

Pertukaran Data Antar Database dengan Menggunakan Teknologi API

Ahmad Hanafi¹, I Made Sukarsa², A.A. Ketut Agung Cahyawan Wiranatha³

Jurusan Teknologi Informasi, Universitas Udayana
Jl. Kampus Bukit Jimbaran, Bali-Indonesia

1ahmadhanafi1207@gmail.com

2sukarsa@gmail.com

3agung.cahyawan@gmail.com

Abstrak

Pertukaran data secara elektronik antar instansi atau perusahaan harus didukung dengan media penyimpanan data yang sesuai kapasitas. Database MySQL merupakan suatu engine yang digunakan untuk melakukan penyimpanan data berupa informasi, dimana data tersebut dapat dimanfaatkan sesuai kebutuhan. MYSQL memiliki kelebihan diantaranya adalah memberikan kemudahan dalam hal pengaksesan dan dapat bekerja di berbagai platform. Kebutuhan sistem yang harus handal dan multitasking mampu menjadikan database tidak hanya sebagai media penyimpanan data, tetapi juga dapat dimanfaatkan sebagai sarana pertukaran data. Dropbox API merupakan solusi terbaik yang dapat dimanfaatkan sebagai teknologi yang mendukung database untuk mampu melakukan pertukaran data. Kombinasi antara Dropbox API dan database dapat dijadikan solusi yang sangat murah untuk perusahaan kecil untuk menerapkan pertukaran data, karena hanya membutuhkan koneksi Internet yang relatif kecil. Pertukaran data melalui database dengan menggunakan Teknologi API Dropbox telah berhasil dilakukan, mekanisme pertukaran data yang dilakukan yaitu melewati Dropbox sebagai perantara dan meneruskan ke tujuan pengiriman dengan memanfaatkan identifier email dari pengguna Dropbox, sehingga pesan disinkronkan ke dalam database penerima.

Kata kunci: *Pertukaran Data Elektronik, Database, MySQL, Dropbox API, Internet*

Abstract

Electronically data interchange between institutions or companies must be supported with appropriate data storage media capacity. MySQL is a database engine that is used to perform data storage in the form of information, where the data can be utilized as needed. MYSQL has the advantage of which is to provide convenience in terms of usage, and able to work on different platforms. System requirements that must be reliable and multitasking capable of making the database not only as a data storage medium, but can also be utilized as a means of data exchange. Dropbox API is the best solution that can be utilized as a technology that supports the database to be able to Exchange data. The combination of the Dropbox API and database can be used as a very cheap solution for small companies to implement data exchange, because it only requires a relatively small Internet connection.

Key words: *Eelctronic Data Interchange, Database, MySQL, Dropbox API, Internet*

1. Pendahuluan

Pertukaran data dalam sebuah perusahaan sangat dibutuhkan untuk menyalurkan segala dokumen bisnis ke pihak lain. Pertukaran data secara elektronik seharusnya bersifat *lest investment*, dimana pelaku bisnis tidak perlu membeli peralatan baru sebagai infrastruktur untuk melakukan pertukaran data, dengan kata lain tetap menggunakan peralatan yang sudah tersedia.

Database merupakan suatu media penyimpanan data yang dapat dimanfaatkan sebagai sarana untuk mengkomunikasikan dokumen bisnis tanpa harus mengeluarkan investasi yang cukup besar dalam menerapkan teknologi pertukaran data. MySQL merupakan pemilihan *database engine* yang tepat, karena kelebihanannya dapat bekerja di berbagai *platform* dan mudah dalam hal pengaksesan.

Menjalankan peran *database* dalam melakukan pertukaran data harus didukung oleh teknologi khusus. Solusi yang terbaik adalah dengan menggunakan teknologi API Dropbox. Dokumentasi yang sederhana serta penggunaan yang lebih mudah dapat dijadikan alasan bahwa API Dropbox merupakan teknologi yang sangat bermanfaat untuk menerapkan konsep pertukaran data kepada perusahaan-perusahaan kecil karena hanya membutuhkan koneksi Internet yang relatif kecil.

