

# **Middleware Data Pembelian ke Aplikasi e-Faktur untuk Meningkatkan Efisien Waktu dan Biaya Pengusaha**

*Purchase Data Middleware to e-Faktur Application to Increase Time and Cost Efficiency for Entrepreneur*

**Tony Wijaya**

STMIK Pontianak, Jl. Merdeka No. 372 Pontianak, Kalimantan Barat

Jurusan Sistem Informasi STMIK Pontianak

E-mail: mail.tonywijaya@gmail.com

## **Abstrak**

Sejak tanggal 1 Juli 2016 setiap pengusaha di Indonesia wajib melaporkan transaksi pembelian melalui aplikasi e-Faktur dari Direktorat Jenderal Pajak. Dengan aplikasi ini, pemerintah dapat mengurangi terjadinya penyalahgunaan faktur pajak. Pengusaha juga merasakan manfaat karena tidak perlu. Namun sisi negatifnya adalah pengusaha harus melakukan dua kali entri data dan menambah biaya karyawan untuk melakukannya. Penulis melihat fakta ini dan berusaha mengembangkan sebuah middleware yang dapat menjembatani sistem berjalan dengan e-Faktur. Tujuannya adalah supaya dapat lebih meningkatkan efisien waktu dan biaya pengusaha dalam menggunakan aplikasi e-Faktur, khususnya pada data pembelian. Pendekatan dalam perancangan menggunakan metode Agile dengan pendekatan Extreme Programming yang lebih mengedepankan tercapainya fitur yang akan dibangun. Pengujian dilakukan dengan metode black box untuk memastikan data hasil ekspor sudah sesuai. Memiliki fitur penyesuaian desimal sehingga tidak akan terjadi selisih nilai pembulatan. Dengan middleware ini, pengusaha tidak perlu melakukan input data ke aplikasi e-Faktur, cukup melakukan ekspor data dari sistem berjalan sehingga meningkatkan efisiensi waktu dan biaya.

**Kata kunci**— Middleware, Pembelian, e-Faktur.

## **Abstract**

*Since July 1<sup>st</sup>, 2016, every entrepreneur in Indonesia must implement e-Faktur application from Indonesia Taxation Authority to report purchasing data. This application can enable entrepreneurs to report the company's purchasing tax either online or offline from their very office and no need to be in tax authority's office. E-Faktur is already saved some time of entrepreneurs. But they have to do data entry twice and cost for employee is increased to use this app. Writer notice this fact and tried to develop a middleware that can integrate running system and eFaktur. The objective is to increase more time efficiency of entrepreneurs when using e-Faktur, especially in purchasing data. This middleware is designed using Agile Methodology with Extreme Programming approach which value working software feature. Tested with black box methodology to ensure exported data is correct. It has feature to adjust decimal so it will be very accurate. With this middleware, entrepreneur do not need to input data into e-Faktur application, but export from running system instead. This means it can increase time and cost efficiency.*

**Keywords**— *Middleware, Purchase, e-Faktur.*

## **1. PENDAHULUAN**

E-Faktur atau Faktur Pajak yang berbentuk elektronik merupakan faktur pajak yang dibuat melalui aplikasi atau sistem elektronik yang ditentukan dan/atau disediakan oleh Direktorat Jenderal Pajak (DJP) [1]. Untuk menerapkan pembuatan e-Faktur ini, DJP telah

menyediakan aplikasi yang dapat diinstall di perangkat komputer Pengusaha Kena Pajak (PKP) dan e-Faktur ini otomatis terhubung ke program e-SPT, sehingga akan memudahkan PKP dalam membuat SPT Masa Pajak Pertambahan Nilai (PPN) secara elektronik menggunakan program e-SPT. Penerapan ini secara nasional dilakukan mulai tanggal 1 Juli 2016.

Pada penelitian yang dilakukan di Filipina, penerapan Information and Communication Technology (ICT) dengan nama aplikasi ETRACS dalam perpajakan terbukti mampu meningkatkan transparansi dan pendapatan pajak secara lokal yang pada akhirnya mampu memberikan keuntungan bagi kedua belah pihak yaitu pemerintah dan pengusaha wajib pajak [2]. Para wajib pajak merasakan keuntungan yaitu penghematan waktu dan biaya karena proses yang efisien dalam menangani validasi dan verifikasi pajak.

CV. Bess merupakan perusahaan dagang yang beroperasi di kabupaten Pemangkat, Kalimantan Barat, Indonesia. Perusahaan ini telah memiliki system berjalan berupa aplikasi Sistem Informasi Dagang (SID) yang digunakan untuk mengelola data penjualan, pembelian, dan stok. Seperti aplikasi ETRACS di Filipina, hadirnya e-Faktur bagi CV. Bess berarti bertambahnya satu aplikasi di lingkungan perusahaan, sehingga mereka perlu melakukan input data sebanyak dua kali. Yang pertama yaitu input data pada aplikasi berjalan, dan yang kedua yaitu input data pada aplikasi e-Faktur. Penelitian ini mencoba menghilangkan redundansi input data tersebut, sehingga pihak CV. Bess hanya perlu menginput data pada aplikasi SID saja. Sedangkan data pada aplikasi e-Faktur akan diinput secara otomatis oleh middleware dengan membaca data SID. Data yang dimaksud dibatasi pada data transaksi pembelian saja.

Middleware adalah sebuah software yang dapat mengatasi heterogenitas dan kompleksitas dalam sistem-sistem yang terdistribusi [3]. Penelitian ini merupakan kelanjutan dari penelitian sebelumnya yang dilakukan pada tahun 2017 [4] yaitu mengimplementasikan middleware untuk menghubungkan aplikasi sistem berjalan pada PT. Cemerlang Andalan Nusantara yang beroperasi di kota Pontianak, Kalimantan Barat, Indonesia. Perusahaan ini sudah memiliki sistem berjalan yang direkayasa oleh programmer lokal. Sama halnya seperti yang telah diuraikan sebelumnya, kehadiran aplikasi e-Faktur membuat pihak perusahaan harus melakukan dua kali input data. Hal ini dikarenakan baik struktur program maupun struktur data kedua aplikasi tidaklah sama. Penelitian ini menghasilkan sebuah middleware yang dapat menjadi penengah (*middle*) antara kedua aplikasi tersebut. Konsep middleware ini yaitu:

1. Membaca data penjualan dari sistem berjalan
2. Mengkonversi struktur datanya menjadi struktur yang dapat diterima oleh e-Faktur
3. Menghasilkan sebuah file *Comma Separated Value* (\*.csv) yang dapat digunakan untuk impor data ke aplikasi e-Faktur.

