DOI: doi.org/10.21009/03.SNF2017.02.CIP.10

ANALISA SISTEM SINKRONISASI FILE DATABASE PADA MULTI-SERVER MENGGUNAKAN AGLETS

Wisnu Maruta[a)], Wisnu Broto[b)]

Jurusan Teknik Elektro, Universitas Pancasila
Jl. Raya Agung, Jagakarsa, Kota Jakarta Selatan, Daerah Khusu Ibukota Jakarta

Email: [a)]wisnu.maruta99@gmail.com, [b)]wisnu.agni@gmail.com

Abstrak

Perkembangan teknologi perangkat keras komputer yang begitu pesat telah mengubah paradigma pengembangan software agar mampu berjalan dalam berbagai platform sistem operasi. Sistem berbasis web server merupakan salah satu aplikasi komputer yang membutuhkan kompatilitas yang tinggi sehingga komputer dengan teknologi hardware dan software yang berbeda dapat berkomunikasi dengan baik. Berbagai penyebab memungkinkan terjadinya masalah pada server tunggal, sehingga perlu dipertimbangkan untuk menggunakan backup server. Backup server berisi data yang sama dengan data dari server utama. Hal tersebut dapat dilakukan dengan cara mensinkronkan file secara berkala. Dalam studi ini, sinkronisasi file diwujudkan menggunakan Aglets. Aglets merupakan framework mobile agent dengan kemampuan berpindah dan melakukan tugas sesuai keinginan. Hasil pengujian menunjukkan sinkronisasi berjalan dengan baik menggunakan prinsip mobile agent. Untuk memaksimalkan kerja sinkronisasi, sebaiknya dua server diletakkan pada satu jaringan untuk mendapatkan kecepatan sinkronisasi yang maksimal. Sinkronisasi file menggunakan data sampai dengan 1200 file berhasil dilakukan.

Kata-kata Kunci: Platform, Backup server, Mobile Agent, Aglets, Sinkronisasi file

Abstract

The rapid development of computer hardware technology has changed the paradigm of software to be able to run on various operating system platform. Web server-based system is one of the computer applications that require jigh compatibility so that compiters with different hardware and software technologies can communicate well. Various causes allow for the occurrence of the problems on a single server, so consideration should be given to using a server backup. The backup server contains the same data as the data from the main server. It can be done by synchronizing files periodically. In this study, file synchronization is realized by using Aglets. Aglets is a mobile agent framework with ability to nove and do tasks as desired. The results of the test shows that synchronization works well using mobile agent principle. To maximize synchronization work, we recommend two servers to be placed on one network to get the maximum synchronization speed. File synchronization using data up to 1200 files successfully done.

Keywords : Platform, Backup server, Mobile Agent, Synchronization file

PENDAHULUAN

Dalam membangun suatu layanan publik, kita akan selalu bertemu atau berhubungan dengan yang namanya server. Server merupakan inti yang sangat penting, misalnya dalam dunia pekerjaan. Dan begitu banyak data yang akan di simpan, sehingga tidak hanya menggunakan satu server saja. Dengan berbagai penyebab yang memungkinkan terjadinya masalah pada suatu layanan atau server tersebut. Maka perlu di pertimbangkan untuk menggunakan *backup server*. *Backup server* berisi tentang data yang sama dengan data dari server utama dan selalu terbaru atau *up-to-date*. Sehingga apabila mengalami atau terjadinya masalah pada server utama, masih memiliki data cadangan terbaru yang siap untuk digunakan sesuai dengan kebutuhan user. Serta mekanisme penyamaan data (data mirroring) ini membutuhkan aplikasi sendiri, yaitu sinkronisasi file. Sinkronisasi file ini perlu dijalankan secara berkala agar memilki data cadangan yang terbaru pada *backup server*.

