

PROTOTIPE PERANGKAT LUNAK *LEAN* *MANUFACTURING* UNTUK MENGIDENTIFIKASI *WASTE**

Muhammad Abdul Mubarak, Rispianda, Khuria Amila

Jurusan Teknik Industri
Institut Teknologi Nasional (Itenas) Bandung

Email: abdulmubarak2205@gmail.com

ABSTRAK

PT X merupakan salah satu perusahaan besar yang memproduksi pesawat terbang, komponen-komponen pesawat terbang serta penyedia jasa service pesawat di Indonesia. Saat ini PT X sudah memiliki divisi Lean Manufacturing, namun pada kenyataannya masih terdapat pemborosan dalam perusahaan. Pengelolaan data pemborosan masih menjadi masalah karena data yang diolah jumlahnya banyak namun untuk mengerjakannya dituntut dalam waktu yang singkat. PT X juga terkendala dengan kurangnya tenaga kerja di divisi Lean dan cara kerja secara manual yang mengakibatkan data pemborosan tidak dapat teridentifikasi dengan baik. Pada penelitian ini dirancang prototipe lean manufacturing dengan menggunakan MySQL untuk pembuatan database, Visual Studio untuk interface, serta Ms. Excel untuk report yang dilakukan oleh evaluator. Dengan aplikasi ini, PT X mampu mengidentifikasi waste dengan waktu yang lebih cepat, up to date, dan lebih spesifik.

Kata kunci: *Pemborosan, basis data, lean manufacturing, prototipe*

ABSTRACT

PT X is one of the major companies that manufacture aircraft, aircraft components as well as the best service provider in Indonesia. Currently PT X already has Lean Manufacturing division, but in reality there are waste within the company. Data waste is still a problem because the data is processed polynomial, but the time required to do it in a short time. PT X is also constrained by a lack of manpower in the division of Lean and how to work the data manually resulting waste can not be identified properly. In this study designed a prototype of lean manufacturing by using MySQL for database creation, Visual Studio for interface, as well as Ms. Excel for report conducted by evaluators. With this application, PT X is able to identify waste in a short time, up to date, and more specific.

Keywords: *Waste, database, Lean manufacturing, prototype*

* Makalah ini merupakan ringkasan dari Tugas Akhir yang disusun oleh penulis pertama dengan pembimbingan penulis kedua dan ketiga. Makalah ini merupakan draft awal dan akan disempurnakan oleh para penulis untuk disajikan pada seminar nasional dan/atau jurnal nasional.

1. PENDAHULUAN

1.1 Pengantar

Lean Manufacturing merupakan salah satu metode yang fokus pada penggunaan dan pemberdayaan sumber daya untuk menciptakan *value* bagi pelanggan dengan cara menghilangkan *waste* (pemborosan) yang terjadi di lantai produksi. Hal tersebut mampu menghasilkan proses yang lebih efektif dan efisien serta menghasilkan *output* yang lebih baik.

PT Dirgantara Indonesia (PT DI) merupakan salah satu perusahaan besar yang memproduksi pesawat terbang, komponen-komponen pesawat serta penyedia jasa *service* pesawat di Indonesia. Saat ini PT DI memiliki masalah dalam proses identifikasi pemborosan dalam sistem produksi perusahaannya. Jika dilihat dari rasio nilai tambah dibandingkan dengan pemborosan yang terjadi dalam sistem produksinya, menurut divisi *Lean Manufacturing* PT DI, rasio yang dimiliki perusahaan ini tidak mampu melebihi angka 10%. Padahal perusahaan yang dikatakan perusahaan *Lean* adalah perusahaan yang memiliki rasio antara nilai tambah dengan pemborosan minimal 30% (Gasperzs, 2007).

Dalam penerapan metode *Lean*, divisi *Lean Manufacturing* masih mengalami kendala dalam mengelola data. Divisi ini biasanya mencatat semua pesanan pada akhir periode tertentu dan dilakukan satu tahun sekali. Divisi *Lean Manufacturing* sendiri belum memiliki jadwal tetap terkait periode perekapan catatan pesanan tersebut. Pada kenyataannya, data yang diolah pun jumlahnya tidak sedikit bisa mencapai 38064 data untuk periode Mei 2014 sampai Mei 2014. Permasalahan lain yang ada adalah jumlah tenaga kerja yang terdapat di divisi *Lean Manufacturing* tidak begitu banyak dan divisi tersebut merasa kewalahan untuk melakukan pengolahan data yang banyak dan masih dilakukan secara manual.

Jumlah data yang tidak sedikit tersebut dituntut untuk dikerjakan dalam waktu yang singkat, hal tersebut dapat menyebabkan pengelolaan data tidak akurat serta proses pengendalian tidak *up to date*. Apabila dilakukan pengolahan data 1 tahun sekali dan dilakukan secara manual, maka ukuran performansi mengenai pemborosan akan diketahui 1 tahun sekali, sehingga kesalahan yang mengakibatkan pemborosan dalam sistem produksi perusahaan tidak dapat dikendalikan secepat mungkin. Oleh karena itu, permasalahan ini dianggap menyulitkan dan sangat mungkin menimbulkan beberapa hal yang terlewatkan atau kesalahan dalam pengerjaannya sehingga sistem *Lean* di PT DI tidak maksimal.