Dengan adanya middleware ini, pihak PT. Cemerlang Andalan Nusantara tidak perlu melakukan input data penjualan ke aplikasi e-Faktur sehingga dapat menghemat waktu. Dengan merujuk kepada penelitian sebelumnya yang mampu menjembatani data penjualan, penelitian ini menambah fitur middleware tersebut sehingga mampu menjadi penengah bagi data pembelian, khususnya untuk sistem berjalan CV. Bess.

## 2. METODE PENELITIAN

Penelitian ini dilakukan dengan pendekatan kualitatif [5]. Dengan pendekatan ini akan dihasilkan sebuah middleware yang sebelumnya belum ada. Pengumpulan data dilakukan dengan melakukan observasi terhadap:

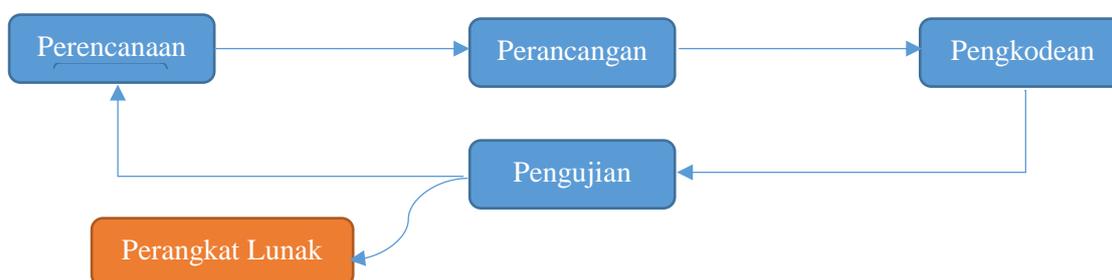
1. Format basis data pada sistem berjalan PT. Cemerlang Andalan Nusantara
2. Data pembelian hasil ekspor dari aplikasi e-Faktur.

Instrumen penelitian adalah peneliti sendiri sehingga peneliti dapat menyesuaikan dengan lingkungan penelitian yaitu sistem berjalan dan aplikasi e-Faktur dari DJP. Kesimpulan dibuat berdasarkan interpretasi data oleh peneliti. Bentuk penelitian adalah penelitian eksperimen dengan objek yaitu sistem berjalan PT. Cemerlang Andalan Nusantara. Peneliti melakukan

percobaan untuk mentransfer data dari sistem berjalan ke aplikasi e-Faktur. Hasilnya adalah sebuah middleware yang mampu menjembatani kedua aplikasi.

Metode perancangan perangkat lunak yang digunakan adalah metode Agile. Metode ini dapat mengelola kebutuhan yang dinamis, memenuhi kepuasan pelanggan, dan menghasilkan produk yang sesuai dengan permintaan pelanggan dan kualitas yang dibutuhkan [6]. Metode ini digunakan dalam penelitian ini karena lebih mengedepankan perangkat lunak yang berfungsi daripada dokumentasi yang lengkap sehingga lebih menghemat waktu pengembangan. Ini merupakan prinsip dasar Agile seperti yang tertuang dalam *Agile Manifesto for Software Development* [7]. Selain itu, metode ini juga dapat merespon dengan cepat terhadap perubahan daripada hanya mengikuti rancangan UML yang dibuat. Hal ini relevan karena penelitian ini dilakukan dengan bentuk eksperimen.

Pendekatan perancangan perangkat lunak yang digunakan adalah Extreme Programming. Extreme Programming merupakan disiplin dari pengembangan perangkat lunak yang berdasar pada nilai-nilai kesederhanaan, komunikasi, umpan balik, dan keberanian [8]. XP melibatkan semua anggota tim dalam setiap pekerjaannya: *pair programming* (coding bersama) dan semua pekerjaan lainnya secara bersama-sama. Jadi setiap anggota tim terlibat dalam semua aktifitas dan tidak ada pembagian tugas. Semua memiliki tanggung jawab yang sama, teknik menulis program yang sama, dan lain sebagainya. Hal ini sangat cocok diterapkan dalam penelitian ini karena coding dalam penelitian ini dilakukan oleh 1 orang saja. Karena dalam XP, setiap anggota tim juga melakukan semua pekerjaan tanpa ada pembagian tugas. Siklus hidup XP dapat dilihat pada gambar di bawah ini. Tahapan-tahapannya merupakan iterasi yang dapat diulang sampai produk sesuai dengan keinginan pelanggan. Dimulai dari tahap perencanaan, perancangan, pengkodean, dan pengujian. Setiap sampai pada tahap pengujian, perangkat lunak sudah dapat dipublikasi kepada pelanggan untuk dicoba. Dan apabila masih ada kekurangan, maka tahapan tersebut dimulai kembali. Begitu seterusnya.



Gambar 1 Siklus hidup Extreme Programming

Perancangan middleware antara aplikasi pada sistem berjalan dan e-Faktur dari DJP menggunakan bahasa pemrograman C# dengan menggunakan IDE Microsoft Visual Studio 2015 [9] yang menghasilkan aplikasi desktop yang dapat dieksekusi dalam lingkungan .NET Framework [10] versi 4.0 atau yang lebih baru.

