

Analisis Performansi Container Docker, LXC dan LXD Pada Web Server, FTP Server dan Mail Server

Henni Endah Wahanani^{1*}, Mohammad Idhom², M. Faris Eka Saputra³

¹²³Program Studi Informatika, Universitas Pembangunan Nasional “Veteran” Jawa Timur

*Corresponding author email: henniendah.if@upnjatim.ac.id

Abstrak— Teknologi tiap tahun berkembang pesat dengan memberi dampak serta perubahan besar bagi manusia dalam mempermudah pekerjaan. Dalam informatika Salah satu teknologi yang terus berkembang adalah teknologi virtualisasi, untuk menggunakan teknologi virtualisasi membutuhkan sumber daya seperti *server* dan membagi menjadi sumber daya virtual yang disebut VM (*Virtual Machine*). Namun untuk mengaksesnya VM terkenal mempunyai *overhead* yang besar. Dengan adanya dampak tersebut, dibandingkan menggunakan VM, maka dalam penelitian ini menggunakan teknologi *container* karena memiliki *overhead* lebih kecil sehingga memiliki performansi yang jauh lebih baik. Dengan dilakukannya ujicoba perbandingan, maka dapat diketahui performa dari masing-masing *container* tersebut. Pada perbandingan yang dilakukan adalah perbandingan secara *overall performance* yang meliputi penggunaan CPU dan RAM tiap-tiap *container* dengan alat uji HTOP serta melakukan pengujian tiap-tiap *container* terhadap FTP server, web server serta mail server, yang meliputi *throughput*, Jeda waktu/ *latency*, serta *transfer rate*. Hasil rata-rata pengujian maka dapat menghasilkan *output* berupa grafik performa dari *container Docker*, dari hasil pengujian *docker* termasuk berat dalam penggunaan RAM yang mencapai rata-rata diatas 200Mb namun untuk performa CPU *docker* sangat stabil, sedangkan untuk performa LXC dan LXD lebih ke ringan penggunaannya, mudah untuk dipelajari, dan sedikit lebih unggul dibanding *Docker* dengan penggunaan RAM yang tidak sampai mencapai 100 Mb serta keunggulan dalam transfer data FTP server, hanya saja LXC dan LXD lemah pada sisi performa yang mudah naik turun dan tidak stabil

Kata Kunci— *virtualisasi, FTP Server, Web Server, Mailserver, container, docker container, LXC, LXD, HTOP, apache jmeter.*

I. PENDAHULUAN

Teknologi saat ini telah berkembang sangatlah pesat, yang dari tahun ke tahun memberi dampak serta perubahan besar bagi manusia agar dapat memberikan keuntungan dan mempermudah pekerjaan. Salah satu teknologi yang dapat mempermudah pekerjaan adalah teknologi virtualisasi. Virtualisasi merupakan sebuah teknologi yang memungkinkan untuk menghasilkan suatu bentuk virtual dari sesuatu yang bersifat fisik. Salah Satu teknologi virtualisasi yang banyak digunakan adalah cloud computing. Menurut Foster *Cloud Computing* adalah Paradigma komputasi terdistribusi dalam skala yang besar yang dilatar belakangi oleh faktor ekonomi, yang mana berisi kumpulan dari virtualisasi abstrak, pengaturan kekuatan komputasi, tempat penyimpanan, platform, dan layanan yang dapat diakses sesuai dengan kebutuhan oleh pelanggan eksternal melalui media Internet [1].

