro², Halim Fathoni³

¹ mahasiswa jurusan ekonomi dan bisnis, ² pembimbing 1, ³ pembimbing 2

Jurnal

Abstrak

Admin perusahaan aCommerce bertanggung jawab dalam mengatur pembaruan sistem operasi, *patch* dan perubahan konfigurasi. Dalam proses pengelolaan *web server* admin biasanya melakukan *start* dan *stop web server* untuk menjalankan *project* yang selesai di *upload* ke *server*, umumnya setiap *project* memiliki 1 *server* sebagai *recovery*. Admin mengakses *server* untuk mengaktifkan *web server* dengan melakukan *remote SSH (Secure Shell)* dan *login user* melalui *CLI (Command Line Interface)*. Dalam mengakses *server* admin melakukan pengecekan status *web server*. Hal ini menjadi kendala karena banyak memakan waktu dalam mengakses dan pengecekan status setiap *server*.

Tujuan Tugas Akhir ini menghasilkan aplikasi yang memudahkan admin untuk mengaktifkan atau menonaktifkan *web server* dan melakukan pengecekan status setiap *server* yang dibangun menggunakan metode *prototype*. Aplikasi ini dibangun menggunakan bahasa pemrograman *php* dan *shell*, *ubuntu* sebagai sistem operasi *server*, dan menggunakan *MySQL* sebagai *database*.

Kata Kunci: *Web Server Nginx, Aplikasi Dashboard, Prototype.*

PENDAHULUAN

Pada perusahaan aCommerce yang bergerak dibidang bisnis, maka jelas akan berkaitan dengan teknologi jaringan *www*, teknologi *database*, dan *server* sebagai pengatur lalu lintas data (aCommerce, 2013). Dalam mengatur dan menggunakan teknologi, aCommerce memiliki seorang admin untuk mengontrol dan manajemen *server*. Admin bertanggung jawab dalam mengatur pembaruan sistem operasi, *patch* dan perubahan konfigurasi (Zakaria, 2017). Dalam proses pengelolaan *web server* admin biasanya melakukan *start* dan *stop web server* untuk menjalankan *project* yang selesai di *upload* ke *server*, umumnya setiap *project* memiliki 1 *server* sebagai *recovery*. Admin mengakses *server* untuk mengaktifkan *web server* dengan melakukan *remote SSH (Secure Shell)* dan *login user* melalui *CLI (Command Line Interface)*.

Dalam mengakses *server* admin melakukan pengecekan status *web server*. Hal ini menjadi kendala karena banyak memakan waktu dalam mengakses dan pengecekan status setiap *server*.

Berdasarkan uraian di atas dalam proses mengaktifkan atau menonaktifkan dan pengecekan status *service web server* membutuhkan waktu yang cukup lama. Sehingga dibutuhkan teknologi untuk meningkatkan efisiensi waktu pada proses tersebut. Maka perlu digunakan teknologi *web* sebagai solusi dalam melakukan akses ke *server* dan pengecekan status *web server*.

Metodologi Penelitian

Alat yang digunakan dalam penelitian terbagi menjadi dua yaitu perangkat keras (*hardware*) dan perangkat lunak (*software*). Bahan yang digunakan berupa hasil wawancara tidak terstruktur dan hasil

observasi. Metode pengembangan sistem yang digunakan yaitu metode *prototype* (Mulyani, 2016). Metode *prototype* terdiri dari beberapa tahapan yaitu :

1. Menganalisis kebutuhan

Dimulai dengan mengumpulkan kebutuhan *user* terhadap perangkat lunak yang akan dibuat dengan program *prototype* sehingga *user* akan terbayang dengan apa yang diinginkan. Tahap analisis ini meliputi kebutuhan *user interface* (menu, dialog, *input* dan *output*) dan fungsi-fungsi pemrosesan sederhana.

2. Melakukan perancangan cepat dan membangun sebuah *prototype*.

Tahapan ini dilakukan dengan merancang kebutuhan sistem sesuai kebutuhan *user* serta pembuatan sebuah simulasi aplikasi *prototype*. Tahapan pertama dilakukan perancangan menggunakan *Flowchart* dan *DFD (Data Flow Diagram)*, selanjutnya perancangan database dilakukan menggunakan *software MySQL*. Tahap terakhir yaitu dengan membuat sebuah *prototype*. Aplikasi *prototype* dibuat menggunakan PHP.