### 3. HASIL DAN PEMBAHASAN

Perancangan middleware antara sistem berjalan dengan aplikasi e-Faktur ini diawali dengan menganalisa format file impor yang dibutuhkan oleh aplikasi e-Faktur dari DJP. Aplikasi ini membutuhkan file impor dengan ekstensi *Comma Separated Value (\*.csv)* [11]. Apabila dibuka dengan aplikasi WPS Office [12] (SITASI ADA DI TENGAH) atau Microsoft Excel 2010 [13] maka tampilannya adalah seperti gambar di bawah ini:

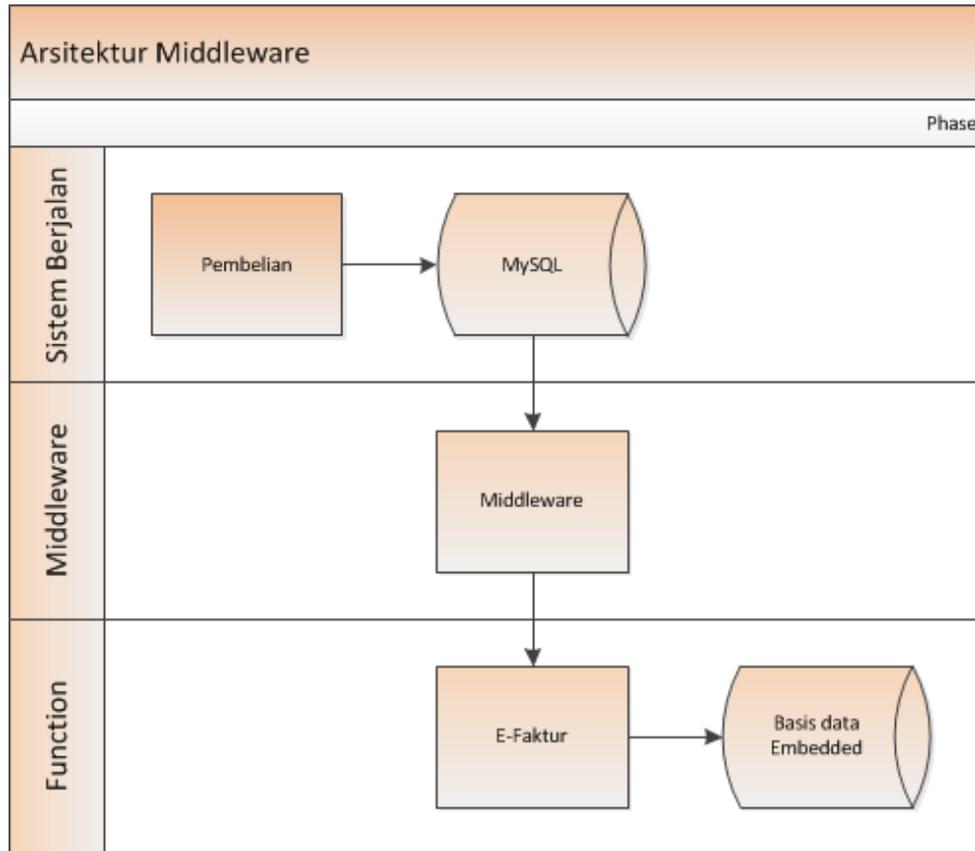
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	FK	KD_JENIS_TI	FG_PENGGANTI	NOMOR_FA	MASA_PAJ	TAHUN_PAJ	TANGGAL_F	NPWP	NAMA	ALAMAT_LE	JUMLAH_DF	JUMLAH_PF	JUMLAH_PF	ID_KETERAN	FG_UANG_J	UANG_MUK	UANG_MUK	UANG_MUK	REFERENSI
2	LT	NPWP	NAMA	JALAN	BLOK	NOMOR	RT	RW	KECAMATA	KELURAHAN	KABUPATEN	PROPINSI	KODE_POS	NOMOR_TELEPON					
3	OF	KODE_OBJE	NAMA	HARGA_SAT	JUMLAH_B	HARGA_TO	DISKON	DPP	PPN	TARIF_PPNE	PPNBM								
4	FK	1	0	3.32E+11	8	2016	29/08/2016	0	CIRCUIT MC	JL RAYA SO	492273	49227	0	0	0	0	0	0	0
5	OF	HB3 12359	HB3 12359	51818	10	466364	25909	492273	49227.3	0	0	0	0	0	0	0	0	0	0
6	FK	1	0	3.32E+11	8	2016	29/08/2016	0	BERJAYA VA	JL SIAM NO	254546	25454	0	0	0	0	0	0	0
7	OF	H4FIT 1256	H4FIT 1256	25456	10	254546	0	254546	25454.5	0	0	0	0	0	0	0	0	0	0
8	FK	1	0	3.32E+11	8	2016	30/08/2016	0	MAYANG S/J	L TRANS KI	672728	67272	0	0	0	0	0	0	0
9	OF	H4FIT 2456	H4FIT 2456	33637	20	672728	0	672728	67272.7	0	0	0	0	0	0	0	0	0	0
10	FK	1	0	3.32E+11	8	2016	31/08/2016	0	SINAR JAYA	JL TRANS KI	2472728	24727	0	0	0	0	0	0	0
11	OF	H1 12258 P	H1 12258 P	21820	10	130911	43636	174547	17454.5	0	0	0	0	0	0	0	0	0	0
12	OF	H1 13258 2	H1 13258 2	33636	10	201818	67273	269091	26909.1	0	0	0	0	0	0	0	0	0	0
13	OF	H3 12336 P	H3 12336 P	16364	10	98182	32727	130909	13090.9	0	0	0	0	0	0	0	0	0	0
14	OF	H3FIT 1245	H3FIT 1245	25455	10	152727	50909	203636	20363.6	0	0	0	0	0	0	0	0	0	0
15	OF	H3 13628 R	H3 13628 R	29091	10	174545	58182	232727	23272.7	0	0	0	0	0	0	0	0	0	0
16	OF	H4FIT 1256	H4FIT 1256	25455	10	152727	50909	203636	20363.6	0	0	0	0	0	0	0	0	0	0
17	OF	H4FIT 2456	H4FIT 2456	33636	10	201818	67273	269091	26909.1	0	0	0	0	0	0	0	0	0	0
18	OF	H4FIT 1245	H4FIT 1245	30000	10	180000	60000	240000	24000	0	0	0	0	0	0	0	0	0	0
19	OF	H4 12342 P	H4 12342 P	24545	10	147273	49091	196364	19636.4	0	0	0	0	0	0	0	0	0	0
20	OF	H11 12362	H11 12362	69091	10	414545	138182	552727	55272.7	0	0	0	0	0	0	0	0	0	0

Gambar 2 Tampilan file impor (\*.csv) pada Microsoft Excel 2010.