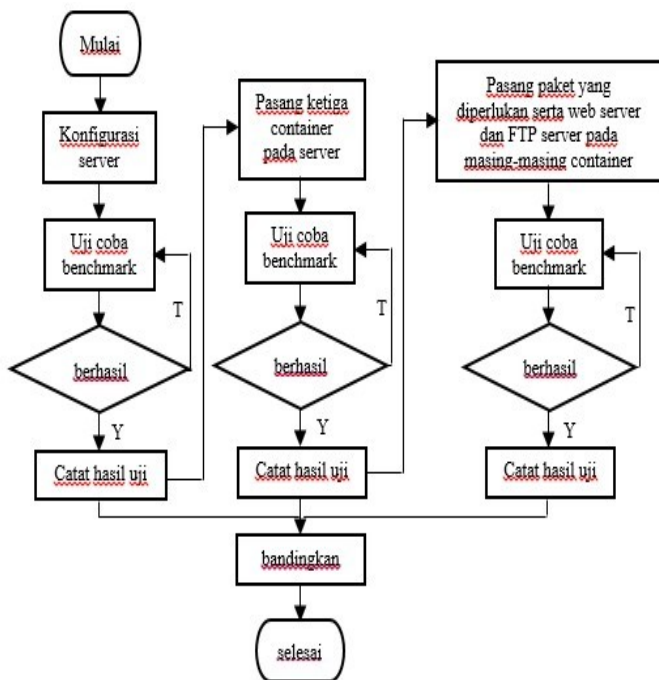
Cloud computing bergantung dengan adanya teknologi virtualisasi yang membutuhkan sumber daya seperti *server*. untuk melakukan pengimplementasian VM dapat memberikan dampak yang sangat besar yaitu beban yang berat akan mengganggu . kinerja *overall performance* dikarenakan VM terkenal memiliki *overhead* yang besar sehingga sangat mempengaruhi performanya, dikarenakan *Hypervisor* yang membuat berat. Dengan adanya dampak tersebut, dibandingkan menggunakan VM, teknologi *container* dapat dijadikan masukan, dikarenakan *overhead* pada *container* lebih kecil sehingga memiliki performa yang jauh lebih baik dari segi entitas virtualisasi[2].

Container merupakan sebuah *lightweight platform* yang merupakan suatu *platform* yang digunakan untuk melakukan proses migrasi data yang dapat berjalan dengan sangat ringan ,karena sebuah program akan diikat beserta *library*-nya, file konfigurasi, dan seluruh hal yang dibutuhkan pada program tersebut.[3] *Container* memungkinkan pengguna untuk menghemat konsumsi sumber daya tanpa adanya *overhead* virtualisasi serta menyediakan isolasi. Selain hal itu VM melakukan *boot up* pada *kernel* nya sendiri, sedangkan *container* menggunakan *kernel* yang dimiliki host, yang menjadikan suatu *container* cenderung lebih cepat dibanding dengan VM.[4]

Dari perkembangan itu munculah berbagai nama produk *container* dari berbagai macam *developer*. Mulai dari LXC, LXD, Docker, kubernetes serta masih banyak lagi dengan keunggulan serta kekurangan di berbagai macamnya. Untuk mengetahui performa beberapa *container*, dilakukan perbandingan dari segi performa 3 aplikasi *container* yang populer antara lain Docker, LXC, serta LXD dengan media perbandingnya adalah pada web server, FTP server, dan juga mail server [5]. yang akan di uji performanya pada masing-masing *container*. serta melakukan pengujian performa RAM, CPU dari masing-masing *container*. Hal ini nanti akan menghasilkan output grafik dari masing-masing Docker, LXC dan LXD serta mendeskripsikan keunggulan ataupun kekurangan tiap masing-masing *container* yang diukur menggunakan *benchmark tools open source* yang tersedia.

II. METODOLOGI

Metode penelitian pada analisis performansi 3 *container* ini berisi alur pengujian performansi *container* pada web server, FTP server dan mail server sseperti pada Gbr. 1.