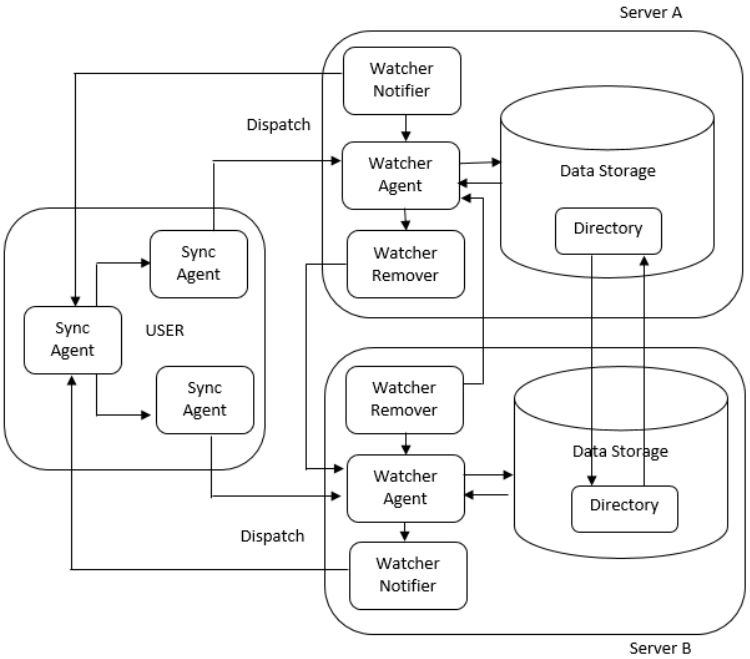
Aglets merupakan suatu Teknologi *Mobile Agent* yang memiliki kemampuan berpindah dari satu host ke host lainnya. Ketika berpindah, state program ini disimpan, dibawa ke host selanjutnya dan berjalan sebagai proses yang normal. Dan dalam hal ini digunakan sebagai media untuk melakukan *sinkronisasi file* antara dua server.

Beberapa riset yang telah dilakukan sebelumnya mengenai teknologi Mobile Agent seperti *Analisa Sistem Keamanan Intrusion Detection System (IDS), Firewall System, Database System dan Monitoring System Menggunakan Agent Bergerak* oleh Bambang Sugiantoro [1], kemudian *Mobile Agent for Networked Electronic Trading* oleh Prithviraj Dasgupta dkk [2] adalah contoh penerapan *Mobile Agent Aglets* pada e-commerce. Contoh lainnya adalah penggunaan Aglets dan JACOB untuk polling halaman web yang dibuat oleh Intan dan Joko [3] dimana agent merekam URL yang berguna (useful) bagi pengguna dan membagikannya kepada pengguna lain. Ada juga yang menggunakan *agent* untuk mendeteksi suatu ancaman tentang keamanan pada jaringan komputer yang dilakukan oleh Anggar Djiwandono.

METODE PENELITIAN

Aplikasi sinkronisasi file ini merupakan aplikasi yang bekerja dengan cara menyamakan isi dari direktori yang telah ditentukan pada Server A dengan isi direktori yang telah ditentukan pada Server B. Server A bertindak sebagai server utama sedangkan Server B sebagai *backup* server. *Graphical User Interface* (GUI) dari aplikasi ini memiliki satu tampilan dari *SyncAgent*. Sedangkan untuk agent yang lain tidak ada. Kemudian untuk notifikasi ditampilkan di command line. Selain GUI dari SyncAgent ada juga Auto Index PHP Script yang digunakan memudahkan user untuk melihat daftar file yang disinkronkan. Auto Index PHP Script merupakan sistem manajemen konten (CMS) open source berbasis web yang dikembangkan menggunakan PHP oleh Justin Hagstorm. CMS ini digunakan untuk mengindeks file dan juga dapat mengunduhnya sesuai keinginan