Pembuatan teknologi pertukaran data telah banyak dilakukan oleh penelitian-penelitian terdahulu. Penelitian-penelitian tersebut memiliki keterkaitan dengan pembuatan penelitian ini, dan beberapa metode ataupun mekanisme kerja dari penelitian terdahulu diterapkan ke dalam penelitian ini.

Penelitian dengan berjudul *Data Exchange between Information System at Low Bandwidth Quality using Messaging* [1]. Penelitian ini memfokuskan untuk melakukan penyelarasan data yang baik dan transparan walaupun data tersebut tersebar diberbagai daerah. Penelitian dengan judul Perancangan *Cloud Storage* dengan Konsep *Auto Syncing* menggunakan Aplikasi *Owncloud* dan Dropbox [2]. Penelitian ini membahas mengenai penerapan aplikasi *cloud computing* melalui terminologi SaaS (*Software as a Service*), yaitu Dropbox yang disinkronkan dengan *owncloud*. Keuntungan yang diberikan yaitu dapat melakukan *back-up* data pada kedua penyedia Layanan *Cloud Computing* hanya dengan menggunakan salah satunya saja. Melakukan *upload file* pada aplikasi Dropbox secara otomatis, *file* yang di *upload* di Dropbox tersedia juga pada aplikasi *owncloud*. Penelitian dengan judul *Development of a Message Oriented Middleware for a Heterogeneous Distributed Database Systems* [3]. Penelitian ini menjelaskan tentang peran *Middleware* adalah untuk memudahkan tugas, merancang pemrograman, dan mengelola aplikasi *database* terdistribusi dengan menyediakan lingkungan pemrograman sederhana, konsisten, dan terintegrasi.

Berdasarkan permasalahan diatas munculah sebuah ide atau motivasi untuk membuat suatu teknologi pertukaran data antar *database* dengan memanfaatkan teknolog API Dropbox, dengan tujuan dapat di terapkan pada perusahaan-perusahaan kecil dengan biaya investasi yang sangat rendah.

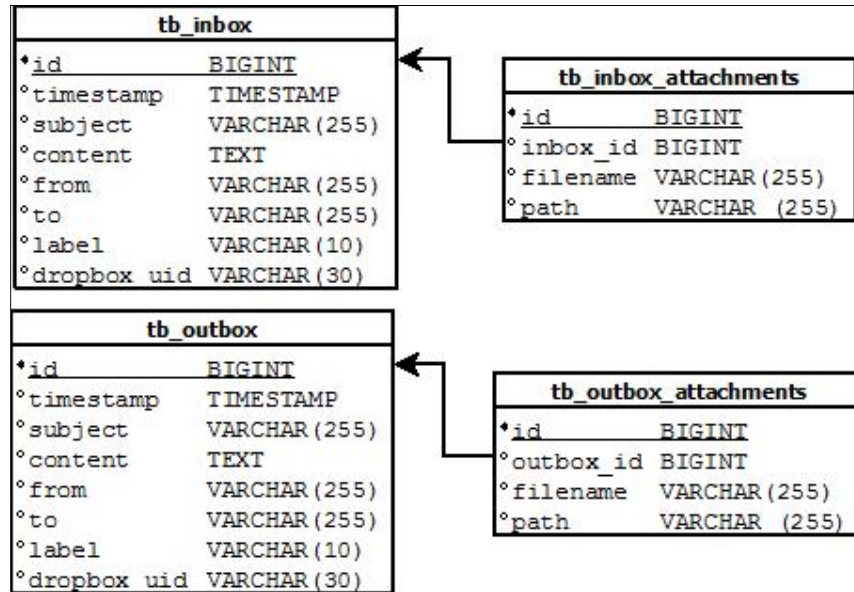
2. Metodologi Penelitian

Metodologi penelitian yang dibuat dalam pembuatan aplikasi ini adalah menggunakan konsep pengembangan sistem yang termasuk ke dalam SDLC, yaitu Metode Waterfall. Metode Waterfall mampu membuat proses pengembangan sistem menjadi lebih terstruktur. Alur analisis dari aplikasi yang dibuat adalah sebagai berikut:

- a. Mendefinisian permasalahan terkait aplikasi yang dirancang.
- b. Mengumpulkan data terkait dengan penyusunan dan pembuatan aplikasi pertukaran data, dengan studi literatur.
- c. Mengerti dan menguasai teori pendukung untuk menunjang dalam pembuatan sistem sehingga dapat dibuat suatu pemodelan sistem.
- d. Perancangan dan pembuatan *database* dengan menggunakan MySQL.
- e. Pengembangan pertukaran data melalui *database*.
- f. Pengujian sistem yang dilakukan secara berulang sampai diperoleh hasil yang sesuai.
- g. Implementasi sistem.
- h. Pengambilan kesimpulan.