Pada gambar di atas, dapat dilihat bahwa file impor ini memiliki 3 buah header, yaitu:

1. Header Faktur Pajak, merupakan header yang berisi informasi lengkap mengenai faktur pajak yang diterbitkan. Kolom-kolomnya antara lain FK, KD\_JENIS\_TRANSAKSI, FG\_PENGGANTI, NOMOR\_FAKTUR, MASA\_PAJAK, TAHUN\_PAJAK, TANGGAL\_FAKTUR, NPWP, NAMA, ALAMAT LENGKAP, JUMLAH\_DPP, JUMLAH\_PPN, JUMLAH\_PPNBM, ID\_KETERANGAN\_TAMBAHAN, FG\_UANG\_MUKA, UANG\_MUKA\_DPP, UANG\_MUKA\_PPN, UANG\_MUKA\_PPNBM, REFERENSI.
2. Header Pengusaha Kena Pajak (lawan transaksi), berisi informasi lengkap mengenai lawan transaksi. Header ini digunakan untuk merekam informasi lawan transaksi ke dalam tabel referensi "lawan transaksi" apabila lawan transaksi tersebut belum pernah direkam sebelumnya. Perlu diperhatikan bahwa apabila lawan transaksi memiliki NPWP, maka informasi di bawah ini harus diisi sesuai dengan informasi yang tertera pada kartu NPWP. Kolom-kolomnya antara lain: NAMA, JALAN, BLOK, NOMOR, RT, RW, KECAMATAN, KELURAHAN, KABUPATEN, PROPINSI, KODE\_POS, NOMOR\_TELEPON.
3. Header Detil Transaksi (barang / jasa), berisi informasi mengenai barang yang dibeli pada faktur pajak. Kolom-kolomnya antara lain: NAMA, HARGA\_SATUAN, JUMLAH\_BARANG, HARGA\_TOTAL, DISKON, DPP, PPN, TARIF\_PPNBM, PPNBM.

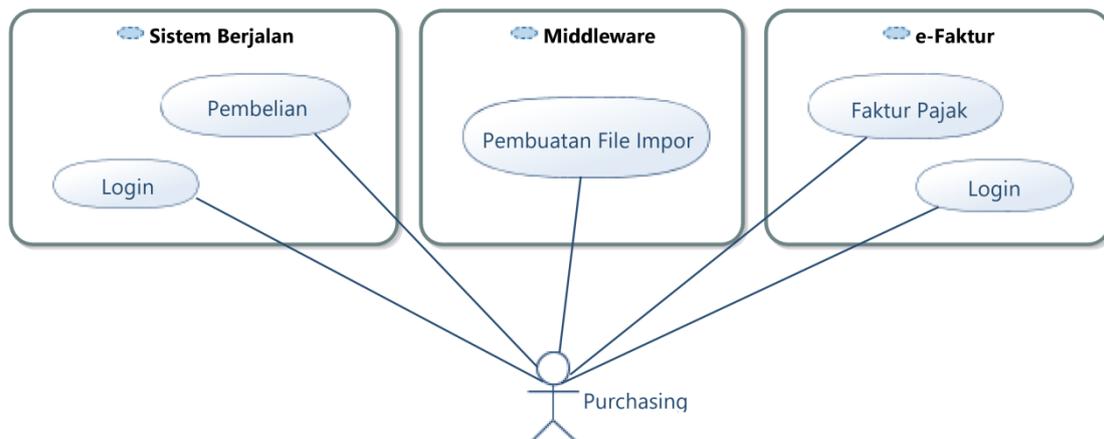
Setelah mengetahui format dari file impor yang dibutuhkan oleh aplikasi e-Faktur, tahap selanjutnya adalah merancang arsitektur middleware yang dapat menghasilkan file impor tersebut dari sistem informasi dagang. Arsitektur middleware bertujuan untuk mendiskripsikan posisi sistem informasi dagang, middleware dan aplikasi e-Faktur serta interaksi yang terjadi antara ketiga komponen tersebut. Berikut ini adalah arsitektur middleware yang diusulkan:



Gambar 3 Arsitektur Middleware antara Sistem Berjalan dengan Aplikasi e-Faktur.

Gambar di atas menunjukkan bahwa middleware tidak masuk ke dalam salah satu sistem yang ada. Middleware berdiri sendiri dan bertugas menghubungkan sistem berjalan dengan aplikasi e-Faktur. Middleware dapat mengakses basis data pembelian pada sistem berjalan untuk memperoleh data yang diperlukan. Dari data tersebut middleware mengkonversinya menjadi data yang dapat dimengerti oleh aplikasi e-Faktur, yaitu sebuah file impor faktur pajak masukan. Middleware tidak dapat mengakses basis data embedded dari aplikasi e-Faktur.

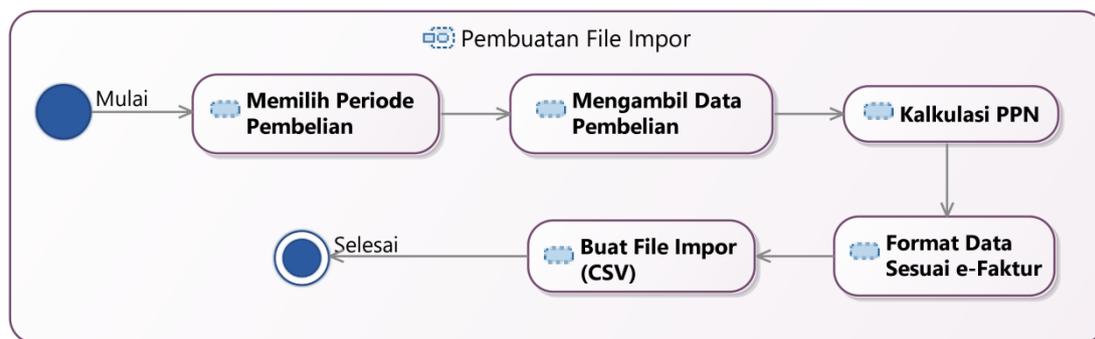
Untuk mengetahui wewenang pemakai terhadap aplikasi yang ada, dapat diketahui melalui salah satu diagram UML yaitu Use Case. Diagram Use Case untuk penelitian ini dapat dilihat pada gambar di bawah ini:



Gambar 4 Use Case Middleware antara Sistem Informasi Dagang dan Aplikasi e-Faktur.