1.2 Identifikasi Masalah

Kesulitan dalam pengelolaan data *Lean* dapat dibantu dengan adanya penggunaan aplikasi. PT DI membutuhkan suatu aplikasi atau sistem perangkat lunak untuk melakukan identifikasi pemborosan dengan metode *Lean Manufacturing*. Saat ini perusahaan sudah memiliki perangkat lunak SAP (*Systems Application and Products in Data Processing*) untuk pengumpulan data yang dibutuhkan namun tidak untuk pengolahan data. Sejauh ini belum terdapat aplikasi yang mengakomodasi permasalahan seperti yang dihadapi PT DI. Oleh karena itu perlu dibuat suatu rancangan aplikasi (prototipe) *Lean Manufacturing*. Dengan adanya prototipe ini diharapkan perusahaan mampu meningkatkan efisiensi kerja dalam melakukan pencatatan sehingga tidak perlu menunggu di akhir periode untuk merekap data serta mempermudah jaringan sistem informasi antar departemen yang saling berhubungan.

2. STUDI LITERATUR

2.1 Konsep Dasar *Lean*

Gasperz (2007) menyatakan bahwa *lean* adalah suatu upaya terus-menerus untuk menghilangkan pemborosan (*waste*) dan meningkatkan nilai tambah (*value added*) produk (barang dan/atau jasa) agar memberikan nilai kepada pelanggan (*customer value*). Tujuan

dari *lean* adalah meningkatkan terus-menerus *customer value* melalui peningkatan terus-menerus rasio antara nilai tambah terhadap *waste* (*the value to- waste ratio*) (Gasperzs, 2007). Pada dasarnya dikenal dua kategori utama pemborosan, yaitu *Type One Waste* dan *Type Two Waste* (Gasperzs, 2007). *Seven plus One* "Type of Waste" dapat dilihat pada Tabel 1.

Tabel 1. "Seven plus One" Type of Waste

No.	Waste	Penjelasan	Akar Penyebab
1	<i>Over-production</i>	Memproduksi lebih daripada kebutuhan pelanggan internal dan eksternal, atau memproduksi lebih cepat atau lebih awal dari pada waktu kebutuhan internal dan eksternal	Ketidakaadaan komunikasi, sistem balas jasa dan penghargaan yang tidak tepat, hanya berfokus pada kesibukan kerja, bukan untuk memenuhi kebutuhan pelanggan internal dan eksternal
2	<i>Delays (waiting time)</i>	Keterlambatan yang tampak melalui orang-orang yang sedang menunggu mesin, peralatan, bahan baku, supplies, perawatan atau pemeliharaan, dll. Atau bisa dianggap mesin-mesin yang sedang menunggu perawatan, orang-orang, bahan baku, peralatan, dll.	Inkonsistensi metode kerja, waktu pergantian produk yang panjang
3	<i>Transportation</i>	Memindahkan material atau orang dalam jarak yang sangat jauh dari suatu proses ke proses berikutnya yang dapat mengakibatkan waktu penanganan material bertambah	Tata letak yang jelek, ketiadaan koordinasi dalam proses, <i>poor housekeeping</i> , organisasi tempat kerja yang jelek, lokasi penyimpanan mmateriaal yang banyak dan saling berjauhan.
4	<i>Proceses</i>	Mencakup proses-proses tambahan atau aktivitas kerja yang tidak perlu dan tidak efisien	Ketidaktepatan penggunaan peralatan, pemeliharaan peralatan yang jelek, gagal mengombinasi operasi-operasi kerja, proses kerja dibuat <i>serial</i> padahal proses-proses itu tidak saling tergantung satu sama lain, yang seyogianya dapat diibuat paralel
5	<i>Inventories</i>	Pada dasarnya <i>inventories</i> menyembunyikan masalah dan menimbulkan aktivitas penanganan tambahan yang seharusnya tidak diperlukan. <i>Inventories</i> juga mengakibatkan <i>extra paperwork, extra space, dan extra cost.</i>	Peralatan yang tidak andal, aliran kerja yang tidak seimbang, pemasok yang tidak kapabel, peramalan kebutuhan yang tidak akurat, ukuran <i>batch</i> yang besar.
6	<i>Motion</i>	Setiap pergerakan dari orang atau mesin yang tidak menambah nilai kepada barang dan jasa yang akan diserahkan kepada pelanggan, tetapi hanya menambah biaya dan waktu saja	Organisasi tempat kerja yang jelek, tata letak yang jelek, metode kerja yang tidak konsisten, desain mesin yang buruk
7	<i>Defective product</i>	<i>Scrap, rework, customer returns, customer dissatisfaction</i>	<i>Incapable processes, insufficient training, ketiadaan prosedur-prosedur operasi standar</i>
7+1	<i>Defective design</i>	Desain yang tidak memenuhi kebutuhan pelanggan, penambahan <i>features</i> yang tidak perlu.	<i>Lack oof customer input in design, over-deesign</i>

2.2 Pengembangan Perangkat Lunak

Terdapat bagian-bagian tertentu dalam melakukan pengembangan perangkat lunak. Untuk merancang struktur dasar dibutuhkan arsitektur perangkat lunak dan pemrograman modular. Untuk sistem aliran data perangkat lunak dibutuhkan context diagram dan entity relationship diagram (ERP), sedangkan untuk tampilan menggunakan visual studio dan database menggunakan MySQL.