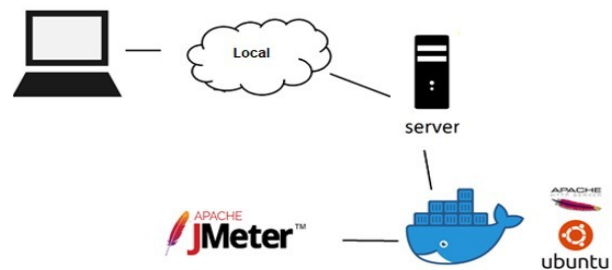


Gbr. 1. Alur pengujian performansi 3 container pada web server dan FTP server

Pada Gbr. 1, *container* akan dijalankan secara bergantian untuk melakukan uji coba. Pertama melakukan konfigurasi pada *container* Docker mulai dari meminta izin pada apt untuk dapat mengakses Docker pada Ubuntu, kemudian atur repositori, lalu mengupdatenya hingga dapat menjalankan *container* Docker. Kemudian membangun *container* dengan citra Ubuntu untuk mengakses FTP server di dalamnya. Setelah pengujian *throughput*, *transfer rate*, dan jeda waktu maka akan dilakukan pengujian *benchmark* dari masing-masing proses tersebut, yang nantinya akan dijadikan perbandingan performa ketiga docker tersebut. Setelah berhasil lanjut melakukan konfigurasi pada LXC dengan cara *update package index* lalu membangun *container* berbasis Ubuntu, dan memulai *container* tersebut, dan untuk langkah selanjutnya sama seperti Docker. Langkah terakhir melakukan konfigurasi pada LXD dengan cara mengupdate terlebih dahulu, lalu mengkonfigurasi *storage backend* pada file system, dilanjut dengan inisialisasi LXD dan kemudian baru dapat dibuat *container* dengan basis Ubuntu. Proses selanjutnya kurang lebih hampir sama. Setelah ketiganya dilakukan uji coba maka di lakukan analisis parameter yang ditentukan adalah penggunaan RAM, CPU tiap masing-masing *container* yang diuji menggunakan HTOP, dan untuk pengujian performa *throughput*, pengiriman data, serta jeda waktu pengiriman/*latency* dilakukan pengujian menggunakan Apache jmeter.

A. Topologi pengujian Web Server

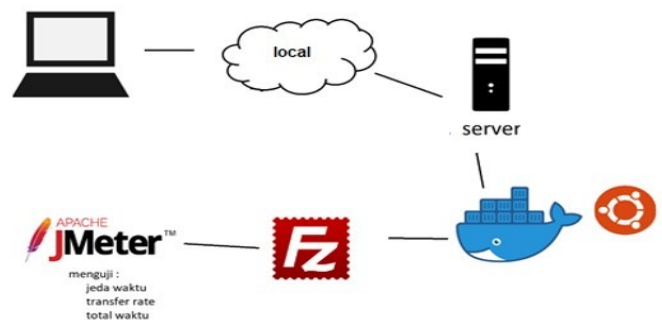
Pada Gbr. 2 merupakan rangkaian yang akan diujikan dalam menjalankan *Web server* pada *container*. Proses awal *server* dikoneksikan dengan internet yang kemudian dilakukan konfigurasi IP yang akan dilakukan pada *client*. Setelah terhubung *client* mengaktifkan *container* dan menginstal aplikasi *apache2* untuk citra Ubuntu untuk *webserver* yang kemudian dilakukan pengujian dan di ukur menggunakan aplikasi *apache jmeter* dalam pengukuran *benchmark* pada *web server*nya dengan mengirimkan *thread* dari aplikasi Jmeter.



Gbr. 2. Topologi pengujian web server

B. Topologi pengujian FTP Server

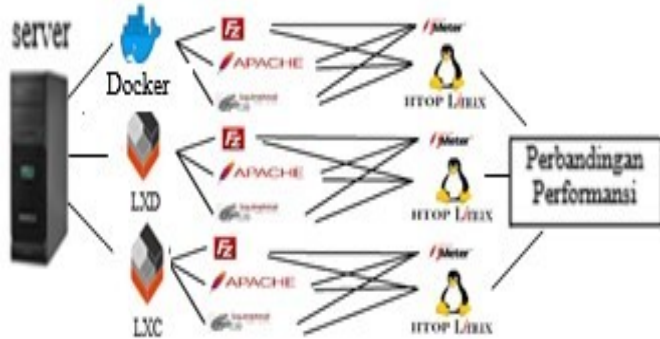
Pada Gbr. 3, *server* melakukan konfigurasi dengan instalasi *proftpd* sebagai *server*nya, *container*, dan aplikasi yang diperlukan untuk pengujian yang kemudian dihubungkan dengan jaringan lokal dan mengkonfigurasi IP, agar dapat dibuka pada *client* 3 *container* tersebut dan menjalankan *proftpd* menggunakan citra Ubuntu yang telah tersedia pada registry dari *container* yang didalamnya berisi aplikasi-aplikasi yang dapat dibuka pada *container*. Setelah masuk *proftpd* maka dilakukan pengujian menggunakan *filezilla* dengan mengujikan waktu pengiriman, jeda waktu pada saat pengiriman, serta *throughput* pengiriman. Ketiga *container* akan di bandingkan dengan melihat masing-masing performa pada *apache jmeter*.



Gbr. 3. Topologi pengujian FTP server

C. Perancangan Topologi Jaringan

Topologi yang akan digunakan dalam penelitian ini dapat dilihat pada Gbr. 4 yaitu melakukan konfigurasi pada masing-masing kontainer, yang didalamnya telah di bangun ruang penyimpanan untuk digunakan sebagai penampung *web server*, *FTP server* dan *mail server* yang akan di uji dengan dua alat pengujian yaitu *Apache jmeter* dan *Htop* dari linux.



