JOISIE Journal Of Information System And Informatics Engineering
Vol. 1, No.1 Juni 2017, Hlm 1-68

p- ISSN: 2503-5304
e- ISSN: 2527-3116

RANCANG BANGUN APLIKASI CHATTING BERBASIS WEB MENGGUNAKAN DOCKER

Firlya Adinta[1)], **Indri Neforawati**[2)]
Program Studi Teknik Komputer dan Jaringan, Politeknik Negeri Jakarta[1,2)]
Jl. Prof. Dr. G.A Siwabessy, Kampus Baru UI Depok 16424[1,2)]
E-mail : 1[1)], indri.neforawati@tik.pnj.ac.id[2)]

Abstrak

Perkembangan teknologi informasi diiringi oleh perkembangan perangkat – perangkat pendukungnya memudahkan pengguna teknologi untuk membantu pekerjaannya sehari-hari dan dapat saling berkomunikasi satu sama lain. Aplikasi chatting berbasis web merupakan salah satu alternatif yang dapat digunakan jika pengguna tidak ingin melakukan instalasi terhadap aplikasi chatting di ponselnya atau didesktop. Dalam pembuatan aplikasi chatting ini menggunakan teknologi virtual mesin yang digunakan untuk menjalankan suatu aplikasi. Teknik kontainerisasi (virtualisasi berbasis container) hadir sebagai solusi dari peningkatan jumlah aplikasi web yang harus dihosting harus diikuti dengan peningkatan kualitas ataupun kuantitas sumber daya. Docker adalah salah satu software yang mengadopsi teknik kontainerisasi dan semakin banyak diterapkan di dalam lingkungan pengembangan aplikasi. Penelitian ini mencoba untuk melakukan rancang bangun aplikasai chatting berbasis web menggunakan teknologi docker dalam pembuatan apliakasi chatting.

Kata Kunci *: Aplikasi Chat, Mesin Virtual, Docker, kontainerisasi, web-based.*

Abstract

The development of information technology is accompanied by development of support tools that allow users of technology to help the daily work and can communicate with each other. Web-based chat application is one alternative that can be used if the user does not want to perform the installation of chat applications on their phone or desktop. In making the chat application using virtual machine technology that is used to run an application. Mechanical containerize (container-based virtualization) is present as a solution of an increase in the number of hosted web application that must be followed by an increase in the quality and quantity of resources. Docker is a software which adopted the technique containerize and increasingly applied in the application development environment. This study tried to design using a web-based chat aplikasai Docker technology in the manufacture of a chat application.

Keywords *: Aplikasi Chat, Mesin Virtual, Docker, kontainerisasi, web-based.*

PENDAHULUAN

Semakin berkembangnya teknologi mempermudah manusia dalam melakukan kegiatan sehari – harinya. Teknologi juga dapat menghemat waktu manusia dalam melakukan tugas – tugasnya. Dengan teknologi manusia dapat berkomunikasi jarak jauh dan tidak terbatas oleh ruang dan juga waktu, kita bahkan dapat bertukar *e-mail* maupun pesan dengan jarank jauh. Sekarang ini sudah banyak tersedia aplikasi – aplikasi pengiriman pesan melalu internet atau yang sering disebut aplikasi *chatting*, *Chatting* merupakan suatu metode komunikasi yang bersifat *real-time*. Untuk dapat menggunakan aplikasi – aplikasi tersebut kita terlebih dahulu melakukan

penginstalan, tetapi jika tidak ingin maka ada sebuah aplikasi *chatting*, yang berbasis *web*.

Terdapat banyak aplikasi – aplikasi berbasis *web* tetapi tidak sepopuler aplikasi – aplikasi pada *mobile phone*. Bicara mengenai aplikasi berbasis web tidak terlepas dari *server* dan hosting yang digunakan. Belakangan teknologi informasi mengalami berbagai perubahan dan telah bergerak dari *grid* ke *cloud*, dari arsitektur yang berorientasi layanan ke *microservices*, dan dari virtualisasi mesin ke teknologi *container*. Karena kemajuan internet dan *cloud computing* saat ini sangat cepat, pelaku bisnis yang memiliki system informasi jaringan memiliki banyak pilihan untuk menanamkan *system* miliknya; hosting dengan shared directory, hosting dalam sebuah VPS, menyewa Data Center di cloud, atau membangun Data Center local sendiri. Tentu saja hal itu akan di sesuaikan dengan kebutuhan dan budget perusahaan tersebut.