2.1 Skema Database

Implementasi pembangunan *database* sebagai pertukaran data menggunakan MySQL sebagai DBMS.



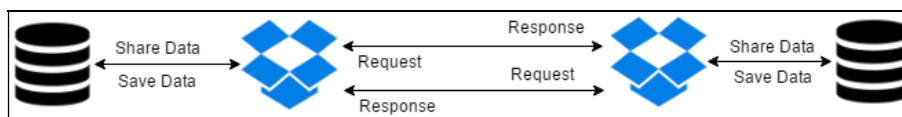
Gambar 1. Skema Database

Gambar 1 menunjukkan skema *database* dari sistem pertukaran data antar *database*. Gambar 1 menunjukkan tabel-tabel pendukung yang memiliki peran dan fungsinya masing-masing dalam melakukan pertukaran data. Tabel-tabel tersebut dapat dijelaskan sebagai berikut:

- Tabel *tb_inbox* berfungsi untuk menampung segala pesan yang masuk ke dalam *database*.
- Tabel *tb_outbox* berfungsi untuk menampung segala pesan yang dikirimkan (pesan keluar).
- Tabel *tb_inbox_attachment* berfungsi untuk menampung segala *file attachment* yang masuk.
- Tabel *tb_outbox_attachment* berfungsi untuk menampung segala *file attachment* yang dikirimkan.

2.2 Arsitektur Sistem

Arsitektur merupakan istilah untuk menyatakan bagaimana cara mendefinisikan komponen-komponen yang lebih spesifik secara terstruktur, dengan tujuan agar struktur yang dirancang dapat menjawab kebutuhan saat ini dan nanti. Gambar 2 menunjukkan arsitektur sistem dari Pertukaran Data Antar Database dengan menggunakan Teknologi API.



Gambar 2. Arsitektur Sistem

Gambar 2 menjelaskan model pertukaran data yang dibuat. Pertukaran data yang dilakukan di mulai dengan proses pengaturan sistem oleh admin. Proses pengaturan ini berupa proses pemilihan *database* sebagai sumber data, proses pengaturan dan pengaktifan Dropbox sebagai media pertukaran data, serta proses penentuan format data yang disimpan untuk dipertukarkan melalui API Dropbox. Layanan yang dibuat menggunakan Dropbox sebagai perantara untuk dapat saling bertukar data antar *database*. Format data yang dipertukarkan harus sesuai dengan format pesan yang telah ditentukan oleh penyedia layanan. Pertukaran data yang dilakukan antar satu *cabang* ke cabang lainnya, dimana setiap cabang memiliki *database* yang

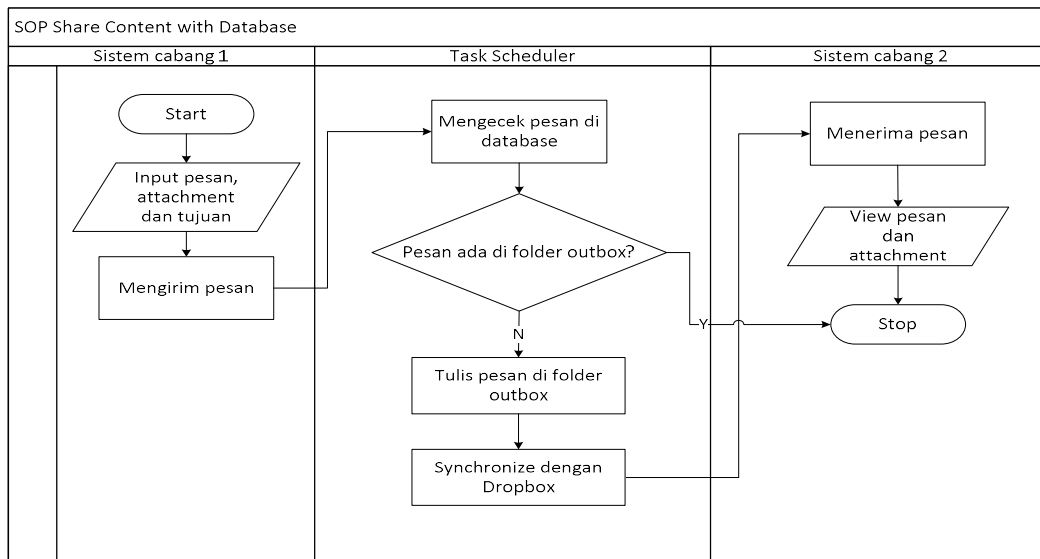
digunakan untuk menampung data dan media pertukaran data. Proses pertukaran data di masukan melalui *database*, kemudian diterima Dropbox terlebih dahulu, dan diteruskan ke cabang tujuan lainnya.