Gbr. 4. Perancangan Topologi perbandingan performansi *container* Docker, LXC dan LXK

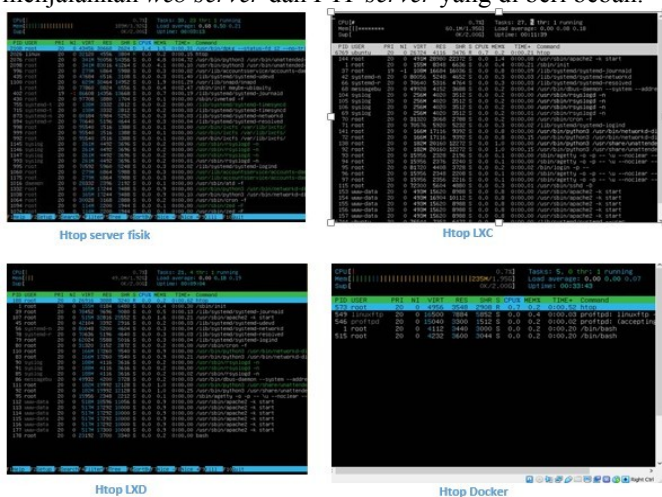
III. HASIL DAN PEMBAHASAN

A. Uji coba dan Analisis Data

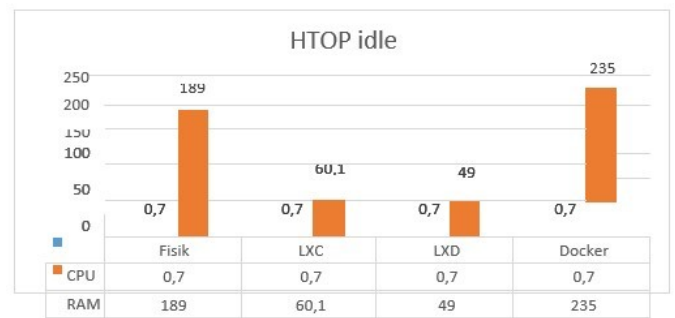
Pada tahap ini dilakukan uji coba performansi *container* pada *web server*, *FTP server* dan *mail server*.

a. Skenario pertama

Skenario pertama dilakukan pada *server* yang telah di jadikan 3 *container* yaitu Docker, LXC, dan LXK, dengan mengirimkan *thread* ke tiap-tiap *container*, untuk dilakukan pengecekan mulai dari kondisi *idle* hingga saat di beri beban *thread*, sehingga dapat dilihat sebagai berikut, untuk Gbr. 5 merupakan gambar performa *server* disaat normal tanpa menjalankan aktivitas untuk *Htop server* fisik, LXC, LXK dan Docker sedangkan untuk Gbr. 7 adalah gambar *server* menjalankan *web server* dan *FTP server* yang di beri beban.