Pemodelan Arsitektur



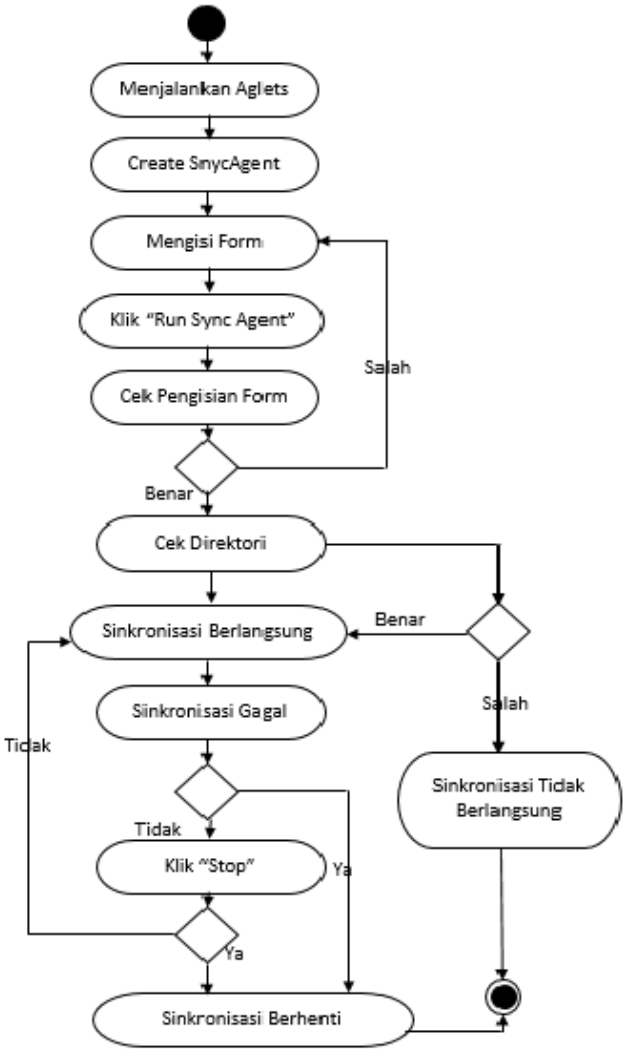
GAMBAR 1. Arsitektur aplikasi sinkronisasi file

Arsitektur dari aplikasi ini dapat dilihat pada Gambar 1. Di Gambar 1 diperlihatkan bagaimana agent dibuat, dikirim kemudian bekerja sama untuk mensinkronkan file. Pada User, SyncAgent membuat dua *WatcherAgent* kemudian masing-masing dikirimkan ke Server A dan Server B. Pada Server A dan Server B masing-masing WatcherAgent mengakses data yang berada di dalam direktori target kemudian saling bekerja sama untuk mensinkronkan data. Apabila *WatcherAgent* mendeteksi ada file yang terhapus, maka *WatcherAgent* akan mengirimkan *WatcherRemover* kepada *WatcherAgent* lainnya. Informasi selama proses sinkronisasi akan dikirimkan oleh *WatcherNotifier*.

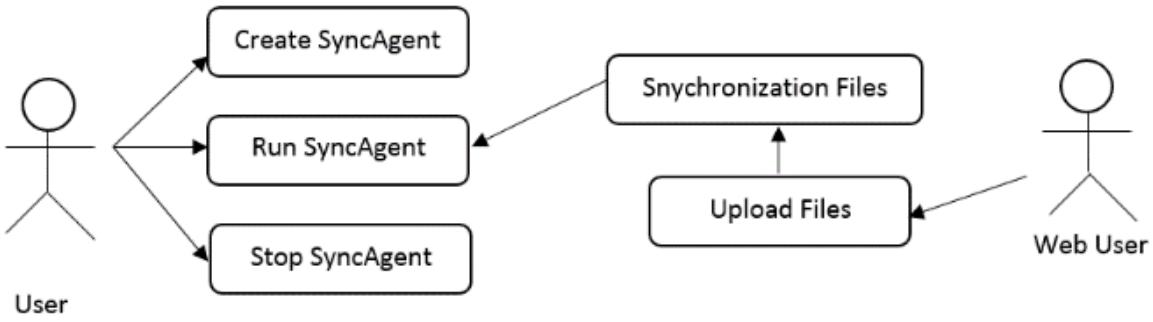
Pemodelan UML

Activity Diagram pada Gambar 2 menunjukkan dimana proses awal sampai akhir yang harus dilakukan untuk menjalankan aplikasi sinkronisasi file melalui Tahiti Server. Berdasarkan Gambar 2 dapat dijelaskan sebagai berikut. Aplikasi dimulai dengan cara menjalankan Aglets. Kemudian membuat *SyncAgent*.