Proses pertukaran data yang dilakukan antar *database* melibatkan salah satu pelaku yang bernama *scheduler*. *Scheduler* merupakan suatu mesin yang dibuat khusus untuk mengeksekusi segala perintah, baik pesan yang masuk maupun pesan yang dikirimkan. Mekanisme pertukaran data melalui *database* dijelaskan pada Gambar 3 berupa SOP (*Standard Operational Procedure*). Pelaku aktivitas yang dilakukan dapat dilihat pada Tabel 1.

Tabel 1. Daftar Aktivitas dalam Pertukaran Data antar *Database*

Pelaku	Aktivitas
Sistem cabang 1	a. Memasukan pesan, <i>attachment</i> dan tujuan.
<i>Scheduler</i>	b. Melakukan proses pengiriman pesan.
	c. Mengecek konten yang terdapat di dalam <i>database</i> .
	d. Melakukan pengecekan apakah konten ada di <i>folder outbox</i> .
	e. Jika tidak ada maka dilanjutkan dengan proses penulisan pesan di <i>folder outbox</i> . Jika tidak proses selesai.
Sistem cabang 2	f. Melakukan proses sinkronisasi dengan <i>database</i> .
	g. Melakukan proses penerimaan konten.
	h. Melihat detail isi konten.

Aktivitas yang dilakukan berdasarkan Tabel 1 dimulai dari sistem cabang 1 yang melakukan pengiriman data yang bersumber dari *database*. *Scheduler* mengeksekusi perintah hingga konten tersebut sampai pada sistem cabang 3.



Gambar 3. SOP Pertukaran Pesan melalui *Database*

Gambar 3 menunjukkan bentuk *Standard Operational Procedure* dari proses pertukaran data melalui *database*. Alurnya dimulai dari proses pemasukan data konten yang dilakukan oleh sistem cabang 1. Proses pengiriman *file* dilakukan dengan memasukan *path* dari *file* terlebih

dahulu. Tugas *scheduler* di sini adalah mengecek data pesan yang ada di *database* dan memastikan jika pesan belum ditulis ke dalam *Folder Outbox* dan menyinkronkan data konten ke dalam Dropbox. Bentuk keberhasilan dari proses ini adalah sampainya pesan ke tujuan dan dapat di *view* secara detail.

3. Kajian Pustaka

3.1 EDI (Electronic Data Interchange)

Electronic Data Interchange atau EDI merupakan suatu metode pertukaran data-data yang terstruktur antar aplikasi komputer, perusahaan atau instansi dengan menggunakan format tertentu yang disetujui untuk keperluan bisnis secara elektronik [4]. Pertukaran data difokuskan untuk aplikasi komputer disebabkan untuk meminimalisir adanya campur tangan manusia dalam menggunakan aplikasi komputer, sedangkan sisanya seperti proses pengiriman dan interpretasi data dapat dilakukan oleh komputer [5].