Sistem informasi dalam sebuah perusahaan bersifat dinamis sesuai perkembangan bisnisnya, sehingga dibutuhkan sebuah system dengan skalabilitas yang tinggi. Jika sebuah system dikembangkan untuk memenuhi kemajuan sebuah perusahaan maka aplikasi akan melakukan migrasi jika terjadi perubahan pada infrastrukturnya. Perkembangan yang bersifat dinamis ini menjadi masalah bagi pengembang dan administrator sebuah system aplikasi, terutama aplikasi berbasis web. Pengembang harus mempunyai aplikasi yang mampu berjalan dalam lingkungan yang berbeda untuk menjawab permasalahan seperti saat melakukan depelopment dan testing pada environment yang berbeda atau migrasi *server*. Kendala yang sering terjadi adalah *Conflicting Dependencies, Missing Dependencies* dan perbedaan platform server itu sendiri.

Container adalah sebuah teknologi baru yang disebut dengan docker. Docker ini adalah *virtual* mesin yang mencakup penempatan aplikasi dalam sebuah *container* di lingkungan operasinya sendiri. Berbeda halnya dengan *virtual* mesin lain yang diterapkan di *server* pada umumnya. docker telah mentransformasi teknologi informasi dengan cepat dan menarik perhatian luar biasa karena keuntungan beragam yang diberikannya. Oleh karena itu munculah ide untuk Implementasi Teknologi Docker pada *cloud computing*. Rancang Bangun Aplikasi *Chatting* Berbasis *Web* Menggunakan Teknologi Docker. Dengan harapan dapat meningkatkan kinerja dari aplikasi – aplikasi tersebut dan memanfaatkan teknologi baru yang mudah dikelola dan didapatkan atau dapat kita sebut *open source*.

BAHAN DAN METODE

Penelitian ini dilakukan dengan metode sebagai berikut:

A. Studi Literatur
Mencari informasi terkait secara riset kepustakaan dengan penelitian dan mempelajari buku-buku yang berhubungan dengan permasalahan yang dihadapi.

B. Analisa Kebutuhan dan Perancangan Sistem
Melakukan analisa terhadap permasalahan yang dihadapi dan melakukan pembuatan kerangka sistem yang akan dijadikan acuan dalam pembangun sistem.

C. Implementasi
Melakukan implementasi setelah berhasil melakukan simulasi sesuai dengan rancangan desain sistem.

D. Pengujian dan Analisis Hasil
Setelah rancangan telah diimplementasikan selanjutnya adalah menguji sistem yang telah dibuat sehingga dapat ditemukan kesalahan-kesalahan sistem yang mungkin terjadi sehingga dapat diperbaiki.

E. Perbaikan Sistem
Melakukan evaluasi dan penyempurnaan sistem.

F. Teori Pendukung

a) Cloud Computing

Cloud Computing adalah sebuah model komputasi / computing, dimana sumber daya seperti processor / *computing power*, *storage*, *network*, dan *software* menjadi abstrak dan diberikan sebagai layann di jaringan / internet menggunakan pola akses remote. Model billing dari layanan in umumnya mirip dengan modem layanan publik. Ketersediaan on-demand sesuai kebutuhkan, mudah untuk di kontrol, dinamik dan skalabilitas yang hampir tanpa limit adalah beberapa atribut penting dari *cloud computing* [3].

b) Docker

Docker adalah sebuah platform terbuka untuk siapapun yang bertujuan menggunakan sebuah platform untuk membangun, mendistribusikan dan menjalankan aplikasi dimanapun seperti laptop, data center,virtual machine ataupun cloud.Docker merupakan open source software di bawah Lisensi Apache Versi 2.0 yang bisa dipergunakan secara gratis. Docker menggunakan arsitektur client-server. Docker client menghubungi Docker daemon, yang melakukan pekerjaan berat, menjalankan, dan mendistribusikan Docker container anda. Kedua Docker client dan daemon dapat berjalan pada sistem yang sama. Docker client dan daemon berkomunikasi via sockets atau lewat API yang disediakan Docker.

