

RASPBERRY PI SEBAGAI STORAGE VERSION CONTROL SYSTEM TUGAS MAHASISWA POLITEKNIK NEGERI MALANG

Rifki Hari Romadhon ^{a,1*}, Ir. Nugroho Suharto, MT ^{a,2}, Aad Hariyadi ^{a,3}

^a Jaringan Telekomunikasi Digital, Soekarno Hatta, Malang, Indonesia

¹ rifkyhari@gmail.com*; ² nugroho.suharto@polinema.ac.id; ³ aad.hariyadi@polinema.ac.id

* Penulis Koresponden

INFO ARTIKEL

Histori Artikel

19-05-2022

26-05-2022

07-07-2022

Kata Kunci

Version Control System

Transfer File

QOS

ABSTRAK

Teknologi *transfer file* kini mulai berkembang pesat banyak dari peneliti mulai mengembangkan metode pada *transfer file*, saat ini *transfer file* memang hadir di kehidupan sehari-hari dalam berbagai macam metode. Contohnya masih banyaknya penggunaan *flashdisk* dalam kehidupan manusia untuk proses memindahkan *file*. Hal ini membuat riskan terkena virus dari komputer atau laptop lain yang menyebabkan *file* tersebut dapat hilang. Pada penelitian ini merancang sebuah *file transfer* yang sekaligus dapat mengontrol dari penggunaannya untuk menerapkan kontrol versi dari sebuah *file* supaya *file* tersebut tetap tercatat dalam *repository* dan memiliki versi tersendiri dari setiap *file* yang diunggah. Dengan menerapkan *Version Control System* diharapkan dapat membantu mahasiswa untuk menyimpan semua *file* pada *repository* yang dikerjakan dan memiliki perubahan setiap *file*. Penelitian ini diharapkan akan membantu memberikan alternative dalam melakukan penilaian oleh setiap dosen, serta dapat mempercepat dan memudahkan pengambilan keputusan yang terkait mengenai penilaian terhadap siswa di Politeknik Negeri Malang. Hasil dari penelitian ini menunjukkan QoS *transfer file* dalam jaringan polinema memiliki nilai *packet loss* 0%, *delay* 0,672ms (*upload*) dan 0,447ms (*download*), serta *throughput* 330kbps (*upload*), 544,50 (*download*).

Ini adalah artikel akses terbuka di bawah lisensi [CC-BY-SA](#).



1. Pendahuluan

Perkembangan teknologi informasi dan komunikasi mengalami kemajuan yang pesat. Banyak perangkat dan aplikasi diciptakan untuk mempermudah pekerjaan manusia, sehingga sebuah informasi dapat diakses dengan cepat. Pesatnya perkembangan teknologi jaringan komputer telah menciptakan kemungkinan untuk menyediakan layanan *storage* dengan mengimplementasikan *control version system* yang dapat dirancang menggunakan *hardware* dengan biaya terjangkau. Salah satu *hardware* yang dapat mengimplementasikan *storage* dengan biaya terjangkau adalah *Raspberry Pi* [1].

Version control system atau juga dikenal dengan *revision control system* adalah sebuah sistem yang mencatat dan melacak setiap perubahan (*revisi*) yang terjadi pada sebuah *file*. Sistem ini akan memonitor dan mencatat perubahan yang terjadi terhadap *file*

termasuk siapa yang melakukan perubahan, kapan perubahan terjadi, mengapa dilakukan perubahan dan sebagainya. Mengapa dibutuhkan *version control system*? *Version control system* adalah hal yang sangat penting dalam pengembangan perangkat, khususnya yang sifatnya kolaboratif dan terdistribusi. Pada umumnya pengembangan perangkat ini melibatkan lebih dari satu orang saja, dan umumnya dilakukan secara tim [2]. Tim ini bekerja secara paralel mengolah sekumpulan *file* (baik itu *source code* atau dokumentasi), yang sama dan biasanya dalam waktu yang lama. Karena masing-masing anggota tim mengerjakan *file-file* yang sama, maka ada beberapa hal yang harus diperhatikan, antara lain:

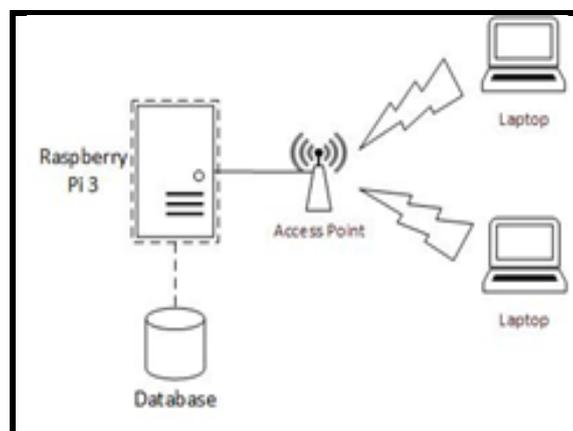
1. Dibutuhkan pelacakan terhadap siapa yang melakukan perubahan, agar setiap perubahan bisa dipertanggung jawabkan;
2. Perubahan-perubahan yang terjadi terhadap *file* harus dicatat sehingga kita bisa mengembalikan *file* ke kondisi sebelum diubah jika terjadi kesalahan;
3. Dan yang tidak kalah pentingnya adalah bagaimana menggabungkan perubahan-perubahan yang dilakukan anggota tim agar bisa dikolaborasikan dan tidak tumpang tindih.

Dalam penelitian ini *version control system* diharapkan dapat menjadi solusi untuk melakukan pengaturan tugas mahasiswa berdasarkan versi serta perubahan yang sudah terjadi dalam dokumen tersebut [3]. Nantinya dokumen tersebut memberikan versi yang berbeda ketika adanya perubahan *major* dan *minor* pada dokumen yang diperbaharui. Perubahan *major* adalah perubahan bersifat besar contohnya pada dokumen terjadi perubahan pada bab baru, dan judul baru. Sedangkan perubahan minor adalah perubahan yang bersifat kecil contohnya adanya penambahan isi dari sub bab tanpa harus mengganti bab tersebut, menambahkan daftar isi dan menambahkan daftar pustaka [4].

2. Metode Penelitian

2.1. Skema Penelitian

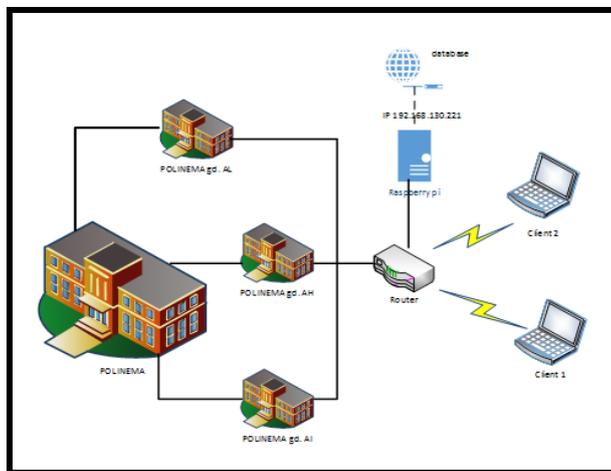
Metode *version control system* adalah sebuah sistem yang mencatat dan melacak setiap perubahan (revisi) yang terjadi pada sebuah *file*. Sistem ini akan memonitor dan mencatat perubahan yang terjadi terhadap *file* termasuk siapa yang melakukan perubahan, kapan perubahan terjadi, mengapa dilakukan perubahan dan sebagainya [5].