Setelah SyncAgent dibuat maka user perlu mengisi form dengan benar. Apabila ada kesalahan user harus membenarkannya agar *SyncAgent* dapat berjalan. Apabila sudah benar *WatcherAgent* akan dikirim untuk mengecek apakah direktori untuk sinkronisasi ada. Apabila direktori tidak ditemukan maka sinkronisasi tidak dapat berlangsung. Apabila direktori ada maka proses sinkronisasi akan berlangsung. Apabila sinkronisasi gagal maka sinkronisasi akan berhenti. Apabila kerja *SyncAgent* dihentikan maka sinkronisasi juga akan berhenti.



GAMBAR 2. Activity Diagram aplikasi sinkronisasi file



GAMBAR 3. Use Case Diagram aplikasi sinkronisasi file

TABEL 1. Identifikasi Use Case

No	Aktor	Use Case	Deskripsi
1	*User*	*Create SyncAgent*	Pengguna Membuat (create agent) SyncAgent melalui Window tahiti : The Aglets Viewer yang dimiliki Aglets.
2	*User*	*Run SyncAgent*	Pengguna menjalankan SyncAgent untuk mengirimkan WatcherAgent ke server agar dapat melakukan sinkronisasi file.
3	*User*	*Stop SyncAgent*	Pengguna menghentikan sinkronisasi file dengan cara membuang (dispose) WatcherAgent.
4	-	*Synchronizing files*	Sinkronisasi file berjalan pada server. Sinkronisasi dijalankan oleh WatcherAgent.
5	*Web User*	*Upload Files*	Pengguna Web melakukan pengunggahan file pada server.

Use Case diagram dari aplikasi ini ditunjukkan pada Gambar 4 sedangkan deskripsi use case dan aktor-aktor yang terlibat didalamnya, dapat dilihat pada Tabel 1.

HASIL DAN PEMBAHASAN

Pengujian Program

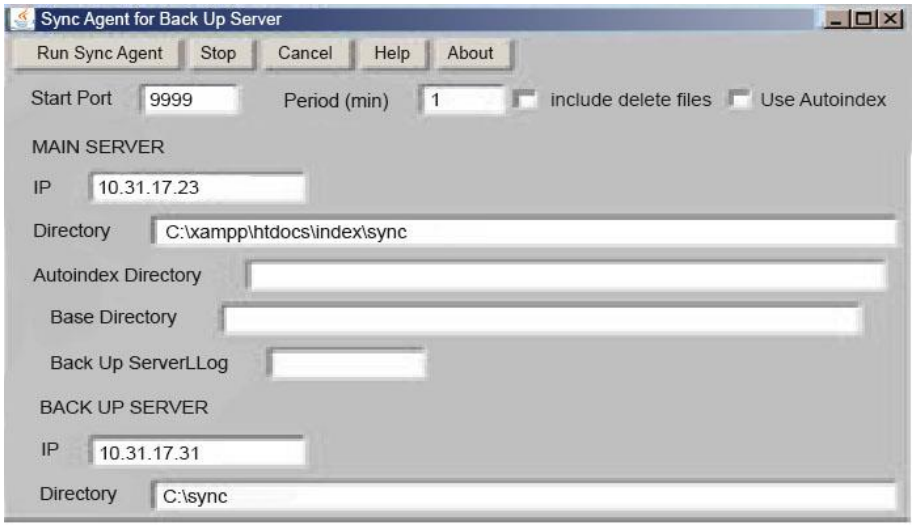
Pada tahap pengujian ini dilakukan untuk mengetahui cara penggunaan program, kecepatan program dan seberapa besar kemampuan program untuk melakukan sinkronisasi. Serta menganalisa kerja program dalam melakukan sinkronisasi. Pengujian program ini dilakukan dengan menggunakan dua host dimana kedua host akan menjalankan sinkronisasi. Selain melakukan sinkronisasi, satu dari dua host tersebut akan bertindak sebagai user yang menjalankan *SyncAgent*.

SyncAgent

GAMBAR 4. Create Aglets

GAMBAR 5. Tampilan Tahiti Server

Pertama yang harus dilakukan untuk menjalankan SyncAgent adalah menjalankan Aglets 2.02 di kedua host. Tampilan utama dari *Aglets* adalah Tahiti Server. Tahiti Server ditunjukkan pada Gambar 5. Kemudian membuat (create) *SyncAgent* dengan cara klik tombol "Create" pada Tahiti. Kemudian memilih *SyncAgent* pada window Create Agent. Lalu klik tombol "Create". Gambar 4 menunjukkan tampilan window Create Agent.