Pada gambar 4 dapat dilihat bahwa terdapat 2 aktor yang berperan dalam prosedur pembelian hingga data masuk ke dalam aplikasi e-Faktur. Aktor pertama adalah Purchasing yang bertugas membuka faktur pembelian untuk operasional perusahaan. Data yang diinput purchasing akan masuk ke dalam basis data sistem berjalan. Setelah data terekam, maka Admin Pajak dapat mengambilnya menggunakan middleware yang dirancang. Middleware menarik data dari basis data sistem berjalan kemudian mengkonversinya menjadi sebuah file impor faktur pajak masukan. Dari file impor tersebut, admin pajak dapat mengimpornya masuk ke dalam basis data embedded aplikasi e-Faktur.

Tahapan aktivitas yang dilakukan dapat dilihat pada UML yaitu Activity Diagram. Activity Diagram pada saat membuat file impor dari middleware dapat dilihat pada gambar di bawah ini:



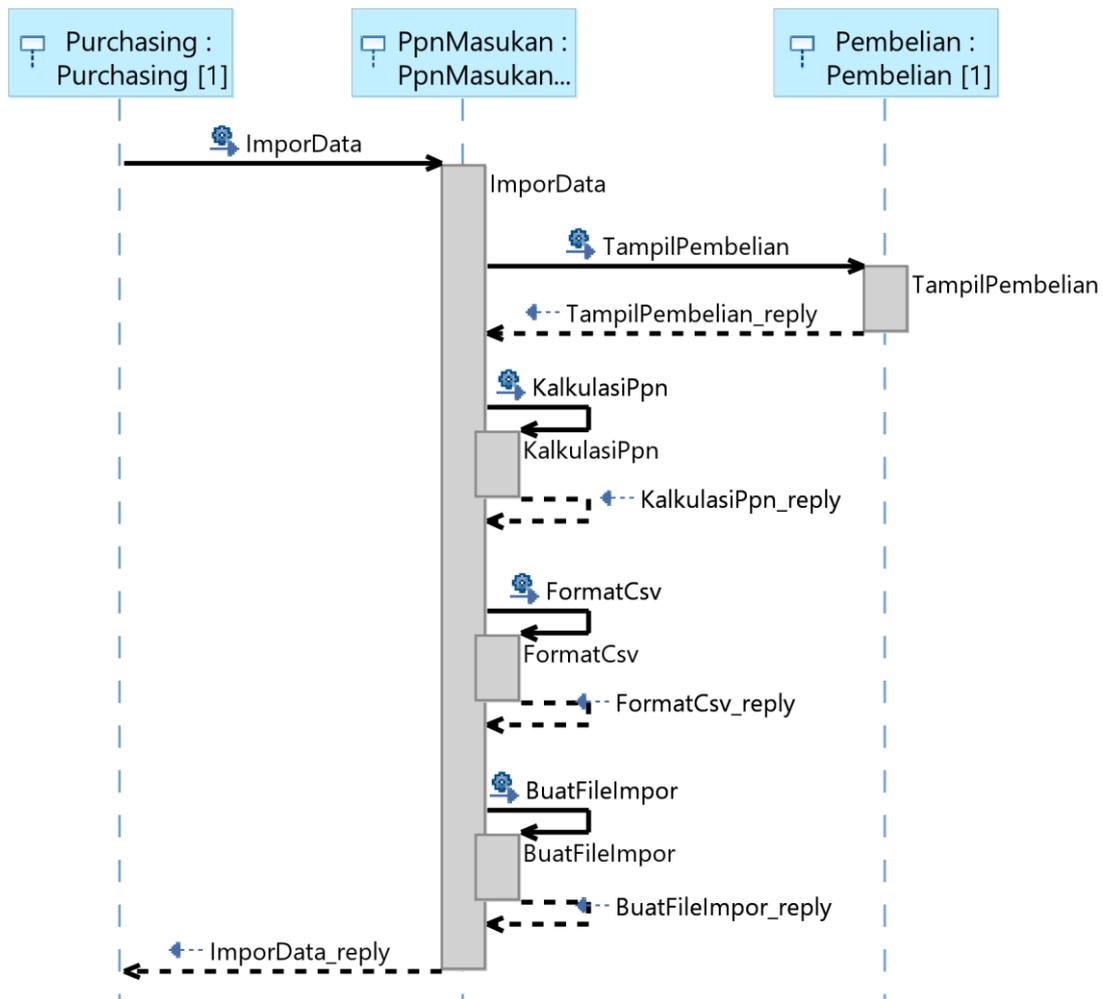
Gambar 5 Activity Diagram Pembuatan File Impor.

Tahapan pertama dari Activity Diagram di atas adalah memasukkan periode pembelian. Admin pajak memasukkan periode pembelian yang diinginkan yaitu tanggal awal dan tanggal akhir periode pembelian. Selanjutnya middleware akan menarik data pembelian berdasarkan periode yang sudah dimasukkan.

Dari data pembelian tersebut middleware mulai menghitung setiap record secara rinci, yaitu per barang yang dibeli. Menghitung Dasar Pengenaan Pajak (DPP) adalah dengan rumus  $(\text{Harga Satuan} - \text{Diskon}) * \text{Qty}$ . Perlu diperhatikan bahwa DPP dihitung setelah dikurangi diskon. Setelah mendapat nilai DPP, maka middleware akan memvalidasi nilai DPP tersebut. Aturan dari aplikasi e-Faktur adalah nilai DPP harus bulat. Maka dari itu, apabila hasil perhitungan DPP adalah desimal. Perlu dilakukan penyesuaian supaya mendapat nilai bulat. Dari nilai DPP yang sudah divalidasi tersebut, kemudian akan dikalikan 10% untuk mendapatkan nilai PPN per barang. Proses ini dilakukan kembali untuk barang yang lain dalam 1 faktur pembelian.