Docker memberikan berbagai macam keuntungan dan kemudahan dalam proses deployment software, dimana dalam infrastructure cloud dapat menjalankan banyak pekerjaan sekaligus dengan menggunakan docker dan AWS untuk mempercepat proses deployment, optimalisasi aplikasi dan isolasi. Platform ini dapat digunakan untuk membangun, mempersiapkan, dan menjalankan aplikasi. Dapat membuat aplikasi dari bahasa pemrograman yang berbeda pada lapisan apapun. Aplikasi dapat di bungkus dalam container, dan aplikasi dapat berjalan pada lingkungan apapun dimana saja. Kelebihan docker pada layanan cloud ini dapat membantu permasalahan pada aplikasi yang menggunakan multi container dan layanan pada server cluster shared [2].

c) Web Server

Web server adalah perangkat lunak yang menjadi tulang belakang dari *world wide web* (www) yang pertama kali tercipta sekitar tahun 1980an**.** *Web server* menunggu permintaan dari *client* yang menggunakan *browser* seperti Netscape Navigator, Internet Explorer, Mozilla Firefox, dan program browser lainnya. Jika ada permintaan dari *browser*, maka *web server* akan memproses permintaan itu kemudian memberikan hasil prosesnya berupa data yang diinginkan kembali ke *browser*.

Data ini mempunyai format yang standar, disebut dengan format SGML (S*tandar General Markup Language*). Data yang berupa format ini kemudian akan ditampilkan oleh *browser* sesuai dengan kemampuan *browser* tersebut. Contohnya, bila data yang dikirim berupa gambar, browser yang hanya mampu menampilkan teks (misalnya *lynx*) tidak akan mampu menampilkan gambar tersebut, dan jika ada akan menampilkan alternatifnya saja.

Web server, untuk berkomunikasi dengan *client*-nya (*web browser*) mempunyai protokol sendiri, yaitu HTTP (*hypertext transfer protocol*). Dengan protokol ini, komunikasi antar *web server* dengan client-nya dapat saling dimengerti dan lebih mudah. Seperti telah dijelaskan diatas, format data pada *world wide web* adalah SGML. Tapi para pengguna internet saat ini lebih banyak menggunakan format HTML (*hypertext markup language*) karena penggunaannya lebih sederhana dan mudah dipelajari [1].

d) Chatting

Chatting adalah percakapan dua orang atau lebih secara realtime melalui jaringan internet. Layanan-layanan untuk chatting di internet antara lain Yahoo Messanger, Mirc, Pidgin, Skype, MSN Messenger, Windows Live Messenger, Trilian, Camfrog, ICQ dan sebagainnya. Dengan adanya layanan Chat memungkinkan kita untuk dapat

berkomunikasi melalui internet dengan orang-orang yang berada di seluruh dunia [4].

e) VPS

VPS (Virtual Private Server) yaitu sebuah teknologi *server side* yang memungkinkan sebuah mesin dengan kapasitas besar dibagi ke beberapa mesin virtual. Tiap mesin virtual melayani sistem operasi dan perangkat lunak secara mandiri dan dengan konfigurasi yang cepat. Kemajuan teknologi saat ini memungkinkan untuk mengembangkan aplikasi remote desktop dari sebuah smartphone ke sebuah mesin virtual. VPS atau disebut juga VDS (Virtual Dedicated Server) atau VM (Virtual Machine) adalah teknologi server side tentang sistem operasi dan perangkat lunak yang memungkinkan sebuah mesin dengan kapasitas besar dibagi ke beberapa virtual mesin. Tiap virtual mesin ini melayani sistem operasi dan perangkat lunak secara mandiri dan dengan konfigurasi yang cepat. Beberapa sistem pembuat virtualisasi (hypervisor) dikenal saat ini misalnya: VMWare, Xen, UML (khusus Linux), VZ/OpenVZ [5].