3.2 Middleware

Middleware adalah suatu teknologi atau *software* yang dirancang untuk membantu mengelola kompleksitas dan heterogenitas yang melekat dalam sistem terdistribusi. Hal ini didefinisikan sebagai lapisan perangkat lunak diatas sistem operasi tetapi di bawah program aplikasi yang menyediakan abstraksi pemrograman umum di sistem terdistribusi. *Middleware frameworks* dirancang untuk menutupi beberapa jenis heterogenitas yang didistribusikan programmer dari sistem. *Middleware frameworks* selalu menutupi heterogenitas jaringan dan *hardware*. Kebanyakan *middleware frameworks* juga menutupi heterogenitas sistem operasi atau bahasa pemrograman, atau keduanya. Beberapa seperti CORBA juga sebagai wadah heterogenitas antara implementasi vendor standar *middleware* yang sama. Hasilnya adalah abstraksi pemrograman ditawarkan oleh *middleware* dapat memberikan transparansi terhadap distribusi dalam satu atau lebih dari dimensi, seperti lokasi, konkurensi, replikasi, kegagalan, dan mobilitas [6].

3.3 API (Application Programming Interface)

Application Programming Interface (API) adalah sebuah teknologi yang memfasilitasi pertukaran informasi atau data antara dua atau lebih aplikasi perangkat lunak. API adalah antarmuka *virtual* antara dua fungsi perangkat lunak yang saling bekerja sama, seperti antara sebuah *word processor* dan sebuah *spreadsheet* [7]. Sebuah API mendefinisikan bagaimana cara programmer memanfaatkan suatu fitur tertentu dari sebuah komputer. API tersedia untuk sistem *windowing*, sistem *file*, sistem *database*, serta sistem jaringan.

3.5 Dropbox

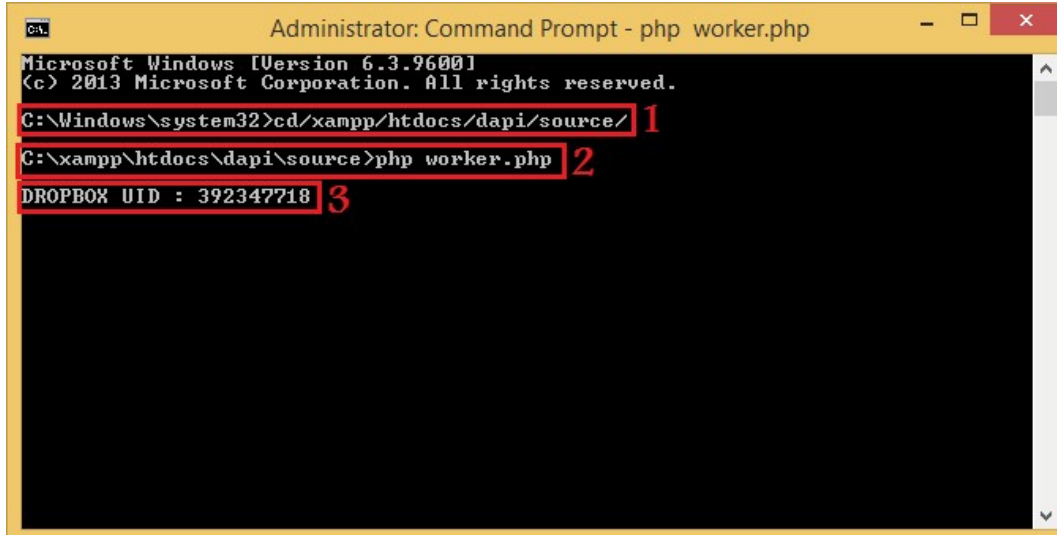
Dropbox adalah layanan penyedia data berbasis *web* yang dioperasikan oleh Dropbox Inc. Dropbox menggunakan sistem penyimpanan berjaringan untuk menyimpan dan berbagi serta berkas dengan *client* lain di Internet menggunakan sinkronisasi data. Dropbox menyediakan dokumentasi API yang mempermudah para developer untuk mengembangkan aplikasi serupa [8].

4. Hasil dan Pembahasan

Hasil dari sistem dapat dilakukan dengan melakukan ujicoba dari layanan pertukaran data dengan menggunakan API Dropbox. Pembahasan dilakukan dengan mengetahui komponen apa saja yang digunakan dalam membangun sistem.