Setelah selesai mengulang proses dalam 1 faktur pembelian, maka dilakukan akumulasi terhadap DPP dan PPN masukan. Middleware selanjutnya akan memvalidasi apakah nilai akumulasi DPP dan PPN masukan sudah sesuai. Apabila belum, maka dilakukan penyesuaian sampai nilainya pas. Yang artinya  $\text{DPP} * 10\%$  harus bernilai setara dengan nilai PPN yang sudah diakumulasi. Proses ini diulang pula untuk setiap faktur pembelian dalam periode pembelian yang dipilih oleh admin pajak.

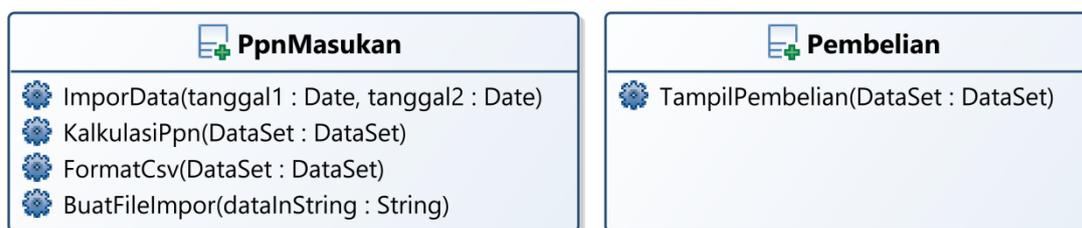
Setelah selesai menghitung DPP dan PPN masukan untuk setiap faktur pembelian, maka middleware akan memformat data yang dikumpulkan menjadi format file impor yang sudah diterangkan pada bagian sebelumnya. File yang sudah diformat akan ditulis ke dalam file *Comma Separated Value* (\*.csv). File ini akan dapat dibaca oleh aplikasi e-Faktur untuk diimpor ke dalam basis data embedded.



Gambar 6 Sequence Diagram Pembuatan File Impor.

Pada gambar 6, penulis menyajikan Sequence Diagram yang menunjukkan interaksi antara purchasing dengan 2 class yaitu PpnMasukan dan class Pembelian. Urutan prosedur juga dapat dilihat pada diagram ini. Yang pertama yaitu purchasing memanggil fungsi `ImporData()` dengan 2 parameter yaitu `tanggal1` dan `tanggal2` dengan tipe data *date*. Kedua parameter ini merupakan periode pembelian yang ingin dibuat file impor-nya. Tahap selanjutnya, class `PpnMasukan` akan meminta data pembelian kepada class `Pembelian` melalui fungsi `SelectPembelian()` yang mengembalikan nilai (*return value*) berupa *DataSet*. *DataSet* merupakan kumpulan record dalam .NET Framework yang berisi record-record dari basis data. *DataSet* tersebut kemudian diproses oleh class `PpnMasukan` melalui fungsi `KalkulasiPpn()`. Rincian proses pada fungsi ini sudah dijelaskan pada diagram Activity. Fungsi ini juga mengembalikan nilai dalam bentuk *DataSet*. Fungsi `FormatDataSesuaiFormatFileImpor()` akan memproses nilai kembalian ini menjadi tipe data *string* yang berisi format file impor yang sudah dapat dibaca oleh aplikasi e-Faktur. Format file impor yang sudah tersedia kemudian diubah oleh fungsi `BuatFileImpor()` menjadi file CSV.

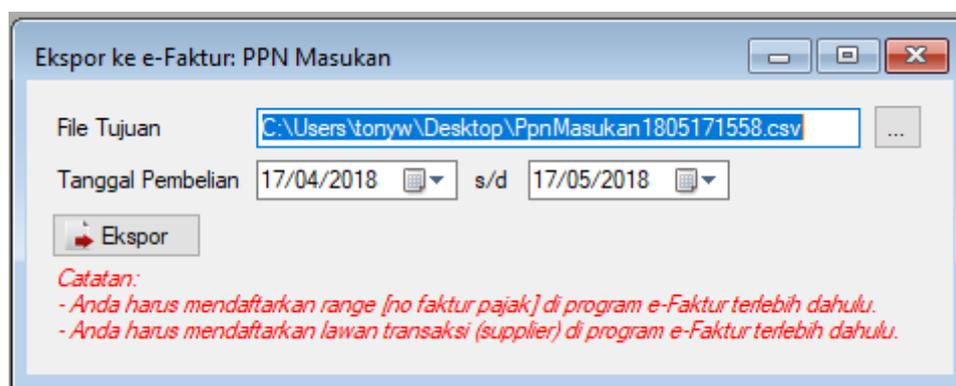
*Class diagram* digunakan untuk menampilkan beberapa kelas yang ada dalam middleware. *Class diagram* mendeskripsikan jenis-jenis objek dalam sistem. *Class diagram* menunjukkan properti dan operasi sebuah kelas dan batasan-batasan yang terdapat dalam hubungan-hubungan objek tersebut. *Class diagram* memiliki 3 bagian utama yaitu attribute, operation, dan name. (Gambar 7).



Gambar 7 Class Diagram Middleware antara Sistem Berjalan dengan Aplikasi e-Faktur.

Pada gambar 7, dapat dilihat semua class beserta atribut dan fungsi yang digunakan pada gambar 6 (Sequence Diagram). Nilai kembali juga dijabarkan secara jelas sehingga koneksi antara gambar 5 dan gambar 6 menjadi nyata.

Middleware antara sistem berjalan dan aplikasi e-Faktur merupakan program berbasis desktop yang dirancang dengan menggunakan bahasa pemrograman C# dari .NET Framework 4.0. Tools yang digunakan adalah Visual Studio 2015. Berikut merupakan hasil rancangan form pembuatan file impor (Gambar 8).