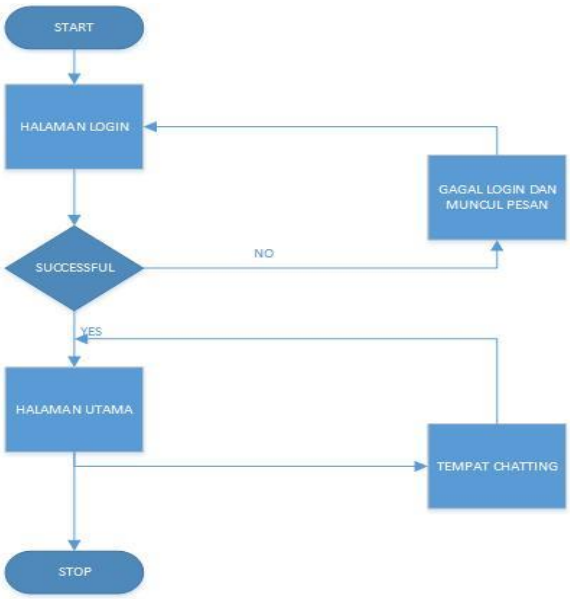
HASIL DAN PEMBAHASAN

A. Deskripsi Program Aplikasi.

Pada project yang kali ini akan menghasilkan *output* atau hasil akhir berupa aplikasi – aplikasi yang menerapkan teknologi docker dalam tahap pengimplementasiannya. Pada aplikasi chatting menggunakan database MongoDB untuk menyimpan data – data *user* yang sudah melakukan registrasi pada aplikasi *chatting* ini yang kemudian akan digunakan untuk *login* sebelum memulai bertukar pesan dengan sesama pengguna aplikasi *chatting* ini. Menggunakan *web* server nginx yang *open source*.

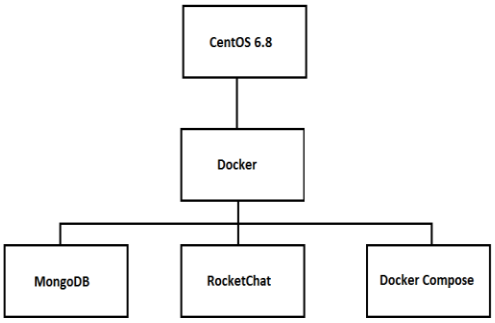
B. Flow Chart

Pada flow chart dapat dijabarkan ketika menggunakan aplikasi *chatting*. Pertama *user* akan melakukan *login* terlebih dahulu jika *login* gagal maka sistem akan tetap berada di halaman *login* sampai *login* berhasil dilakukan. Setelah itu *user* akan dapat mengakses halaman utama dari aplikasi *chatting* tersebut. Jika ingin mengirim pesan maka *user* harus masuk ke halaman *chatting*, jika sudah selesai melakukan *chatting* maka *user* dapat kembali ke halaman utama.



Gambar 1. Flow Chart Aplikasi Chatting berbasis web

C. Storyboard



Gambar 2. Story Board Docker

D. Uji Coba system

Pada pengujian kali ini kita melakukan pemanfaatan teknologi docker pada aplikasi *chatting* berbasis *web*. Penelitian ini dimaksudkan untuk dapat memudahkan