Menurut Microsoft (2009), arsitektur perangkat lunak adalah proses yg mendefinisikan solusi yg terstruktur yang memenuhi kebutuhan teknis dan operasional, disisi lain mengoptimasi quality dari sebuah aplikasi yg meliputi: *performance, security, dan manageability.*

Pemrograman modular adalah teknik desain perangkat lunak yang menekankan memisahkan fungsi program ke masing-masing bagian yang independen, modul pertukaran, sehingga masing-masing berisi segala sesuatu yang diperlukan untuk mengeksekusi hanya satu aspek dari fungsi yang diinginkan (Haridi, 2015).

Context Diagram merupakan suatu diagram yang menggambarkan secara menyeluruh dari ruang lingkup sistem. *Context diagram* tersebut merupakan *level* tertinggi dari DFD yang menggambarkan seluruh *input* ke sistem, *output* dari sistem dan memberi gambaran tentang keseluruhan sistem. Sistem dibatasi oleh *boundary* (dapat digambarkan dengan garis putus). Dalam diagram konteks hanya ada satu proses dan tidak boleh terdapat *store* (Ladjamudin, 2005).

Entity Relationship Diagram (ERD) digunakan untuk mendokumentasikan data perusahaan dengan mengidentifikasi entitas dan hubungannya. ERD merupakan suatu model jaringan yang menggunakan susunan data yang disimpan pada sistem secara abstrak (Yakub, 2012).

Bahasa yang digunakan dalam *Visual Studio 2008* adalah bahasa *Visual Basic*. *Microsoft Visual Basic* (sering disingkat sebagai VB saja) merupakan sebuah bahasa pemrograman yang menawarkan *Integrated Development Environment* (IDE) visual untuk membuat program perangkat lunak berbasis sistem operasi *Microsoft Windows* dengan menggunakan model pemrograman (COM) (Phil, 2001). *Visual Basic 2008* sendiri merupakan bagian dari *Microsoft Visual Basic .NET*. *Microsoft Visual Basic .NET* adalah sebuah alat untuk mengembangkan dan membangun aplikasi yang bergerak di atas sistem *.NET Framework*, dengan menggunakan bahasa BASIC.

MySQL merupakan manajemen *database* SQL yang bersifat *Open Source*. Sistem *database* *MySQL* mendukung beberapa fitur seperti *multithreaded*, *multi user*, dan *SQL database management system* (DBMS). *Database* ini dibuat untuk keperluan sistem *database* yang cepat, handal dan mudah untuk digunakan. *MySQL* juga merupakan sebuah implementasi dari sistem manajemen basis data relasional (RDBMS) yang didistribusikan secara gratis di bawah lisensi GPL (*General Public License*) (Raharjo, 2011).

3. METODOLOGI PENELITIAN

Dalam penelitian ini, dilakukan metode penelitian sebagai berikut:

(i) Tahapan Rumusan Masalah

Pada tahap ini akan dilakukan perumusan masalah dari sistem yang nantinya akan dibuat suatu prototipe. Terdapat solusi terhadap permasalahan tersebut, maka dibuatlah suatu prototipe perangkat lunak *lean manufacturing* sebagai alat bantu untuk mempermudah dalam melakukan pengidentifikasian pemborosan yang ada dalam perusahaan.

(ii) Tahapan Studi Literatur

Studi literatur memuat kumpulan materi mengenai sejarah *lean*, konsep dasar *lean*, arsitektur perangkat lunak, pemrograman modular, alat perancang aliran data, alat perancang basis data, serta alat penunjang pembuatan prototipe.

(iii) Tahapan Pengidentifikasian Sistem *Lean Manufacturing* Perusahaan

Tahap pengidentifikasian sistem *lean* perusahaan menjelaskan tentang kondisi dari prosedur perusahaan saat ini ketika memproses produk dari mulai pesanan sampai menjadi barang jadi serta jenis pemborosan yang mampu diidentifikasi disesuaikan dengan kondisi nyata yang terjadi dalam perusahaan.

(iv) Tahapan Perancangan Prototipe Perangkat Lunak *Lean Manufacturing*

Tahap perancangan prototipe berisi mengenai langkah-langkah penulis dalam merancang prototipe. Dimulai dari memberikan usulan prosedur hasil analisis dari prosedur perusahaan saat ini. Langkah berikutnya adalah membuat arsitektur perangkat lunak dan merancang modul-modul untuk lebih mempermudah dalam membagi peran entitas yang ada. Kemudian melakukan perancangan sistem aliran data yaitu dengan membuat *context diagram*, *hierarchy chart*, diagram nol serta diagram rinci. Lalu dibuat suatu formulasi perhitungan *waste* untuk memudahkan dalam melakukan perhitungan. Langkah selanjutnya adalah pembuatan kamus data dan ERD. Langkah terakhir adalah implementasi dari langkah-langkah sebelumnya menggunakan aplikasi VB.Net dan MySQL.