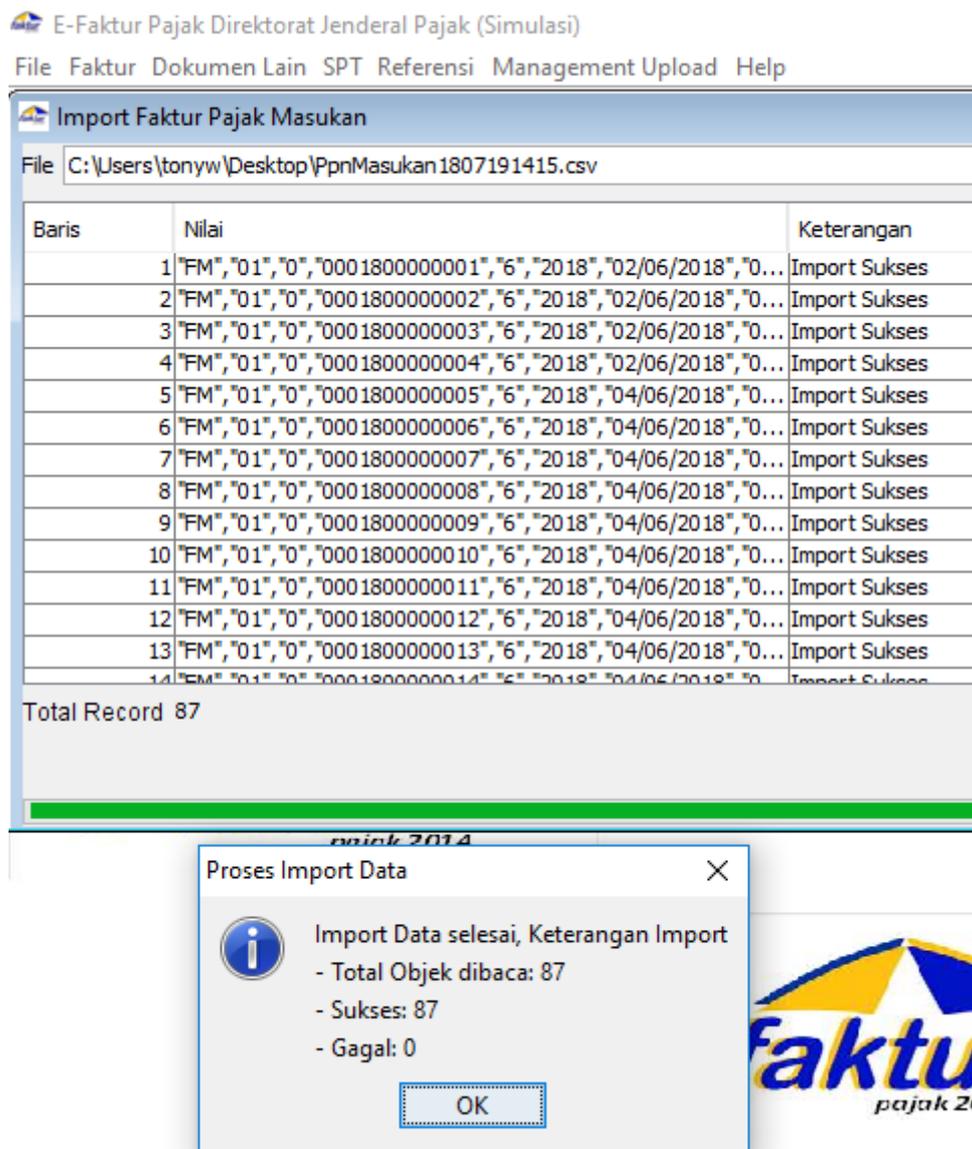


Gambar 8 Hasil Rancangan Form Pembuatan File Impor

Gambar di atas menunjukkan tampilan yang sangat sederhana dari middleware. Sesuai dengan tujuan dari middleware yaitu middleware adalah sebuah software yang dapat mengatasi heterogenitas dan kompleksitas sebuah sistem yang terdistribusi [2], maka rancangan ini benar-benar merefleksikan tujuan tersebut. Pada formulir ini seorang fakturis cukup memilih file tujuan dan periode pembelian. Setelah klik tombol Ekspor, maka middleware akan melakukan proses yang sudah dibahas pada bagian sebelumnya dan menghasilkan sebuah file *Comma Separated Value* (\*.csv) yang dapat dibaca oleh aplikasi e-Faktur.

Pengujian sistem dilakukan penulis untuk menguji 2 hal. Yang pertama adalah apakah middleware yang dibangun sudah sesuai dengan kebutuhan fungsional atau belum. Pengujian kedua adalah seberapa besar dampak yang dirasakan oleh kehadiran middleware ini dibandingkan tanpa adanya middleware.

Pengujian pertama dilakukan dengan metode blackbox testing, yaitu menguji apakah fitur yang menjadi kebutuhan sistem sudah dapat dipenuhi oleh middleware yang dihasilkan. Berikut penulis sajikan capture hasil impor pada aplikasi e-Faktur menggunakan file \*.csv yang dihasilkan oleh middleware (Gambar 8).



Gambar 9 Hasil Impor pada Aplikasi e-Faktur dari DJP.

Pada gambar 9, dapat dilihat bahwa penulis menggunakan aplikasi e-Faktur untuk melakukan proses impor dan berjalan dengan sukses. Total record yang dibaca adalah 87 record, yang berarti ada 87 faktur pembelian. Record yang sukses diimpor adalah 87 record, yang artinya berhasil 100%. Dengan demikian, pengujian pertama telah berhasil.