Gambar 1. Perancangan Sistem

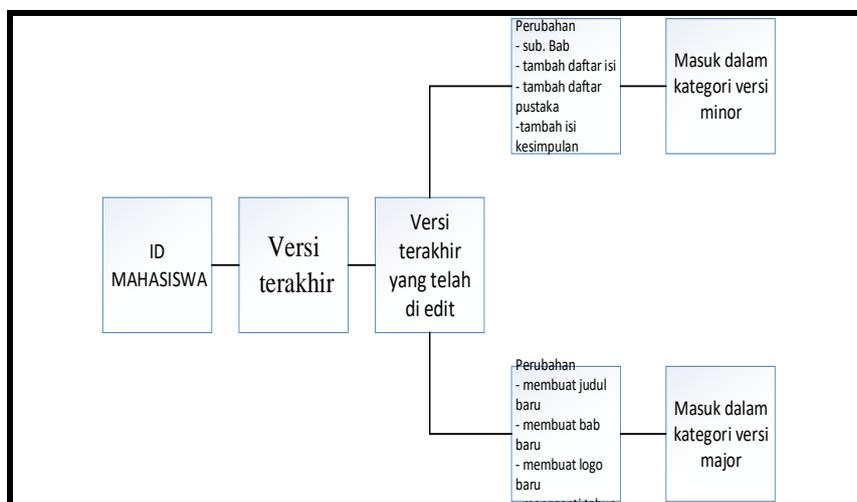
Pada Gambar 1. menunjukkan proses dari metode yang akan digunakan. *Raspberry Pi* sebagai *server* bertugas untuk menampung *file storage* yang telah diunggah oleh *client* yang

nantinya akan diteruskan oleh *raspberry* menuju *database* yang telah dibuat. Dalam hal ini peranan *access point* sangatlah penting untuk menunjang proses terjadinya *transfer file* antara *client* dengan *server* [6]. Metode ini akan diimplementasikan ke dalam jaringan Politeknik Negeri Malang, pada gambar2 akan dijelaskan diagram keseluruhan sistem jaringan.



Gambar 2. Diagram Keseluruhan Sistem

Pada Gambar 2. dijelaskan proses sistem secara keseluruhan menggunakan jaringan yang berada didalam Politeknik Negeri Malang. Memanfaatkan *router* yang telah terpasang pada jaringan di Politeknik Negeri Malang *raspberry pi* terlebih dahulu di setting *ip static* untuk dapat masuk ke dalam jaringan Politeknik Negeri Malang. *Ip static* di setting pada “192.168.130. 221”. Kemudian *raspberry pi* dihubungkan melalui kabel LAN untuk dapat tersambung ke *router* [7]. Setelah itu *client* dapat melakukan proses *transfer file* namun terlebih dahulu harus terkoneksi dengan jaringan yang berada didalam Politeknik Negeri Malang. *File* yang telah di unggah akan masuk ke dalam *database* melalui *raspberry pi*, *raspberry pi* berfungsi menyimpan *file* yang telah diunggah oleh *client* dan akan diteruskan ke dalam *database* apabila *client* mengupload melalui *web* yang terhubung dengan *raspberry pi*. Didalam *database* ini sekaligus mengatur *version control system*, *client* akan mengatur versi *file* sesuai dengan keinginan masing-masing. Namun harus diketahui untuk mencegah terjadinya versi yang sama maka ditetapkan aturan untuk mengunggah file yang telah diperbaharui. Berikut aturan *version control system*:



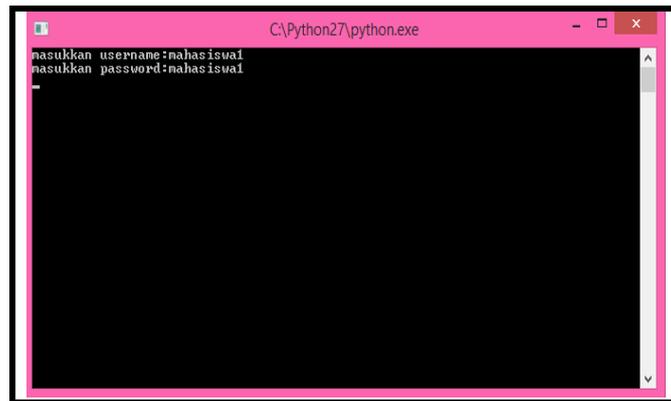
Gambar 3. Blog Diagram VCS

Pada Gambar 3. dijelaskan bahwa perubahan dilakukan *minor* dan *major*, perubahan minor meliputi perubahan pada sub bab, menambah isi dari daftar pustaka dan seterusnya. Perubahan major meliputi perubahan pada membuat judul baru, membuat logo baru dan seterusnya.

3. Hasil dan Analisis

3.1. Hasil

3.1.1. Tampilan Program Phyton



Gambar 4. Tampilan Program Phyton

Merupakan tampilan awal sebelum melakukan proses transfer *file* atau dokumen antar *client* diharapkan memasukan *user name* dan *password* masing-masing.

3.1.2. Tampilan Database Web



Gambar 5. Tampilan Web

Gambar 3.2. menunjukkan tampilan *Interface web database* untuk dapat masuk ke dalam database yang terhubung dalam *server raspberry pi*.

3.1.3. Pengujian Packet Loss

Berikut adalah tabel hasil uji *packet loss* menggunakan aplikasi *wireshark*.

Tabel 1. Hasil Uji Packet Loss

Download			Upload		
Interval Start	All Pakets	TCP Errors	Interval Start	All Pakets	TCP Errors
22	13	0	22	1	0

Download			Upload		
Interval Start	All Pakets	TCP Errors	Interval Start	All Pakets	TCP Errors
23	1	0	23	0	0
24	0	0	24	0	0
25	1	0	25	1	0
26	0	0	26	1	0
27	3	0	27	1	0
28	1	0	28	0	0
29	1	0	29	15	0
30	0	0	30	2	0
31	0	0	31	2	0
32	1	0	32	1	0
33	1	0	33	1	0
34	0	0	34	3	0
35	1	0	35	4	0
36	0	0	36	1	0
37	1	0	37	0	0
38	1	0	38	0	0
39	0	0	39	17	0
40	1	0	40	11	0
41	0	0	41	12	0
42	1	0	42	11	0
43	1	0	43	10	0
44	0	0	44	10	0
45	1	0	45	47	0
46	0	0	46	11	0
47	1	0	47	11	0
48	0	0	48	10	0
49	1	0	49	10	0
50	1	0	50	11	0
51	0	0	51	11	0
52	1	0	52	11	0
53	1	0	53	10	0
54	0	0	54	10	0
55	1	0	55	11	0
56	1	0	56	11	0
57	245	0	57	11	0
58	2	0	58	10	0
59	3	0	59	10	0
60	2	0	60	11	0
Total	288	0	Total	310	0

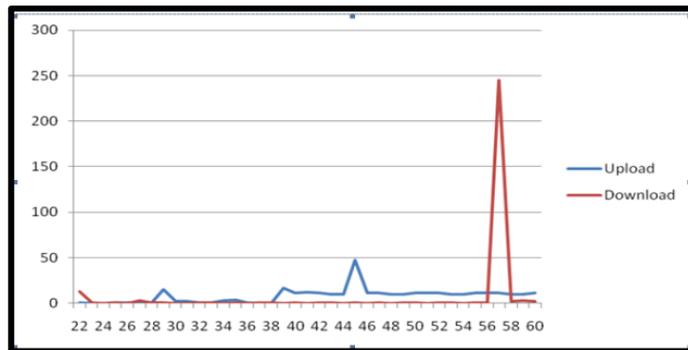
Pengujian ini dilakukan [8] untuk mengetahui *packet* data yang diterima dan *transfer file* yang gagal atau cacat. Hasil yang diperoleh lalu dihitung menggunakan rumus:

$$\frac{TCP\ Error}{All\ Packet} \times 100\% = Packet\ Loss \quad (1)$$

Sehingga memperoleh hasil

$$Download \quad \frac{0}{288} \times 100\% = 0\%$$

$$Upload \quad \frac{0}{310} \times 100\% = 0\%$$



Gambar 6. Grafik Uji Packet Loss

3.1.4. Pengujian Throughput

Hasil Pengukuran *throughput* dibagi menjadi dua yaitu *upload* dan *download* yang dapat dilihat dari capture dalam fitur aplikasi *wireshark*.