(v) Tahapan Pengujian Prototipe Perangkat Lunak *Lean Manufacturing* dan Analisis

Pengujian dilakukan untuk memeriksa keberhasilan dari masing-masing modul dalam menjalankan perannya dan membandingkan perhitungan dari perangkat lunak dengan perhitungan manual. Diketahui dari hasil pengujian menyatakan bahwa peran masing-masing modul dapat berjalan dengan baik dan hasil perhitungan dapat dikatakan valid.

(vi) Kesimpulan dan Saran

Setelah dilakukan pengujian dan analisis terhadap prototipe, maka dapat ditarik kesimpulan dari hasil yang diperoleh yaitu secara keseluruhan prototipe mampu mempermudah dan menjadi solusi dari permasalahan yang ada sesuai dengan tujuan penelitian. Kemudian terdapat beberapa saran yang ditunjukkan terhadap pengembangan prototipe untuk implementasi persoalan lainnya.

4. PENGEMBANGAN PROTOTIPE PERANGKAT LUNAK

4.1 Identifikasi Prosedur Perusahaan Saat Ini

Hasil analisis prosedur perusahaan saat ini:

- Prosedur ini memungkinkan adanya kesalahan dalam pencatatan terutama apabila data yang dimasukkan cukup banyak. Hal ini juga merupakan sebuah kegiatan yang tidak efektif dan tidak efisien karena membutuhkan waktu yang lama sehingga informasi tidak tersampaikan dengan maksimal. Kegiatan seperti itu juga bertentangan dengan konsep *Lean*.
- Pencatatan dilakukan dengan menggunakan lembar proses dari PPIC bagian DPM yang akan terus menerus dipindahtanggankan sampai akhirnya lembar tersebut berada di tangan *entryman*.
- Tidak terdapat data yang jelas mengenai operator yang bertugas atau mengeksekusi produk di stasiun kerja yang terdapat dalam perusahaan.
- Pencatatan yang dilakukan manual akan menyebabkan penumpukan arsip lembar proses pada saat *entryman* hendak memasukan data. Bahkan kemungkinan terjadinya lembar proses hilang sangat besar karena penumpukan tersebut.
- Lembar proses yang hendak di rekap jumlahnya sangat banyak dan akan menumpuk karena proses perekapan dilakukan pada akhir periode tertentu.

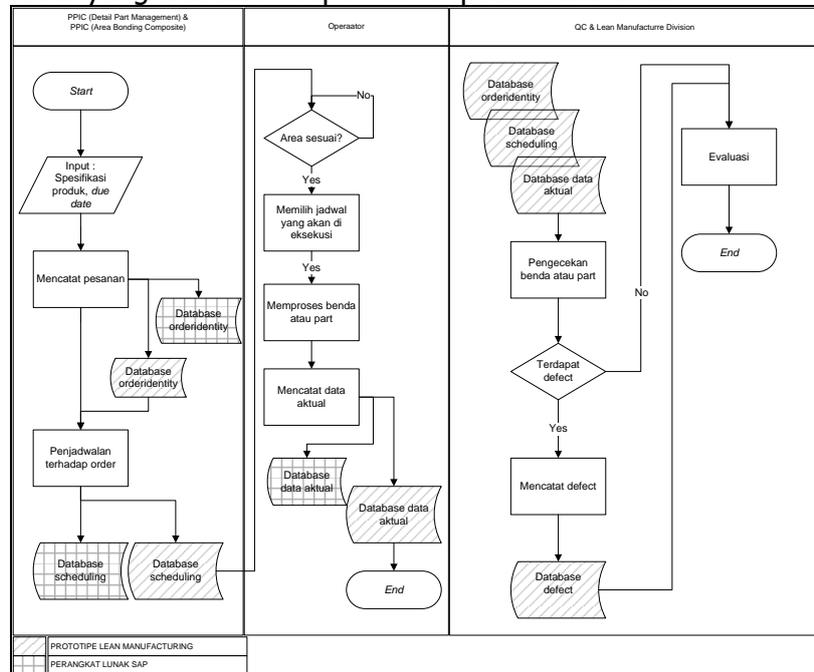
4.2 Identifikasi Pemborosan

Dari hasil pengidentifikasian pemborosan disesuaikan dengan sistem perusahaan yang ada, terdapat 2 jenis *waste* yang tidak dapat diidentifikasi, yaitu *waste overproseses* dan *waste motion*. *Waste overproseses* tidak dapat diidentifikasi karena PT DI memiliki sistem *job order*, sedangkan untuk pemborosan jenis ini diperuntukan untuk sistem *mass production*. *Waste motion* tidak dapat diidentifikasi karena penelitian tidak sampai meneliti bagaimana

operator melakukan pekerjaan secara rinci dan dibutuhkan waktu lebih lanjut untuk melakukan pengamatan.

4.3 Usulan Rancangan Prosedur

Rancangan prosedur yang diusulkan dapat dilihat pada Gambar 1.