4.1 Instalasi Scheduler

Scheduler memiliki tugas untuk mengatur lalu lintas terhadap pesan yang masuk maupun pesan yang keluar. Ketentuan yang berlaku di dalam sistem adalah *scheduler* harus selalu aktif ketika melakukan transaksi dari *database*. *Scheduler* yang dibuat berupa *file php* (*worker.php*) yang memiliki tugas untuk menjalankan segala fungsi yang terdapat pada *file engine* yang berisi fungsi API Dropbox.



Gambar 4. Menjalankan Scheduler

Gambar 4 menunjukkan cara menjalankan atau mengaktifkan *scheduler*. Poin-poin pada Gambar 4 menunjukkan tahapan untuk melakukan pengaktifkan scheduler yang dapat dipaparkan sebagai berikut:

- a. Tahapan pertama yang dilakukan adalah masuk ke direktori tempat dimana *file scheduler* berada.
- b. Tahapan kedua yang dilakukan adalah memanggil *file scheduler*, dengan mengetikkan perintah `php worker.php`
- c. Keberhasilan mengaktifkan *scheduler* dapat dilihat munculnya Dropbox User ID, yang menandakan *user* yang sedang aktif menggunakan sistem.

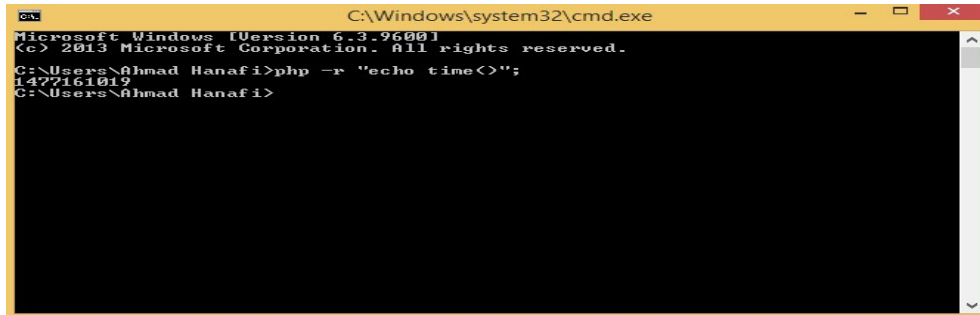
4.2. Pengujian sistem

Pengujian pengiriman data dapat dilakukan di dalam tabel `tb_outbox` yang terdapat di dalam *database*. *Identifier* yang digunakan adalah *email* dari pengguna Dropbox.

id	timestamp	subject	content	from	to	dropbox_uid	status
1	1477161019	test pesan db	te... 38B	ah3128582@gmail.	ah44816@gmail.	392347718	0
(Auto)	(NULL)	(NULL)	(N... OK	(NULL)	(NULL)	(NULL)	0

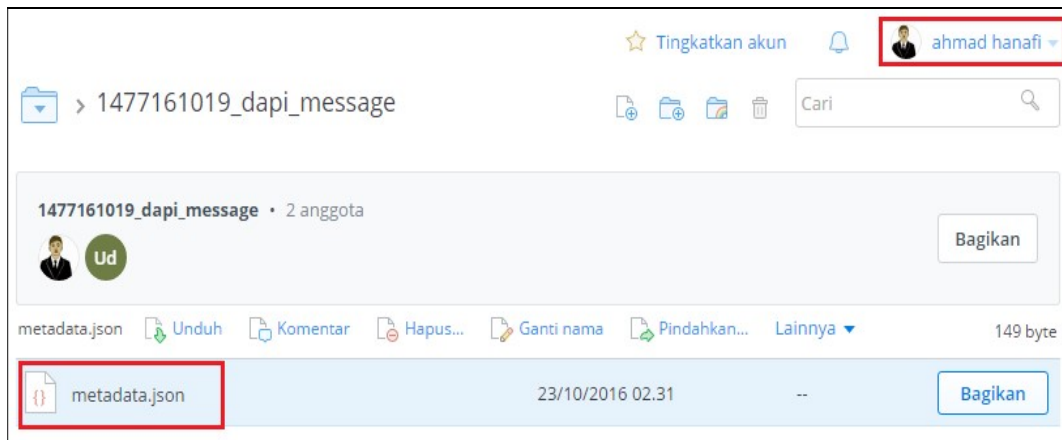
Gambar 5. Menulis Pesan ke dalam Tabel `tb_outbox`