3. Uji dan evaluasi *prototype*.

Tahap uji dan evaluasi *prototype* dilakukan oleh konsumen atas *prototype*. Tahap ini dilakukan untuk memastikan *prototype* dapat dengan mudah dijalankan untuk tujuan demonstrasi.

4. Transformasi *prototype*.

Tahap ini dilakukan dengan mentransformasikan *prototype* menjadi perangkat lunak yang beroperasi penuh dengan melakukan penghilangan kode-kode yang tidak dibutuhkan, penambahan program-program

yang memang dibutuhkan dan perbaikan dan pengujian perangkat lunak secara berulang.

Hasil dan Pembahasan

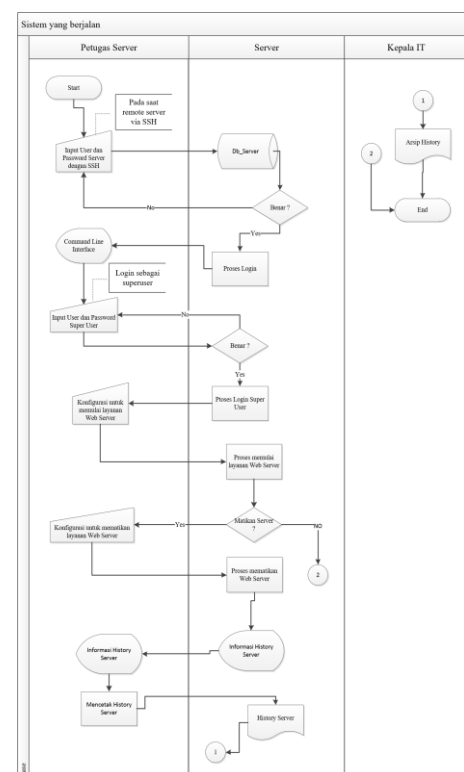
Analisis kebutuhan merupakan analisis sistem yang menghasilkan perancangan mapping chart sistem yang berjalan dan diusulkan..

1. Analisis

a. Sistem yang berjalan

Tahap analisis sistem yang berjalan ini memiliki tujuan untuk mengetahui sistem yang sedang berjalan, dan menemukan solusi pada objek yang diteliti dalam sistem. Sistem aplikasi yang berjalan

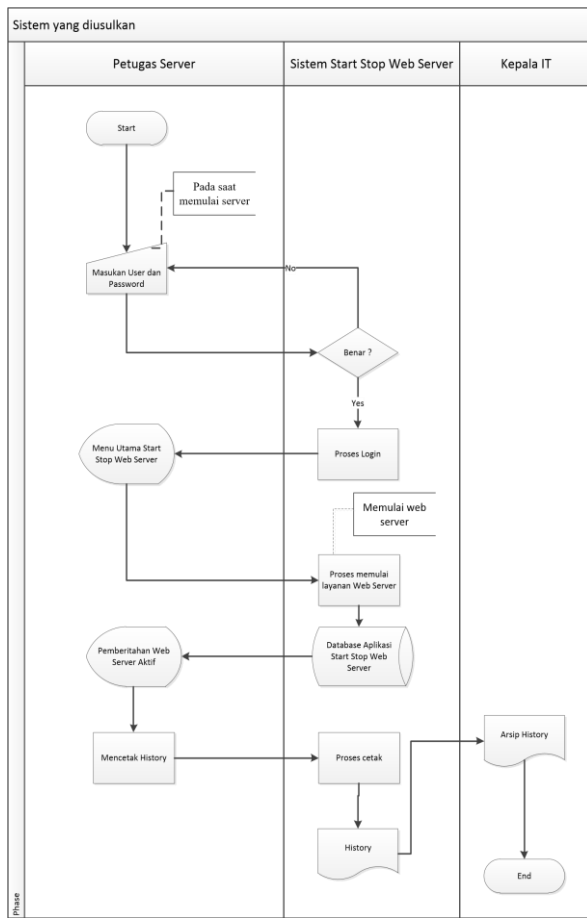
Rencana Kebutuhan merupakan analisis sistem yang menghasilkan Mapping Chart sistem yang berjalan. Sistem yang berjalan disajikan pada Gambar 1.