Gambar 1. Usulan Prosedur Sistem Produksi

- Terlihat dari entitas prosedur yang diusulkan peran *entryman* tidak menjadi entitas tersendiri, melainkan sudah terwakili oleh entitas-entitas yang lainnya. Karena untuk memasukkan data ke database langsung tanpa harus melakukan pencatatan melalui lembar proses terlebih dahulu. Peran PPIC DPM dan PPIC bagian area menjadi satu peran yaitu memasukan data *order* dan kemudian melakukan penjadwalan atau membuat jadwal rinci terhadap pesanan.
- Pihak *Lean Manufacture* akan menerima semua data mengenai jadwal rinci, data aktual, serta data order yang sudah tersusun dan terintegrasi satu sama lain. Pihak *Lean* akan melakukan pencatatan untuk defect yang ada pada produk. Kemudian setelah semua data sudah komplit, pihak *Lean* akan melakukan evaluasi untuk mengidentifikasi pemborosan yang mungkin terjadi dalam sistem produksi.
- Peran dari perangkat lunak SAP tidak dihilangkan tetapi terdapat dua buah *database* yang menyimpan data jadwal rinci dan data aktual.

4.4 Perancangan Modul Perangkat Lunak

Dalam arsitektur perangkat lunak, diterapkan konsep *modularity*. *Modularity* merupakan suatu konsep yang memecah suatu sistem utuh menjadi bagian bagian sistem yang kecil dengan tujuan untuk mempermudah proses perancangan. Dengan diterapkannya konsep tersebut, maka sistem yang akan dibangun akan lebih mudah untuk meresmikan atau membagi tanggung jawab yang akan di lakukan oleh masing-masing kelompok bagian sistem yang menyusun sistem utuh tersebut.

Modul *planner* adalah modul yang berperan untuk menerima order dan mengubahnya menjadi jadwal rinci sebagai masukan dari proses nantinya. Setiap order yang masuk akan

dicatat oleh modul ini yang kemudian akan dijabarkan proses produksi apa saja yang perlu dilakukan untuk *part* dalam *order* tersebut.

Modul *doer* adalah modul yang digunakan untuk memproses data jadwal rinci berdasarkan data aktual yang ada. Modul ini akan memproses jadwal rinci yang terdapat di stasiun kerja dimana mereka ditempatkan.

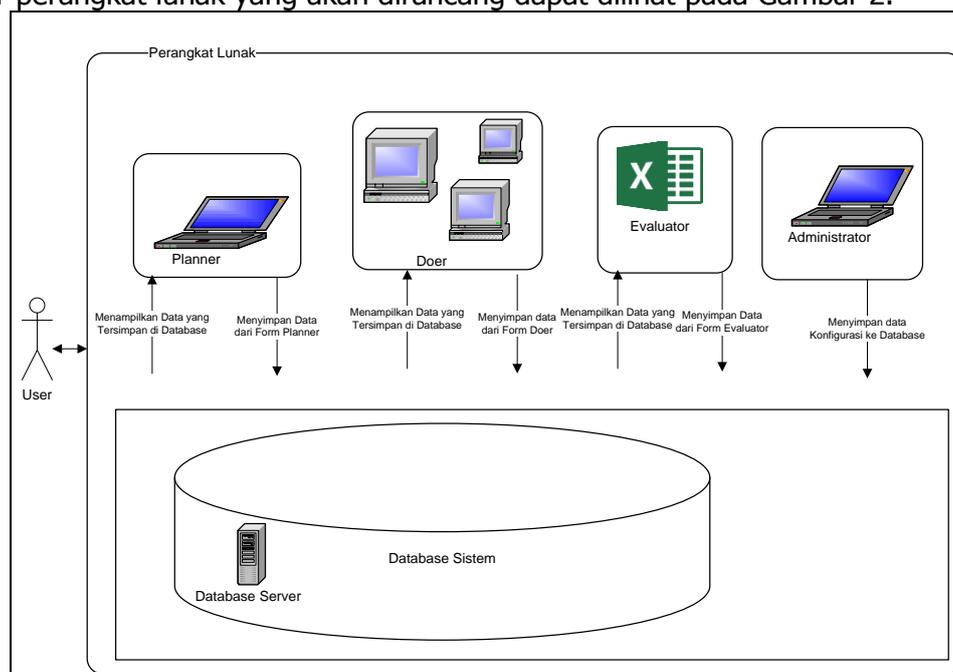
Modul *evaluator* adalah modul yang digunakan untuk mengevaluasi data aktual yang di proses oleh modul *doer*. *Evaluator* tersebut akan memperoleh data rinci baik semua data *order* yang sudah masuk dalam sistem, grafik berdasarkan data, yang kemudian akan diperiksa apakah terdapat pemborosan dalam sistem berdasarkan data *order* yang ada.

Modul *administrator* adalah modul yang digunakan untuk memonitor data *order* yang terdapat dalam *database* dan bertugas untuk memasukkan data-data petugas dan data barang sebagai penunjang proses produksi.

Modul *database server* adalah modul yang digunakan sebagai pemasok data untuk modul-modul yang lainnya.

4.5 Penerapan Arsitektur Berdasarkan Modul Sistem

Arsitektur perangkat lunak yang akan dirancang dapat dilihat pada Gambar 2.