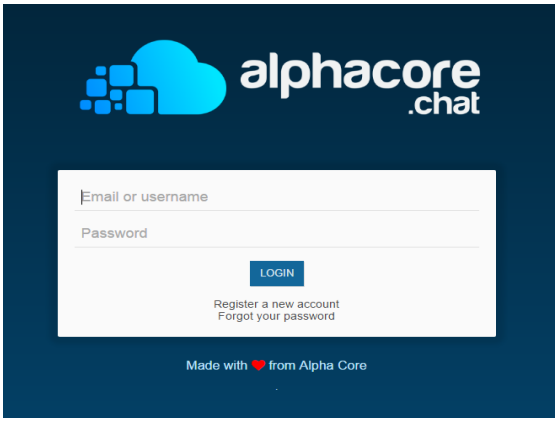
pengembang aplikasi atau pun sysadmin dalam mengembangkan sebuah aplikasi. Sehingga pengembang aplikasi tidak perlu lagi membuat website bertulisankan maintance karena sedang dalam tahap pengembangan atau perbaikan. Jika menggunakan docker pengembang hanya tinggal mengganti sebuah *port* yang ada pada *container* yang ingin di *up* kan.

a) Kecepatan, Dalam pengukuran kecepatan akses pada aplikasi *chatting*, terdapat beberapa factor yang mempengaruhi diantarannya yaitu ukuran -ukuran gambar, pesan, atau *video* yang dikirim. pada tahap ini dilakukan pengujian dengan membandingkan kecepatan akses melalui jaringan komputer lokal, dan *Internet* dengan menggunakan *web browser* dan hasilnya menunjukkan bahwa sistem dapat berjalan dengan baik.

b) Kompatibilitas *browser*, pada tahap ini merupakan pengujian yang dilihat dari sisi *user* hal ini dilakukan untuk memastikan bahwa setiap informasi pada aplikasi chatting dapat diakses oleh *user* dengan baik. Tahap pengujian ini telah menguji kompatibilitas *browser* dengan menggunakan beberapa *web browser* di antaranya , mozilla firefox, Google Chrome dan Opera. Secara garis besar tidak ada perbadaan yang mendasar dari hasil pengujian tersebut dan menunjukkan bahwa sistem dapat berjalan dengan baik.