Statistics			
Measurement	Captured	Displayed	Marked
Packets	357	48 (13.4%)	N/A
Time span, s	82.524	32.269	N/A
Average pps	4.3	1.5	N/A
Average packet size, B	127.5	222.5	N/A
Bytes	45565	10657 (23.4%)	0
Average bytes/s	552	330	N/A
Average bits/s	4417	2642	N/A

Gambar 7. Throughput Upload

Setelah mendapatkan nilai dari *Time Spandan Bytes*, maka dapat dihitung dengan menggunakan rumus:

$$\frac{\text{Bytes}}{\text{Time Span}} = \text{Throughput}$$

$$\frac{1068}{32.269} = 330.25\text{kbps}$$

Statistics			
Measurement	Captured	Displayed	Marked
Packets	245	38 (15.5%)	N/A
Time span, s	17.339	17.100	N/A
Average pps	14.1	2.2	N/A
Average packet size, B	158.5	245.5	N/A
Bytes	38946	9311 (23.9%)	0
Average bytes/s	2246	544	N/A
Average bits/s	17 k	4355	N/A

Gambar 8. Throughput Download

Dapat dihitung menggunakan rumus:

$$\frac{\text{Bytes}}{\text{Time Span}} = \text{Throughput} \quad (2)$$

$$\frac{9311}{17.100} = 544.50 \text{ kbps}$$

3.1.5. Pengujian Delay

Pengujian *delay* dilakukan pada tahap *upload* dan *download* pada saat transfer file terjadi.

Statistics			
Measurement	Captured	Displayed	Marked
Packets	357	48 (13.4%)	N/A
Time span, s	82.524	32.269	N/A
Average pps	4.3	1.5	N/A
Average packet size, B	127.5	222.5	N/A
Bytes	45565	10657 (23.4%)	0
Average bytes/s	552	330	N/A
Average bits/s	4417	2642	N/A

Gambar 9. Delay Upload

Setelah didapatkan nilai dari *Time span* serta *Packets*, maka dapat dihitung menggunakan rumus:

$$\frac{\text{Time Span}}{\text{Packet}} = \text{Delay} \quad (3)$$

$$\frac{32,269}{48} = 0,672 \text{ ms}$$

Statistics			
Measurement	Captured	Displayed	Marked
Packets	245	38 (15.5%)	N/A
Time span, s	17.339	17.100	N/A
Average pps	14.1	2.2	N/A
Average packet size, B	158.5	245.5	N/A
Bytes	38946	9311 (23.9%)	0
Average bytes/s	2246	544	N/A
Average bits/s	17 k	4355	N/A

Gambar 10. Delay Download

Berdasarkan pengujian yang telah dilakukan maka diperoleh nilai *Time span* dan *Packets* yang diterima, sehingga dapat dihitung menggunakan rumus:

$$\frac{\text{Time Span}}{\text{Packet}} = \text{Delay} \quad (4)$$

$$\frac{17.000}{38} = 0.447 \text{ ms}$$

3.1.6. Pengujian CPU Load dan RAM Usage

Melalui fitur *command htop* yang disediakan di dalam *raspberry pi* dapat menunjukkan hasil dari *cpu load* dan *ram usage*. Didapatkan hasil seperti berikut ini:

```

1  [ 0.0%]
2  [ | 1.3%]
3  [ | 0.6%]
4  [ || 3.3%]
Mem [ ||||| | 185M/927M]
Swp [ 0K/100.0M]

```

Gambar 11. cpu load dan ram usage

Berdasarkan hasil pengujian *cpu load* bekerja pada 4 titik *core*, masing-masing memiliki nilai yang berbeda, sedangkan *ram usage* telah digunakan sebesar 185mb dari total 927mb.