Gambar 5 menunjukkan proses penulisan pesan yang dilakukan pada tabel `tb_outbox`. Data-data yang diisi harus sesuai dengan *filed* yang ada. Khusus untuk pengisian *filed* `dropbox_uid`, pengguna diharuskan untuk *login* terlebih dahulu ke dalam Dropbox, sehingga sistem dapat memiliki akses untuk mengambil data Dropbox *user* ID. Waktu pengiriman didefinisikan dalam bentuk *timestamp*, yang merupakan fungsi PHP yang sudah ada.



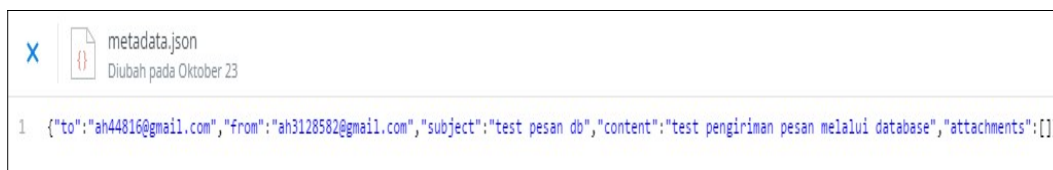
Gambar 6. Mendapatkan Timestamp

Gambar 6 menunjukkan cara yang dilakukan untuk menentukan waktu (Timestamp) pengiriman pesan. Pengisian timestamp untuk penentuan waktu pengiriman pesan saat itu juga, dapat dilakukan dengan mengetikkan perintah `php -r "echo time()";`



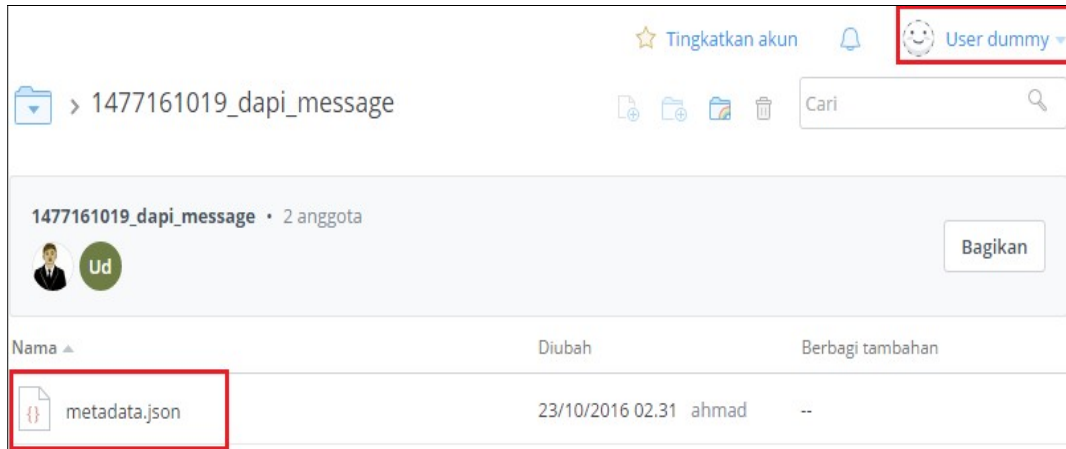
Gambar 7. Pesan masuk ke Dropbox Pengirim

Pesan yang dikirimkan awalnya masuk ke dalam Dropbox pengirim terlebih dahulu, sebelum diteruskan ke tujuan. Gambar 7 menunjukkan pesan yang dikirimkan telah di unggah di dalam Dropbox pengirim. Pesan yang dikirimkan di tampung dalam format JSON (`metadata.json`).