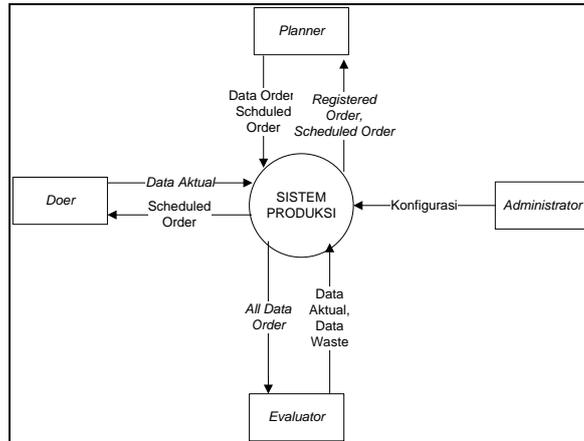
Gambar 2. Arsitektur Program Perangkat Lunak

Dari arsitektur program dapat dilihat bahwa semua komponen modul yang ada dalam perangkat lunak mendapatkan/memperoleh informasi dari satu *database* besar yang kemudian semua modul tersebutpun mengirimkan data hasil olahan ke dalam *database*. Semua data disimpan dalam satu wadah atau *server* yang didalamnya menyimpan berbagai data jadwal rinci, data aktual, maupun data hasil evaluasi.

4.6 Context Diagram

Context Diagram merupakan suatu diagram yang menggambarkan secara menyeluruh dari ruang lingkup sistem. *Context diagram* tersebut merupakan *level* tertinggi dari DFD yang menggambarkan seluruh *input* ke sistem, *output* dari sistem dan memberi gambaran tentang

keseluruhan sistem. *Context diagram* untuk sistem perangkat lunak dapat dilihat pada Gambar 3.



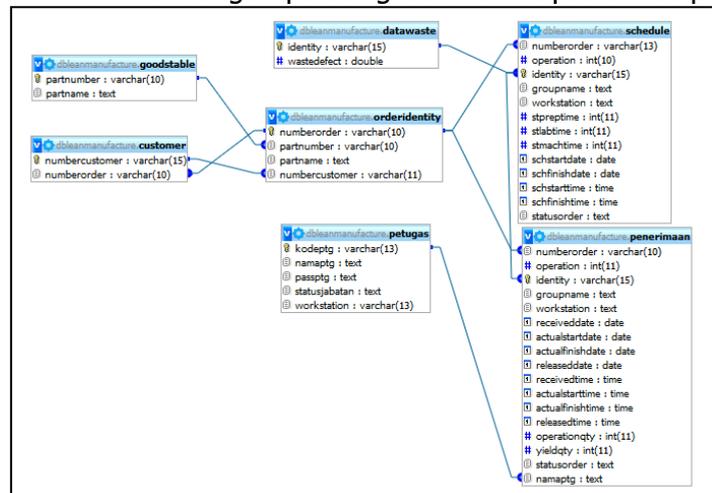
Gambar 3. Context Diagram

4.8 Formulasi Perhitungan Waste

Sub bab ini berisi tentang algoritma perhitungan *waste* yang terdapat didalam perangkat lunak. Algoritma perhitungan terdiri dari algoritma perhitungan *waste transportation*, *waste inventory*, *waste overprocess*, *waste delay*, dan *waste defect*.

4.10 ERD (Entity Relationship Diagram)

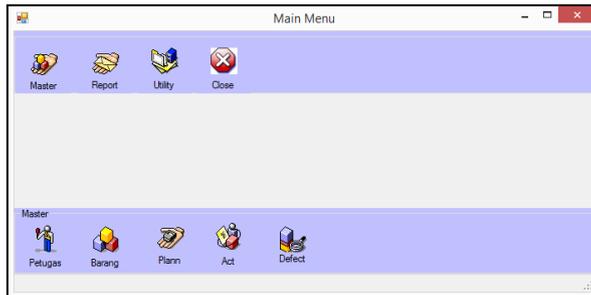
ERD (*Entity Relationship Diagram*) digunakan untuk mendokumentasikan data perusahaan dengan mengidentifikasi jenis entitas dan hubungannya. ERD menggambarkan hubungan antara satu entitas yang memiliki sejumlah atribut dengan entitas lainnya dalam suatu sistem terintegrasi. ERD dari rancangan per perangkat lunak dapat dilihat pada Gambar 4.



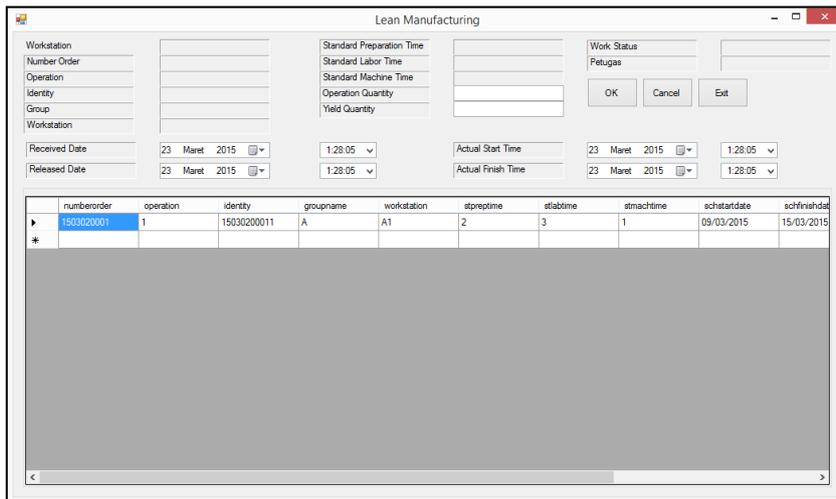
Gambar 4. ERD (Entity Relationship Diagram)

4.12 Perancangan Interface Perangkat Lunak

Rancangan *interface* perangkat lunak di rancang dengan menggunakan *software* Visual Studio 2008. Rancangan *interface* perangkat lunak ini terdiri dari *interface login*, sampai dengan *interface* untuk *report* dari *database* yang telah dibuat sebelumnya. Gambar 5 dan 6 berikut ini merupakan *interface* yang terdapat pada perangkat lunak.