Gambar 1. Mapping Chart Sistem Yang Sedang Berjalan

b. Sistem yang diusulkan

Tahap analisis sistem yang diusulkan merupakan tahapan dalam pembuatan sistem baru yang akan dibangun sesuai dengan kebutuhan berdasarkan analisis sistem yang telah berjalan. Analisis system yang diusulkan digambarkan dengan menggunakan *mapping chart*. *Mapping chart* sistem ini disajikan pada Gambar 2.

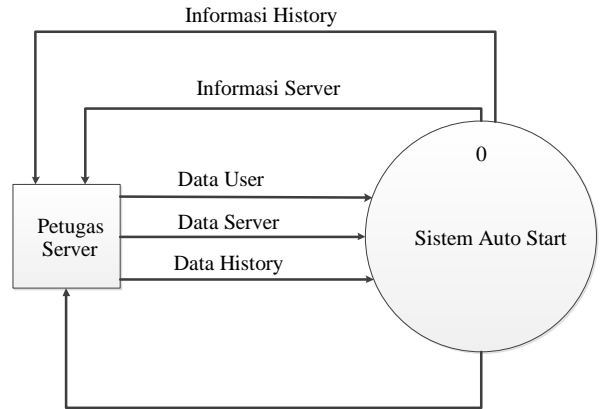


Gambar 2. Mapping Chart Sistem Yang Diusulkan

2. Perancangan/ Desain

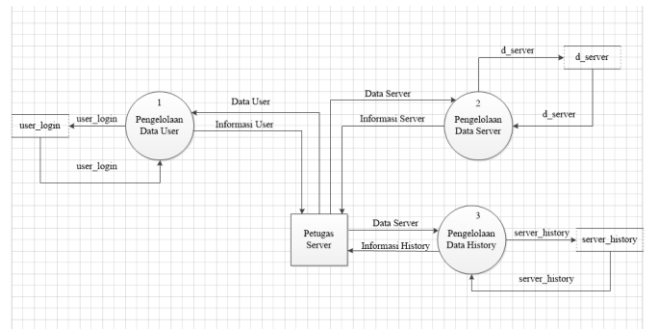
a. Perancangan DFD

DFD yang digambarkan pada perancangan sistem yaitu DFD level 0 dan DFD level 1. DFD level 0 disajikan pada Gambar 3.



Gambar 3. DFD level 0

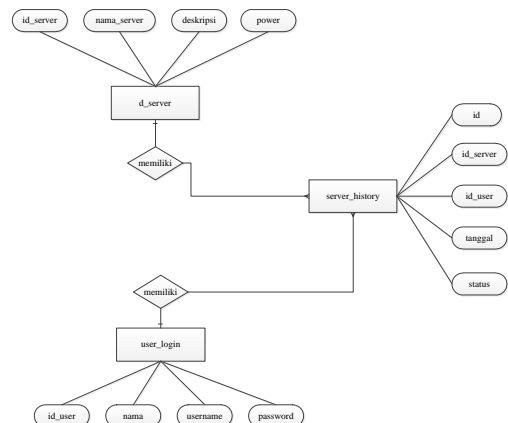
DFD level 1 merupakan pemecahan dari DFD konteks yang memuat penyimpanan dan menampilkan data. DFD level 1 disajikan pada Gambar 4.



Gambar 4. DFD level 1

b. Perancangan ERD

ERD berfungsi merancang relasi antar entitas atau tabel beserta atributnya dalam membuat database. Perancangan ERD dibuat menggunakan Microsoft Office Visio disajikan pada Gambar 5.



Gambar 5. Rancangan ERD

3. Implementasi

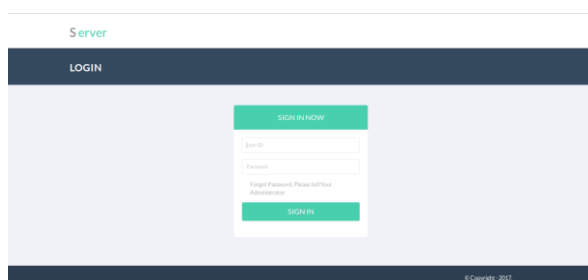
Implementasi merupakan tahap penerapan dari hasil analisis dan perancangan untuk diterjemahkan ke dalam bahasa komputer. Implementasi dilakukan dengan pengkodean yang menghubungkan data-data perancangan yang telah dibuat sebelumnya.

a. Tampilan program

Pada tahapan ini akan ditampilkan beberapa tampilan program yang dibuat berdasarkan rancangan-rancangan yang telah diusulkan pada tahap sebelumnya :