***GAMBAR 6**. GUI SyncAgent*

Setelah SyncAgent sudah dibuat maka akan muncul tampilan yang ditunjukkan pada Gambar 6. Informasi dari host pertama diisikan pada form MAIN SERVER, sedangkan informasi dari host kedua pada form BACKUP SERVER. Form pada AutoIndex Directory, Base Directory dan Backup Server Log diisi sama dengan pengaturan pada AutoIndex PHP Script apabila Use AutoIndex diaktifkan. Setelah semua form diisi, klik tombol "Run Sync Agent" untuk menjalankan SyncAgent dan klik tombol "Stop" untuk menghentikan kerja SyncAgent. Selanjutnya masuk ke halaman web dari AutoIndex PHP Script. Kemudian login menggunakan akun Admin. Pada menu admin klik "Reconfigure Script" kemudian isi form pada "base_dir" dan "backup_log". "base_dir" menunjukkan letak direktori dasar yang akan ditampilkan sedangkan "backup_log" untuk menampilkan kolom *Backup Information*.

Index_of/File/

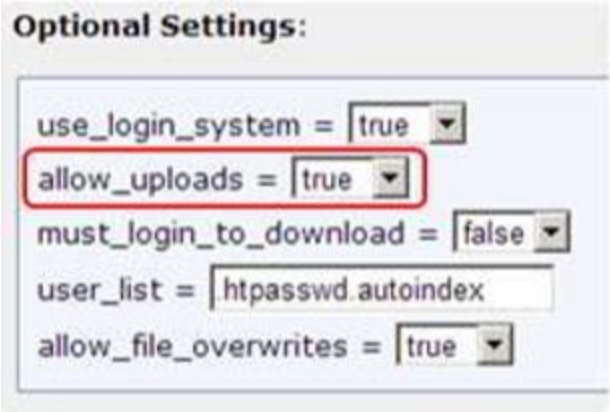
File	Download	Size	Modified	Back Up Files
[BAGAS31] KMSpico 10.2.0 Final Activator.zip	1	8.9 MB	2017-Jan-02	2017-Jan-02
ASUS_T00I_WW_4_3_5_UpdateLauncher.zip	2	7.6 MB	2017-Jan-02	2017-Jan-02
ACS6MCAP-CZv6.03 - SSSL.rar	1	5.6 MB	2017-Jan-04	2017-Jan-04
Autodesk 3ds Max 2012 English Win32-64bit	2	4.9 MB	2017-Jan-05	2017-Jan-05
adb-setup-1.4.3(afandindo.blogspot.com).rar	2	4.5 MB	2017-Jan-07	2017-Jan-07
python-3.5.2.exe	1	4.4 MB	2017-Jan-07	2017-Jan-07
NI Multisim & Ultiboard (Circuit Design Suite) v13.0.iso	1	3.4 MB	2017-Jan-08	2017-Jan-08
cz_ost_complete.zip	1	8.5 MB	2017-Jan-09	2017-Jan-09
Input_Driver_FW20D_WN_8.1206.101.112_A00.EXE	4	14.2 MB	2017-Jan-10	2017-Jan-10
[BAGAS31] Microsoft Visual C++ Pack.rar	4	15.9 MB	2017-Jan-12	2017-Jan-12
studio-install-windows.mp4	4	800 KB	2017-Jan-12	2017-Jan-12
Total	**Total : 23**	**Total Sze : 78.7 MB**		

GAMBAR 7. Tampilan Backup Information

Setiap informasi yang ditampilkan pada kolom Backup Information berupa tulisan last modified file dengan ketentuan sebagai berikut :

1. Berwarna abu-abu bertuliskan "not synced", artinya file belum pernah disinkronkan.
2. Berwarna hijau, artinya file sudah disinkronkan.
3. Berwarna merah, artinya file lebih baru (update) belum disinkronkan.