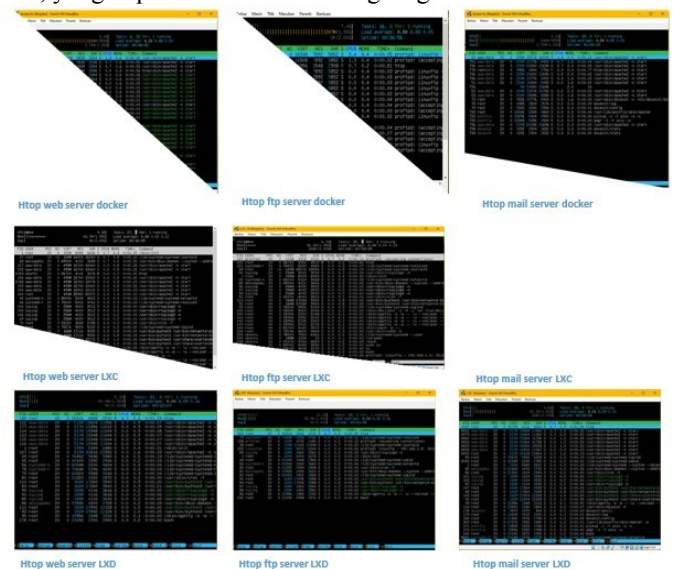


Gbr. 5. Htop masing-masing *container* saat *idle*

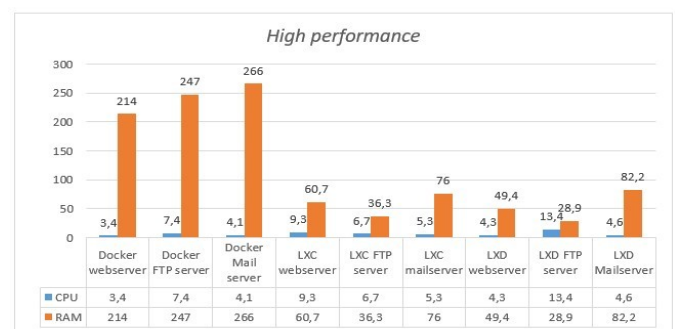


Gbr. 6. Perbandingan htop kondisi *idle*

Dari uji coba tersebut, menunjukkan kinerja CPU dan RAM selama kondisi normal tanpa diberi beban. Dari grafik berikut pada Gbr. 6, penggunaan rata-rata CPU semuanya sama namun untuk RAM yang digunakan total rata-rata penggunaan RAM tertinggi ada pada Docker dengan penggunaan rata-rata 235 Mb, sedangkan untuk penggunaan RAM terendah ada pada LXK dan LXC yang beda tipis, untuk LXC penggunaan rata-rata 60,1 Mb sedangkan untuk penggunaan RAM LXK merupakan yang paling kecil yaitu 49 mb, yang dapat dikatakan terbilang ringan.



Gbr. 7. Htop masing-masing *container* saat *high performance*



Gbr. 8. Perbandingan htop saat *high performance*

Uji coba pada Gbr. 7 menunjukkan kinerja CPU dan RAM selama kondisi ditekan dengan dilakukan pengiriman *thread* terus-menerus sebanyak 500 data untuk *webserver* dan *mail server*, 50 file untuk *FTP server*. Dari grafik pada Gbr. 8, penggunaan rata-rata CPU terendah ada pada *web server* pada docker sebesar 3,4% namun rata-rata penggunaan RAM tertinggi ada pada Docker dengan penggunaan rata-rata diatas 200 Mb, sedangkan untuk penggunaan RAM terendah ada pada LXD dan LXC yang beda tipis masih di bawah 100 Mb.

b. Skenario kedua

Pengujian skenario kedua dilakukan pada *server* yang telah di jadikan 3 *container* yaitu docker, LXC, dan LXD, dengan mengirimkan *thread* ke tiap-tiap *container*, dapat dilihat sebagai berikut:

1. Pengujian *web server*

TABEL I
PENGUJIAN *WEB SERVER*

Rata- rata uji coba	Throughput	Transfer rate (Kb/sec)	Jeda waktu
Docker	5,0/sec	0,57	8
LXC	5,0/sec	0,58	9
LXD	5,0/sec	0,57	9

2. Pengujian *FTP server*

TABEL III
PENGUJIAN *FTP SERVER*

Rata- rata uji coba	Throughput	Transfer rate (Kb/sec)	Jeda waktu
Docker	30,6/min	18,41	17
LXC	30,6/min	20,31	15
LXD	30,6/min	21,30	14

3. Pengujian *mail server*

TABEL IIIII
PENGUJIAN *MAIL SERVER*

Rata- rata uji coba	Throughput	Transfer rate (Kb/sec)	Jeda waktu
Docker	5,0/min	0,57	9
LXC	5,0/min	0,58	8
LXD	5,0/sec	0,57	9

Performansi masing-masing *container* tidak begitu signifikan perbedaannya. Pada tabel I, pengujian *web server* masing-masing *container* memiliki *throughput* yang sama yaitu 5,0/sec dimana dalam pengiriman, data yang dapat dikirimkan hanya selisih 1 kb/sec, jeda waktu docker yang hanya selisih 1 sec. Untuk pengujian *FTP server* pada tabel II, merupakan pengujian yang paling terlihat perbedaannya, pada

pengujian *FTP server* docker merupakan yang paling lama dalam pengiriman, dengan transfer rate hanya 18,41 kb/sec dan jeda waktu butuh 17 sec dalam pengiriman datanya. Sedangkan untuk pengujian *mail server* pada tabel III, ketiganya tidak jauh berbeda dari hasil pengujian *web server*.