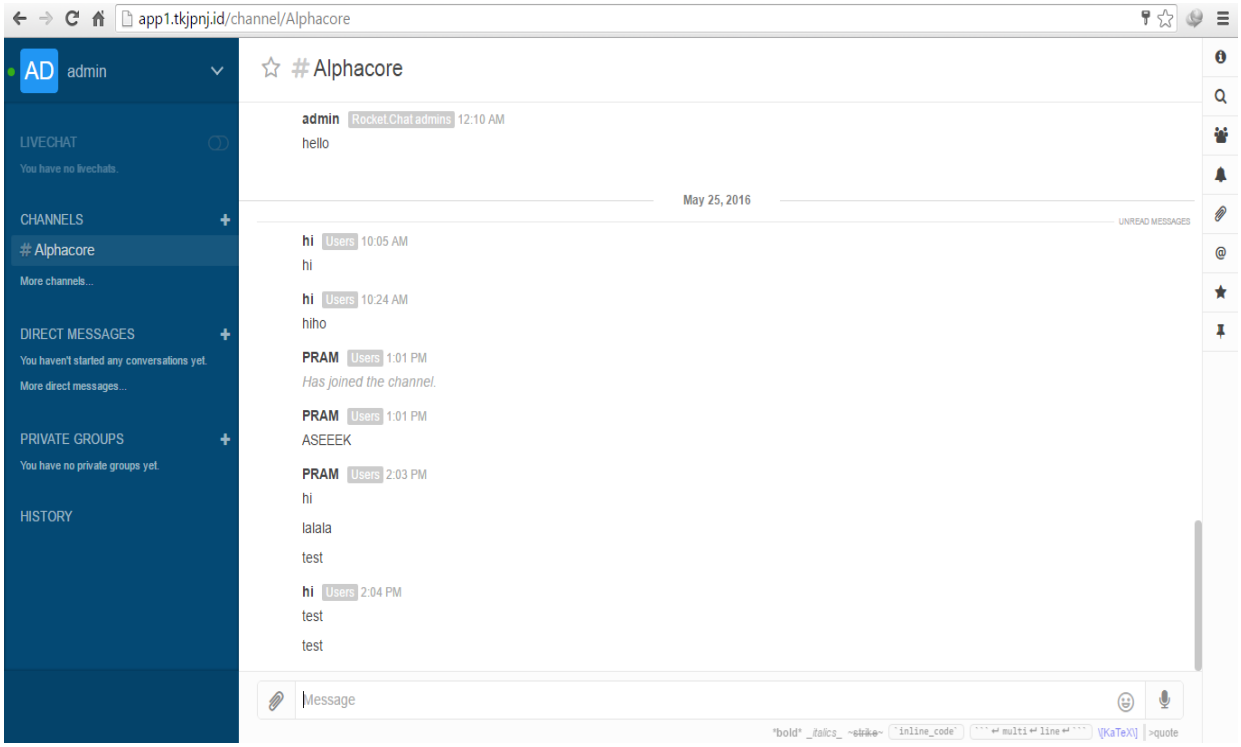
E. Tampilan Aplikasi

a) Form Login, tahap awal untuk dapat menggunakan aplikasi chatting adalah dengan melakukan *login* terlebih dahulu. Halam *login* dapat diakses dengan mengakses nama *domain* yang telah dibuat.



Gambar 3. Halaman Login

b) Halaman Chatting, setelah dilakukan *login* kepada aplikasi *chatting* maka akan langsung diarahkan oleh sistem pada halam *chatting*. Halaman ini digunakan oleh user untuk saling menukar pesan.



Gambar. 4 Halaman Chatting

KESIMPULAN

1. Kesimpulan

Berdasarkan Kegiatan yang telah dilakukan oleh penulis selama perancangan sampai implementasi teknologi docker pada *cloud computing* ini, maka dapat diambil beberapa kesimpulan sebagai berikut :

a. Aplikasi *Chatting* dapat berjalan dengan baik menggunakan teknologi Docker.

b. Berberapa aplikasi diatas hanya dapat dibuka jika *computer user* terhubung dengan internet.

c. Jika *server* yang satu sedang *down* dan ingin memindahkan aplikasi kita ke *server* yang lain dengan docker akan menjadi lebih mudah.

2. Saran

Pemanfaatan teknologi Docker pada aplikasi *chatting* berbasis *web* tidak terlepas dari kekurangan dan kelemahan. Oleh karena itu, penulis memberikan beberapa saran yang dapat digunakan sebagai acuan dalam penelitian atau pengembangan selanjutnya, yaitu :

a. Aplikasi *chatting* dalam penelitian hanya dapat mengirim pesan teks, gambar, pesan suara maupun *video call* berbasis *web*. Ke depannya bisa dikembangkan lagi menjadi suatu apikasi yang lebih baik dengan menggembangkan aplikasi *chatting* ini menjadi berbasis *mobile*.

b. Dalam melakukan konfigurasi dalam melakukan implementasi teknologi docker pada *cloud computing* ini masih menggunakan CLI. Ke depannya bias dikembangkan lagi dengan membuat tampilan antar muka yang *user friendly*.

DAFTAR PUSTAKA

[1] Evy , N., 2012. ANALISIS DAN PERANCANGAN WEB SERVER PADA HANDPHONE. *Sistem Informasi,* 5(2), pp. 1-17

[3] Onno, W. P., 2011. *Cloud Computing Mengunakan Open Source*. 1st ed. Jakarta: One Destination Center .

[4] BIBLIOGRAPHY \l 1033 Priyanto, D. (2009). Belajar Mudah Internet. In D. Priyanto, *Belajar Mudah Internet; Browsing, Download-Upload, E-Mail, Chatting, mailing list, Friendster, Blog dan Facebook* (p. 28). Jakarta: MediaKom.

[5] Rohmat, H., Edhy , S. & Suwanto , R., 2014. PENGONTROLAN VPS (VIRTUAL PRIVATE SERVER) SEBAGAI SERVER RADIO STREAMING VIA ANDROID. *JARKOM ,* 2(1), pp. 99-107.

[2] BIBLIOGRAPHY M. Agung Nugroho, M., & Katardi, R. (n.d.). ANALISIS KINERJA PENERAPAN CONTAINER UNTUK LOAD BALANCING WEB SERVER PADA RASPBERRY PI. *JIPI*, 7-15.