Penambahan, Pembaruan dan Penghapusan File



GAMBAR 8. Pengaturan pengunggahan bagi user

Ada dua cara untuk menambahkan *file* dan memperbarui *file* menggunakan SyncAgent. Pertama, salin dan tempel (*copy paste*) *file* ke direktori target. Ini bisa dilakukan secara langsung atau melalui *remote*. Kedua, mengunggah *file* melalui form *upload* pada halaman Admin. Cara kedua merupakan cara yang lebih mudah, karena *user* tidak perlu masuk ke dalam sistem hanya perlu mengakses melalui web. *User* yang melakukan pengunggahan tidak harus menggunakan akun Admin karena Autoindex PHP Script memberikan pengaturan pengunggahan selain Admin. Gambar menunjukkan letak pengaturan pengunggahan bagi *user*. Kemudian cara untuk menghapus *file* hanya dapat dilakukan secara langsung maupun melalui *remote.*

Pengujian Sinkronisasi

Pengujian dilakukan dengan memberikan variasi ukuran file dan jumlah file. Koneksi antara kedua host menggunakan kabel UTP cross secara point-to-point. Bertujuan untuk mengetahui waktu yang di butuhkan *SyncAgent* untuk menjalankan sinkronisasi dalam satu kali siklus. Kondisi dan kebutuhan pengujian sinkronisasi dapat dilihat pada Tabel 2.

TABEL 2. Kondisi dan Kebutuhan Pengujian

Host	Server	Alamat IP	Spesifikasi Host	
			Perangkat Keras	Perangkat Lunak
Host 1 (User)	Main Server	10.31.17.23	• Intel Core i3 • Harddisk 300 GB • RAM 4 GB • Realtek PCI, GBE Family Controller	• Windows 7 • Java 7 • Aglets 2.02
Host 2	Backup Server	10.31.36.31	• AMD C-50 1 GHZ • Harddisk 500 GB • RAM 2 GB • Atheron ARS 152/81.58 PCI-E Fast Ethernet	• Windows 7 • Java 7 • Aglets 2.02

Pengujian dengan Variasi Ukuran File

TABEL 3. Pengujian Sinkronisasi dengan variasi besar file.

No.	Besar File (MB)	Waktu Sinkronisasi WatcherAgent (ms)		Waktu Rata –Rata Sinkronisasi (ms)	Kecepatan Rata – Rata (MB/s)
		1	2		
1	5,2	2350	2396	2373	2,541089
2	19,6	5604	5688	5672	4,372845
3	30	7868	7878	7890	4,396745
4	42,3	8891	8876	8890	4,076897
5	50,5	1234	1255	1298	4,291181
6	56	13218	13452	13776	3,890011
7	72,2	15457	15099	15432	4,679120
8	75	15533	15132	15554	3,488901
9	99,7	19902	19456	19789	4,690122
10	101	22345	22333	22378	4,234442
11	115	24889	24987	24211,8	4,543251
12	126	25508	25332	25890	4,334521
13	134	29978	29544	29876	4,176123
14	150	32099	32876	32778	4,231333

Pengujian ini dilakukan dengan cara memberikan file baru secara berkala dengan kelipatan 5,2 MB. Kemudian mencatat waktu yang dibutuhkan program untuk menyelesaikan sinkronisasi pada kedua *WatcherAgent*. Hasil pengujian dapat dilihat pada Tabel 3.

Berdasarkan Tabel 3 dapat dilihat bahwa semakin besar *file* maka semakin besar pula waktu yang dibutuhkan untuk melakukan sinkronisasi. Selain itu didapatkan kecepatan sinkronisasi *file* dengan kecepatan minimum sekitar 2,54 MB/s ,kecepatan maksimum sekitar 4,69 MB/s dan kecepatan rata-rata sekitar 4,33 MB/s. Kecepatan sinkronisasi yang tidak tetap tersebut terjadi karena aplikasi dibuat menggunakan koneksi stream menggunakan protokol TCP. Selama proses pengiriman *file* dimungkinkan terjadi paket data hilang, rusak ataupun kesalahan pengiriman sehingga paket data yang hilang tersebut perlu dikirimkan ulang. Ini merupakan karakteristik dari protokol TCP. Selain itu waktu rata-rata yang hampir sama dapat dilihat pada data nomor 7 dan 8, walaupun kecepatan rata rata transfer data tidaklah sama. Hal ini disebabkan karena sinkronisasi dilakukan pada jaringan Internet aktif, sehingga waktu yang dibutuhkan untuk melakukan sinkronisasi dimungkinkan bertambah atau berkurang sesuai arus data yang melewati jaringan.