IV. KESIMPULAN

Container Docker memiliki kelebihan stabil dalam menjaga kinerja CPU sehingga tidak terjadi kenaikan signifikan, serta dalam paket penginstalan terdapat banyak image yang dapat langsung di terapkan dan digunakan dengan menginstallnya. Namun dalam pengerjaan Docker merupakan yang paling susah dikarenakan harus memasukkan port manual, serta penggunaan RAM yang tinggi yang rata-rata pengujiannya selalu ada pada 200 Mb. Sedangkan untuk *Container* LXC dan LXD merupakan *container* yang tidak jauh beda sehingga kinerja dalam uji coba tidak jauh berbeda juga. Kelebihan dari LXC adalah kemudahan dalam penerapan, kinerja yang digunakan sangatlah ringan karena hanya memerlukan aplikasi pendukung yang diperlukan saja dalam pengoperasiannya, namun dalam segi penggunaan LXC dan LXD bisa dibilang sangat tidak stabil karena untuk CPU yang digunakan sering naik dan turun secara drastis. Untuk pengujian *throughput*, jeda waktu pengiriman serta transfer rate, keduanya tidak jauh beda namun pada *Ftp server* LXD lebih cepat dibanding LXC dan docker.

REFERENSI

- [1] Foster, Ian (2008). Cloud Computing and Grid Computing 360-Degree Compared. Chicago: University of Chicago
- [2] Rabindra K Barik, Rakesh K. Lenka, K. Rahul Rao, Devam Ghose, "Performance Analysis of Virtual Machine and Containers in Cloud Computing", International Conference on Computing, Communication and Automation, pp. 1204-1210, 2016.
- [3] Rajdeep Dua, A. Reddy Raja, Dharmesh Kakadia, "Virtualization vs Containerization to support PaaS", IEEE International Conference on Cloud Engineering, pp. 610-614, 2014
- [4] Miguel G. Xavier, Marcelo V. Neves, Fabio D. Rossi, Tiago C. Ferreto, "Performance Evaluation of Container-Based Virtualization for High Performance Computing Environments", 2013 21st Euromicro International Conference on Parallel, Distributed, and Network-Based Processing, pp. 233- 240. 2013
- [5] Adiputra, Firmansyah (2015). Container dan Docker: teknik virtualisasi dalam pengelolaan banyak aplikasi *web*. Madura: Universitas Trunojoyo jurnal simatic vol.4 no 3 ISSN 2088-2130
- [6] Carlos Arango dkk. (2017). Performance Evaluation of Container-based Virtualization for High Performance Computing Environments. Colombia: universidad del valle
- [7] Harijanto, Budi (2015). Desain dan Analisis Kinerja Virtualisasi *Server* Menggunakan PROXMOX Virtual Environment. Malang: jurnal simatic vol.5 no 1 ISSN 2088-2130
- [8] Kurniawan, Erick (2015). Penerapan Teknologi Cloud Computing di Universitas Studi Kasus: Fakultas Teknologi Informasi UKDW.yogyakarta: Jurnal EKSIS Vol 08 No 01 page 29-36
- [9] Pratama, Rivaldy Arif dkk. (2018). Implementasi *Web Server* Cluster Menggunakan Metode Load Balancing pada Container Docker, LXC, DAN LXD. Bandung: Telkom university, e-Proceeding of Engineering Vol.5, No.3 ISSN : 2355-9365
- [10] Ravis, Muhammad dkk. (2019). Perbandingan Performansi Single *Web Server* Dan Multi *Web Server* Dengan Metode Paired Sample TTest.

- Palembang, STMIK GI MDP, Jurnal SISFOKOM, Volume 08, Nomor 02 ISSN : 2301-7988.
- [11] Putri, Adinda Riztia dkk. l(2020). Performance Analysis of Multi Services on Container Docker, LXC, and LXD. Bulletin of Electrical Engineering and Informatics vol. 9 no. 5, October, pp. 2008 – 2011 ISSN: 2302-9285
 - [12] Ruwaida, Devi dkk. (2018). RANCANG BANGUN FILE TRANSFER PROTOCOL (FTP) DENGAN PENGAMANAN OPEN SSL PADA JARINGAN VPN MIKROTIK DI SMKS DWIWARNA. Medan, journal of computer engineering system and science vol 3 no 1 page 45-49