4. Kesimpulan

Dari hasil perancangan dan pengujian yang sudah dilakukan maka dapat ditarik kesimpulan sebagai berikut.

1. Server dari *file storage* dapat dibangun dalam *raspberry pi* versi 3 untuk menciptakan system yang dapat menampung *database* didalamnya.
2. Dari hasil pengujian performansi QoS ketika *upload* dan *download* didapatkan kualitas performansi *packet loss* sangat baik dengan nilai *upload* dan *download* sebesar 0%, nilai throughput saat *upload* dan *download* sangat baik dengan nilai 330,25 kbps dan 544,50, nilai delay saat *upload* dan *download* sangat baik dengan nilai 0,672ms dan 0,447ms.
3. Dari performansi *CPU Load* dan *RAM usage*. Didapatkan nilai 4 *core* cpu *core* 1 sebesar (0.0%), *core* 2 (1.3%), *core* 3 (0.6%) dan *core* 4 (3.3%). Sedangkan *RAM usage* yang digunakan sebesar 185M/927M yang artinya *RAM* telah terpakai 185M dari 927M.

Pengakuan dan Penghargaan

Pemberian terimakasih ditujukan kepada Jurusan Jaringan Telekomunikasi Digital Politeknik Negeri Malang yang telah mendukung proses pembuatan jurnal ini dari awal sampai akhir.

Daftar Pustaka

- [1] M. P. Mahardiyanto, "Analisa Performansi Keamanan Jaringan Vpn Pptp Dan L2Tp/Ipsec Untuk Ftp Server Di Politeknik Negeri Malang," *J. Jar. Telekomun.*, vol. 3, no. 2, pp. 26–32, 2016.
- [2] S. Aisa, "Implementasi Private Cloud Menggunakan Raspberrry PI Untuk Pengaksesan Data Pribadi," *J. Penelit. Pos dan Inform.*, vol. 6, no. 2, pp. 137–152, 2016.
- [3] J. Young, M. Galinium, and J. Purnama, "PENGEMBANGAN APLIKASI MOBILE BAGI PENYANDANG CEREBRAL PALSY DALAM AKTIFITAS SEHARI-HARI."
- [4] M. M. Noor, N. Dengen, and E. Budiman, "Repository Tugas Dan Bahan Ajar Menggunakan Layanan Cloud Storage Pada Fakultas Ilmu Komputer Dan Teknologi Informasi," in *Prosiding Seminar Ilmu Komputer Dan Teknologi Informasi (Sakti)*, 2017, vol. 2, no. 1.
- [5] F. Sirait, "Sistem Monitoring Keamanan Gedung berbasis Rasberrry Pi," *J. Teknol. Elektro*, vol. 6, no. 1, pp. 55–60, 2015.
- [6] N. N. Zolkifli, A. Ngah, and A. Deraman, "Version control system: A review," *Procedia Comput. Sci.*, vol. 135, pp. 408–415, 2018.
- [7] M. R. Arief, "Pemrograman web dinamis menggunakan php dan mysql," *Yogyakarta Andi*, pp. 7–19, 2011.
- [8] W. Swierstra and A. Löh, "The semantics of version control," in *Proceedings of the 2014 ACM International Symposium on New Ideas, New Paradigms, and Reflections on Programming & Software*, 2014, pp. 43–54.



Rifki Hari Romadhon lahir di Malang 30 Januari 1996, telah berhasil menempuh pendidikan D4 dan S2 di Politeknik Negeri Malang.

Alamat Email: rifkyhari@gmail.com