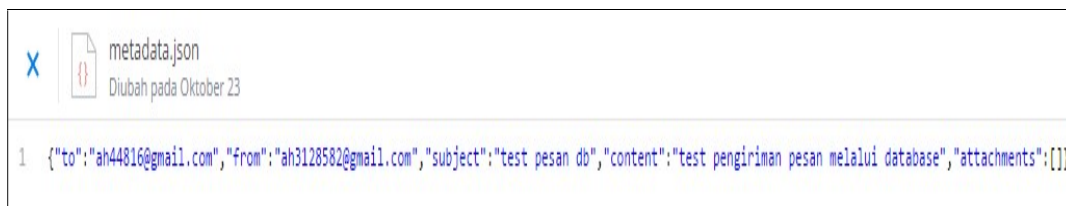
Gambar 8. Isi Pesan Pengirim

Gambar 8 menunjukkan isi pesan dalam `metadata.json`, dimana isi pesan ini sesuai dengan yang dimasukkan melalui *database*.



Gambar 9. Pesan masuk ke Dropbox Penerima

Gambar 9 menunjukkan pesan yang dikirimkan telah masuk ke Dropbox Penerima. Pesan yang diterima berupa *file* JSON (*metadata.json*).



Gambar 10. Isi Pesan Masuk

Gambar 9 menunjukkan isi pesan yang masuk penerima. Isi pesan disesuaikan dengan struktur format JSON.

id	timestamp	subject	content	from	to	dropbox_uid	status
1	1477161019	test pesan db	test pengiriman...	38B ah3128582@gmail.com	ah44816@gmail.com	519188918	0
(Auto)	(NULL)	(NULL)	(NULL)	OK (NULL)	(NULL)	(NULL)	0

Gambar 11. Pesan masuk ke *Database* *tb_inbox* Penerima

Pesan yang sudah masuk ke dalam Dropbox penerima, kemudian *scheduler* melakukan sinkronisasi, sehingga pesan tersebut disimpan dan dapat di *view* di *database* penerima. Gambar 11 menunjukkan bahwa pesan yang dikirimkan oleh *user* "ahmad hanafi" telah masuk ke dalam *database* tabel *tb_inbox* *user* "user_dummy".

5. Kesimpulan

Pertukaran data yang dilakukan dengan menggunakan *database*, terbukti membantu pelaku bisnis, karena tidak perlu membeli peralatan baru sebagai infrastruktur dan hanya menggunakan peralatan yang sudah tersedia. Menggunakan Teknologi API mempermudah pembangunan sistem tanpa harus mengeluarkan biaya dalam pembangunan *server* dan tidak perlu melakukan akses IP *Public*.

Hasil pengujian sistem dalam melakukan pertukaran data melalui *database* dengan menggunakan Teknologi API Dropbox telah berhasil dilakukan. Mekanisme pertukaran data yang dilakukan melewati Dropbox sebagai perantara dan meneruskan ke tujuan pengiriman dengan memanfaatkan *identifier email* dari pengguna Dropbox, sehingga pesan tersebut

disingkrankan ke dalam *database* penerima. Kombinasi *database* dengan menerapkan Teknologi API Dropbox terbukti menjadi solusi yang tepat, handal dan murah untuk menerapkan konteks pertukaran data kepada perusahaan-perusahaan kecil yang memiliki kualitas Internet terbatas.

Daftar Pustaka

- [1] I. M. Sukarsa, N. W. Wiswani, and P. Wirabuana, "Data Exchange between Information System at Low Bandwidth Quality using Messaging," vol. 60, no. 2, pp. 417–422, 2014.
- [2] J. Cahyadi and U. P. Marteus, "Perancangan Cloud Storage Dengan Konsep Auto Syncing Menggunakan Aplikasi Owncloud Dan Dropbox," pp. 1–11, 2013.
- [3] L. H. Khalid and M. F. Younis, "Development of a Message-Oriented Middleware for a Heterogeneous Distributed Database Systems," vol. 16, no. 4, 2013.
- [4] JTC, "Electronic Data Interchange," no. March, pp. 1–11, 1998.
- [5] L. Mrkonjic, "B2B Series Whitepaper WHAT IS EDI AND HOW DOES IT WORK?," p. 24, 2009.
- [6] D. E. Bakken, "Middleware Introduction," *Washingt. State Univ.*, 2002.
- [7] M. Bray, *Application Programming Interface*. The Software Engineering Interface Institute, 1997.
- [8] J. Ying, "Introducing Dropbox for Teams," 2011. [Online]. Available: <https://blogs.dropbox.com/dropbox/2011/11/introducing-dropbox-for-teams>.