Pengujian dengan Variasi Jumlah File

TABEL 4. Pengujian Sinkronisasi dengan variasi jumlah file.

No.	Jumlah File (1 file = 5,2 MB)	Waktu Sinkronisasi WatcherAgent (ms)		Waktu Rata –Rata Sinkronisasi (ms)	Kecepatan Rata – Rata (MB/s)
		1	2		
1	1	1350	1367	1369	2,541089
2	2	2567	2546	2534	4,372845
3	3	3455	3425	3477	4,396745
4	4	4567	4599	4578	5,076897
5	5	6778	6755	6758	4,291181
6	6	7455	7456	7432	6,890011
7	7	8767	8777	8798	8,679120
8	8	8098	8066	8023	7,488901
9	9	8976	8958	8934	7,690122
10	10	1265	1277	1234	9,234442
11	11	1345	1344	1324	8,543251
12	12	1478	1453	1467	9,334521

Pengujian ini dilakukan dengan cara memberikan *file* baru dengan jumlah tertentu dengan ukuran yang sama yaitu 5,2 MB. Kemudian mencatat waktu yang dibutuhkan program untuk menyelesaikan sinkronisasi pada kedua *WatcherAgent*. Berdasarkan Tabel 4 dapat dilihat bahwa semakin banyak jumlah file (dengan ukuran masing-masing sama) semakin besar pula waktu yang dibutuhkan untuk melakukan sinkronisasi. Ini terbukti dari nilai waktu rata-rata sinkronisasi yang cenderung bertambah dari 2548 ms sampai 1478 ms. Namun dari hasil perhitungan kecepatan sinkronisasi didapatkan semakin banyak jumlah *file* maka kecepatan sinkronisasi cenderung naik. Ini dikarenakan proses sinkronisasi dijalankan secara *multithreading*. Setiap *file* disinkronkan oleh satu thread. Semakin banyak *file* semakin banyak *thread* dibuat dan *thread-thread* tersebut dieksekusi secara bersamaan. Meski demikian, penurunan kecepatan sinkronisasi terjadi pada data nomor 5, 8 dan 9. Ini dikarenakan lalu lintas data pada jaringan mempengaruhi *latency* pada koneksi stream masing-masing *thread*. *Latency* merupakan waktu yang dibutuhkan untuk menjalin koneksi antara kedua *host*. Apabila arus data pada jaringan meningkat maka semakin lama waktu yang dibutuhkan untuk menjalin koneksi antara kedua *host* dan ini berakibat pula pada waktu ratarata sinkronisasi yang semakin lama.

Pengujian dengan Variasi Ukuran dan Jumlah File

TABEL 5. Pengujian Sinkronisasi dengan variasi ukuran dan jumlah file..

No.	Jumlah File	Ukuran File (MB)	Waktu Sinkronisasi WatcherAgent (ms)		Waktu Rata – Rata Sinkronisasi (ms)	Kecepatan Rata – Rata (MB/s)	Keterangan
			1	2			
1	100	222	29087	29542	29590	6,541089	Berhasil
2	200	435	53452	53342	53456	7,372845	Berhasil
3	300	667	64431	64433	67689	8,396745	Berhasil
4	400	978	89765	89092	83457	8,076897	Berhasil
5	500	1146	111233	111453	111233,2	8,291181	Berhasil
6	600	13345,70	152345	152413	153451	7,890011	Berhasil
7	700	17889,72	178921	173312	173907	7,679120	Berhasil
8	800	19001,88	189763	183421	183566	8,488901	Berhasil
9	900	20987,90	257778	257432	258970	8,690122	Berhasil
10	1000	22345,81	278342	274258	273456	8,234442	Berhasil
11	1100	25637,88	312631	317008	316095	8,543251	Berhasil
12	1200	28990,61	339290	3392934	339298,6	8,334521	Berhasil