Gambar 5. Menu Utama Master

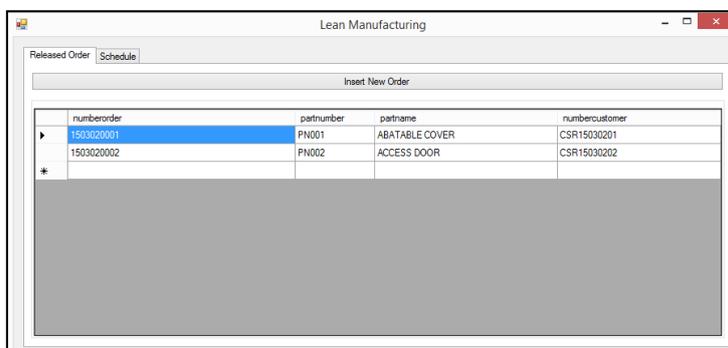


Gambar 6. Tampilan Doer

5. PENGUJIAN PERANGKAT LUNAK DAN ANALISIS

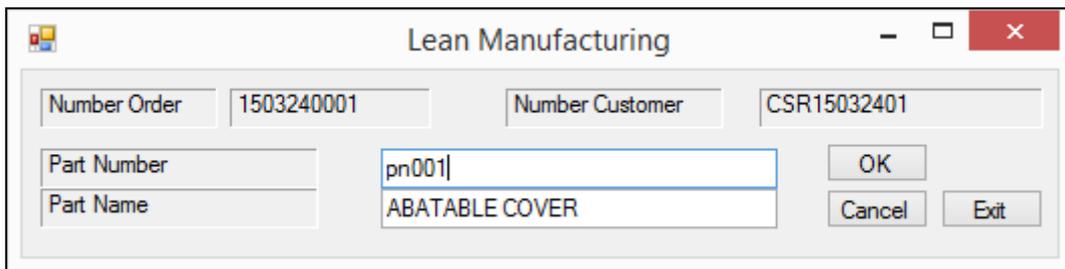
5.1 Pengujian Perangkat Lunak

Pada tahap ini dilakukan pengujian terhadap masing-masing modul yang sudah dirancang dalam menjalankan perannya masing-masing. Setiap modul memiliki tampilan yang berbeda dan terdapat beberapa skenario untuk menjalankan peran atau fungsi dari modul sesuai dengan kewenangan yang dimilikinya. Gambar 7, 8 dan 9 berikut ini merupakan contoh dari pengujian fungsi modul.

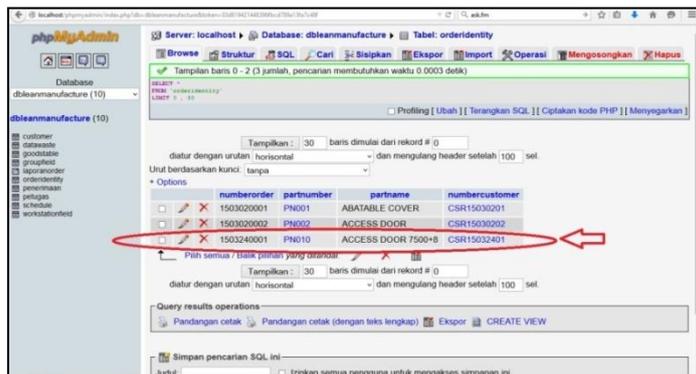


Gambar 7. Interface Released Order Untuk Planner

Mubarok, dkk.