1. Tampilan *login*

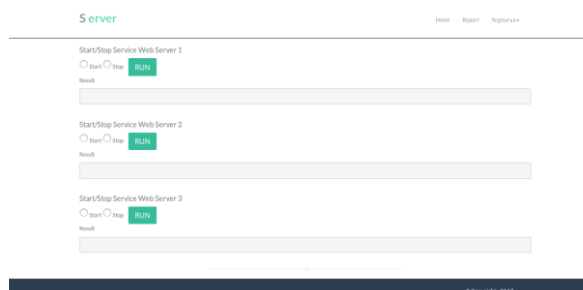
Halaman login dapat diakses oleh petugas server. Dengan memasukkan user dan password yang sah. Tampilan Halaman Login disajikan pada Gambar 6.



Gambar 6. Tampilan *Login*

2. Tampilan Halaman Petugas *Server*

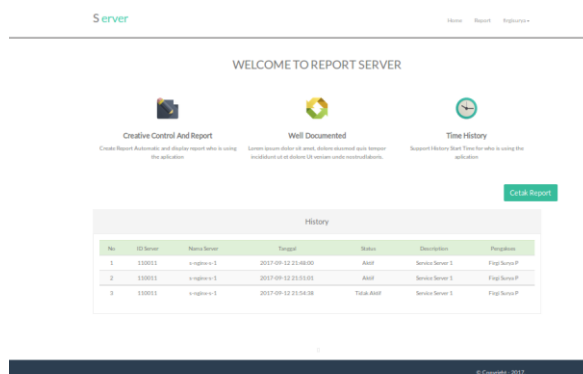
Halaman petugas server menampilkan menu *report*, *logout*, dan *start/stop Web Server*. Jika petugas server memilih menu *report* maka akan diarahkan ke halaman *report*. Jika memilih menu *logout* maka admin akan diarahkan ke halaman login. Dan petugas *server* dapat menjalankan menu mengaktifkan atau menonaktifkan *web server*. Tampilan Halaman petugas *server* disajikan pada Gambar 7.



Gambar 7. Tampilan Petugas *Server*

3. Tampilan Halaman *Report*

Halaman *report* menampilkan *log web server* yang aktif dan tidak aktif. Petugas *server* dapat memilih aksi cetak untuk mencetak *log web server*. Tampilan Halaman *Report* disajikan pada Gambar 8.



Gambar 8. Tampilan *Report*

b. Pengujian

Pengujian sistem secara lengkap dilakukan untuk menjamin bahwa syarat dan spesifikasi sistem telah terpenuhi berdasarkan persyaratan-persyaratan yang ada pada tahap sebelumnya. Metode pengujian yang adalah *black box testing*.

Hasil pengujian aplikasi ini dengan menggunakan metode *black box testing*, didapatkan hasil pengujian yaitu :

1. *Web server* aktif dengan melihat *box result*.

2. Pengecekan dilakukan dengan memanggil *localhost* pada *web browser*.
3. Melakukan cetak pada halaman *report*.

Zakaria, M. (2017). *Tugas dan Tanggung Jawab Seorang Admin Server*. Retrieved June 5, 2017, from <http://www.nesabamedia.com>.

Kesimpulan dan Saran

Kesimpulan

Kesimpulan dapat diambil yaitu telah dihasilkannya “Aplikasi *Dashboard Administrator Server Nginx Pada aCommerce*” yang mempermudah admin untuk melakukan *start*, *stop* dan pengecekan status beberapa *web server* secara bersamaan.

Saran

Berdasarkan laporan Tugas Akhir tentang aplikasi ini, maka disarankan untuk melakukan pengembangan perlu ada penambahan *database* untuk *web server* dan konfigurasi otomatis *start stop* yang ada pada virtualisasi sehingga memudahkan penambahan *web server*.

REFERENSI

- aCommerce. (2013, May). <http://acommerce.co.id/who-we-are/>. Retrieved October 02, 2017, from acommerce.co.id: <http://acommerce.co.id/who-we-are/>
- Mulyani, S. (2016). *Analisis dan Perancangan Sistem Informasi Manajemen Keuangan Daerah*. Bandung: Abdi Sistematika.
- Mulyani, S. (2016). *SISTEM INFORMASI MANAJEMEN RUMAH SAKIT ANALISIS DAN PERANCANGAN*. Bandung, Jawa Barat, Indonesia: Abdi Sistematika.