KESIMPULAN

Pengujian ini dilakukan dengan cara memberikan file baru dengan jumlah tertentu dengan ukuran yang berbeda-beda. Ukuran tersebut dibuat secara acak menggunakan program dengan range antara 1 byte sampai dengan 20 MB. Kemudian mencatat waktu yang dibutuhkan program untuk menyelesaikan sinkronisasi pada kedua *WatcherAgent*. Berdasarkan pengujian pada Tabel 5 dapat dijelaskan sebagai berikut. Selain itu didapatkan kecepatan yang bervariasi pula dengan kecepatan minimum sekitar 7,37 MB/s, kecepatan maksimum sekitar 8,69 MB/s dan kecepatan rata-rata sinkronisasi file adalah 8,32 MB/s. Namun pada data nomor 2, 6 dan 7 terjadi penurunan sekitar 1 MB/s terhadap data lainnya. Hal ini dikarenakan lalu lintas data pada jaringan yang terkadang tinggi memperlambat komunikasi yang dibuat secara *multithreading. Thread* maksimal yang dibuat pada aplikasi ini jumlahnya adalah 25. Setiap mencapai 25 *thread* maka proses sinkronisasi berhenti pada selang waktu tertentu untuk menunggu semua *trhead* selesai berjalan. Kecepatan rata – rata saat melakukan sinkronisasi *file* adalah 8,32 MB/s. Auto Index PHP Script dapat menampilkan informasi sinkronisasi *file* pada kolom Backup Informastion, namun karena tidak ada (fitur) batasan jumlah *file* yang ditampilkan pada satu halaman, Auto Index PHP Script hanya dapat menampilkan 1400 *file* dan atau folder dalam satu halaman. Sinkronisasi *file* akan lebih bermanfaat apabila ditambahkan fitur pelaporan sinkronisasi via email ataupun SMS *gateway*. Agar kerja Aglets lebih maksimal dan keamanannya lebih baik lagi, karena banyak nya *user* yang tidak menyimpan *file* dalam bentuk yang hanya memakai satu server maupun dua server saja.

UCAPAN TERIMAKASIH

Terimakasih banyak kepada pembimbing saya, Wisnu Broto, ST.MT yang telah membantu selama membuat serta mengerjakan makalah full paper dan kepada Prodi Elektro Fakultas Teknik Universitas Pancasila, karena telah memberikan kesempatan untuk mengikuti Seminar Nasional Fisika UNJ 2017 dan menjadi pemakalah serta membimbing untuk perisapan dalam Seminar Nasional Fisika 2017.

REFERENSI

[1] Yudianto, Scifo Anggi, Somantri, M., dan Isnanto, R. Rizal, "Perancangan dan Pembuatan Perangkat Lunak Berbasis Mobile Agent untuk Pencarian Buku", Universitas Diponegoro, Semarang, 2011.

[2] Available: http://ojs.unud.ac.id/index.php/lontar/article/view/23072 [Online]

[3] Available: http://mydjiwandono.blogspot.co.id/2013/01/jurnal-aplikasi-mobile-agent-untuk.html [Online]

[4] S. Adang, M. Adityo, B. Julius, "SISTEM MOBILE AGENT DALAM KOMPUTASI GRID" *Proceeding, Seminar Ilmiah Nasional Komputer dan Sistem Intelijen. (KOMMIT) ISSN* : *1411-6286*, 2008.

[5] M. Naylor. "The Use of Mobile Agents in Network Management", *MSc Information Technology (Systems Integration),* 2000.

[6] P. Dasgupta, N. Narasimhan, L.E. Moser and P.M. Melliar-Smith, "MAgNET: Mobile Agents for Networked Electronic Trading", *IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING*, Vol. 11, No. 4, 1999.

[7] Purbasari, Intan Yuniar and, Joko Lianto, "Rancang Bangun Perangkat Lunak Agent untuk Polling Halaman Web pada Selancar Web dalam Komunitas Menggunakan Aglet". *Jurnal Teknologi Informasi dan Komunikasi*, vol. 2 no. 1, pp. 15-21. ISSN 1978-0087, 2006.