Pengujian kedua dilakukan dengan cara membandingkan peningkatan efisiensi sebelum dan sesudah penerapan middleware. Waktu yang diperlukan untuk menginput data berupa 87 faktur pembelian ke dalam aplikasi e-Faktur secara manual dibandingkan dengan waktu yang diperlukan untuk membuat file impor dengan middleware dan kemudian mengimpornya ke dalam aplikasi e-Faktur. Untuk input manual ke dalam aplikasi e-Faktur, seorang admin pajak membutuhkan waktu paling cepat 2 menit untuk setiap faktur pembelian. Apabila dikalikan dengan 87 faktur pembelian, maka waktu yang diperlukan adalah 174 menit (1 jam 14 menit) dengan asumsi admin pajak tersebut tidak beristirahat atau jeda sama sekali. Sedangkan waktu yang diperlukan oleh middleware untuk membuat sebuah file impor yang berisi 87 faktur pembelian adalah antara 1 – 2 detik saja. Ditambah dengan waktu yang diperlukan oleh admin pajak untuk melakukan impor ke dalam aplikasi e-Faktur yaitu paling cepat 1 menit. Jadi total waktu yang dibutuhkan hanya 1 menit 2 detik saja. Untuk aktivitas input secara manual ke

dalam aplikasi e-Faktur, admin pajak perlu memeriksa kembali hasil input untuk memastikan tidak ada kesalahan. Waktu yang diperlukan untuk memeriksa kembali sebanyak 87 faktur pembelian adalah sekitar 30 – 40 menit. Dan tingkat kesalahan yang biasa terjadi dalam menginput 176 faktur pembelian adalah 5 hingga 10 faktur pembelian.

Tabel hasil perbandingan sistem lama dibandingkan dengan sistem baru sesudah implementasi middleware dapat dilihat pada Tabel 1.

Tabel 1 Perbandingan Pengujian Sistem

Pengujian	Manual	Middleware
Memindahkan 87 faktur pembelian ke aplikasi e-Faktur	1 jam 14 menit	1 menit 2 detik
Memeriksa kembali faktur hasil input	30 – 40 menit	0 detik
Tingkat kesalahan input	5 – 10 faktur	0 faktur

#### 4. KESIMPULAN

Berdasarkan hasil dan pembahasan, maka dapat diambil kesimpulan bahwa implementasi middleware berhasil menghubungkan sistem berjalan dengan aplikasi e-Faktur. Hal ini berarti middleware mampu meningkatkan efisiensi kerja yang sangat besar karena perusahaan tidak perlu melakukan input data operasional dua kali seperti sebelum dilakukan implementasi. Selain itu, hasil implementasi sistem mengurangi tingkat kesalahan input data sebanyak 100%. Berdasarkan pengujian sistem yang dilakukan, penerapan middleware ini dapat dikatakan telah menghilangkan kompleksitas terhadap heterogenitas sistem yang dialami oleh CV. Bess.

#### 5. SARAN

Penulis menyadari bahwa middleware ini masih bisa dikembangkan lagi untuk fitur Retur PPN Penjualan dan Retur PPN Pembelian. Maka harapan penulis untuk penelitian selanjutnya adalah supaya dapat dikembangkan menjadi lebih lengkap dan semakin meningkatkan efisiensi kerja serta membantu pengusaha dalam melaporkan pajak yang benar dan akurat.

#### DAFTAR PUSTAKA

- [1] "DJP dan PKP siap melaksanakan E-Faktur," 2015. [Online]. Available: <http://www.pajak.go.id/content/article/djp-dan-pkp-siap-melaksanakan-e-faktur>.
- [2] M. P. Canares, "Creating the Enabling Environment for More Transparent and Betterresourced Local Governments: A Case of E-taxation in the Philippines," *Information Technology for Development*, vol. 22, no. 1, p. 121, 2016.
- [3] H. Ali, A. A. Mamun and S. Anwar, "A Comprehensive Study of Advancement of Electrical Power Grid and Middleware based Smart Grid Communication Platform," *International Journal of Advancements in Technology*, vol. 7, no. 2, p. 4, 2016.

- [4] T. Wijaya, "Perancangan Middleware untuk Menghubungkan Sistem Informasi Dagang dengan Aplikasi e-Faktur dari Direktorat Jenderal Pajak," *CCIT Journal*, vol. 10, no. 1, pp. 35-45, 2017.
- [5] W. Purhantara, *Metode Penelitian Kualitatif untuk Bisnis*, Yogyakarta: Graha Ilmu, 2010.
- [6] A. A. Albarqi, "The Proposed L-Scrumban Methodology to Improve the Efficiency of Agile Software Development," *I.J. Information Engineering and Electronic Business*, vol. 3, p. 13, 2018.
- [7] M. Beedle, A. v. Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Highsmith, A. Hunt, R. Jeffries, J. Kern, B. Marick, R. C. Martin, K. Schwaber, J. Sutherland and D. Thomas, "Signatories: The Agile Manifesto," 2001. [Online]. Available: <http://agilemanifesto.org/>. [Accessed 8 September 2018].
- [8] L. Lindstrom and R. Jeffries, "Extreme Programming and Agile Software Development Methodologies," *Information Systems Management*, pp. 41-52, 2004.
- [9] Microsoft Developer Network (MSDN), "Visual Studio IDE," Microsoft, 23 August 2018. [Online]. Available: <https://msdn.microsoft.com/en-us/library/dn762121.aspx>. [Accessed 8 September 2018].
- [10] Microsoft Docs, "Overview of the .NET Framework," Microsoft, 30 March 2017. [Online]. Available: <https://docs.microsoft.com/en-us/dotnet/framework/get-started/overview>. [Accessed 8 September 2018].
- [11] Y. Shafranovic, "RFC 4180 - Common Format and MIME Type for Comma-Separated Values (CSV) Files," SolidMatrix Technologies, Inc., Oktober 2005. [Online]. Available: <https://tools.ietf.org/html/rfc4180>. [Accessed 24 Juni 2018].
- [12] WPS, "WPS Office Perfect Compatible with Microsoft (MS) (Word, Excel, PowerPoint) for PC and Mobile - Free Download," KingSoft Corp, [Online]. Available: <https://www.wps.com/>. [Accessed 25 Juni 2018].
- [13] Microsoft Office, "Microsoft Excel 2010," Microsoft, [Online]. Available: <https://products.office.com/id-id/microsoft-excel-2010>. [Accessed 26 Juni 2018].