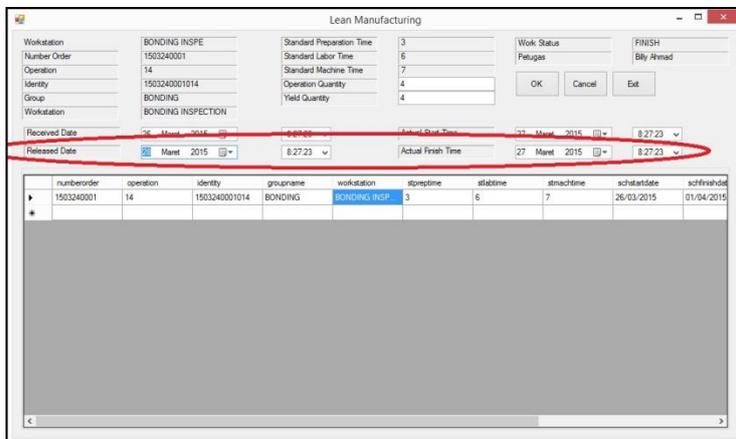


Gambar 8. Interface New Order

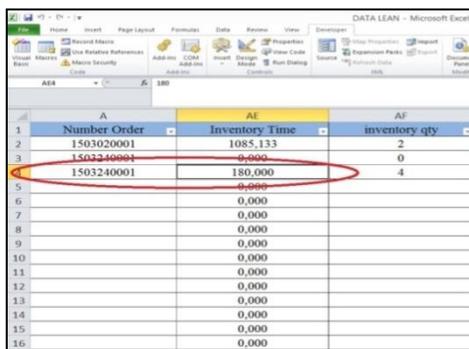


Gambar 9. Order Baru Masuk ke Database

Selain terdapat pengujian untuk masing-masing modul, terdapat juga pengujian untuk perhitungan yang dilakukan oleh perangkat lunak dengan nilai perhitungan manual. Gambar 10 dan 11 berikut merupakan contoh dari pengujian perhitungan perangkat lunak.



Gambar 10. Doer Memasukan Released dan Actual Finish



Gambar 11. Hasil Perhitungan Waste Inventory Time

Gambar 11 merupakan gambaran dari proses perhitungan *waste* yang dilakukan setelah data yang terdapat dalam *database* di ekspor ke Ms. Excel. Proses perhitungan dilakukan secara otomatis oleh perangkat lunak.

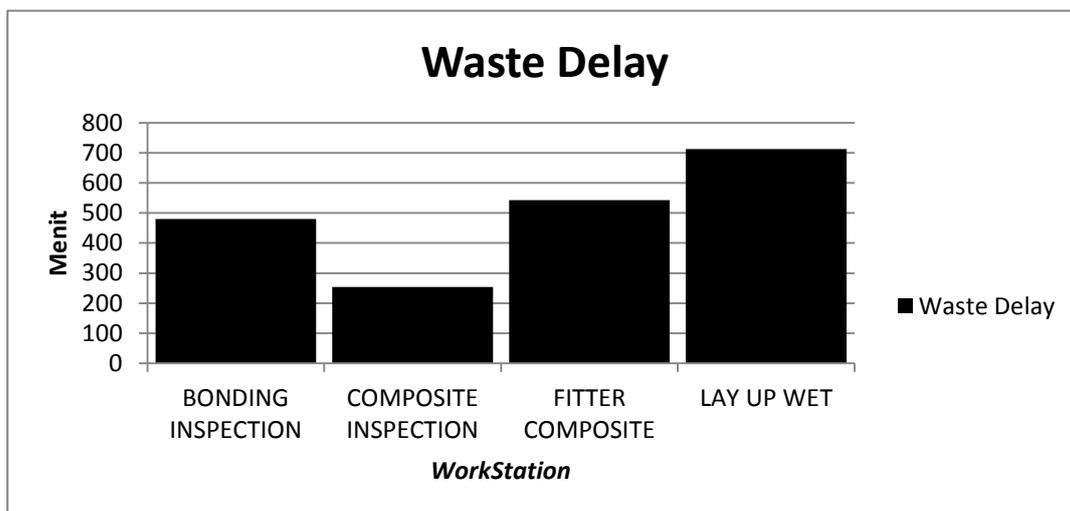
$$\begin{aligned}
 \text{Waste inventory time} &= \{ \text{jam pulang} - \text{actual finish time} + (\text{released time} - \text{jam masuk}) \} \\
 &\quad + \{ \text{jam kerja} * (\text{selisih hari} - 1) \} \\
 &= (17.00 - 14.30) + (08.30 - 08.00) + [8 \text{ jam} * \{ (1 \text{ hari}) - 1 \}] \\
 &= (2 \text{ jam } 30 \text{ menit}) + (30 \text{ menit}) + 0 \\
 &= 3 \text{ jam} \\
 &= 180 \text{ menit}
 \end{aligned}$$

Selain terdapat pengujian untuk masing-masing modul dan pengujian perhitungan, terdapat juga pengujian untuk hasil akhir atau *report*. Gambar 12 dan 13 berikut merupakan contoh dari *report* perhitungan perangkat lunak.

	A	B	C	D	E	F	G	H	I
	Workstation	Count	Defect Total (Unit)	Delay Total (Minute)	Transportation Total (Minute)	Inventory Time Total (Minute)	Inventory Quantity Total (Unit)	Over Process (Minute)	
1									
2		4095	0	0,000	0,000	0,000	0	0	
3	BONDING INSPECTION	1	2	480,4666667	167,2666667	180	4	0	
4	COMPOSITE INSPECTION	1	3	253,5166667	0	0	0	0	
5	FITTER COMPOSITE	1	0	542,1333333	1016,316667	120	5	0	
6	LAY UP WET	1	5	712,4166667	0	1085,133333	2	0	
7	Totals	4099	10	1988,533333	1183,583333	1385,133333	11	0	
8	Average		2,5	497,1333333	295,8958333	346,2833333	2,75	0	
9									
10									
11									
12									
13									
14									

Gambar 12. Hasil *Report* Rekapitulasi Data *Waste*

Gambar 12 merupakan hasil *report* yang dilakukan perangkat lunak setelah proses perhitungan telah dilakukan. Proses perekapan dihasilkan dengan menggunakan metode *data mining* dengan bantuan *tools-tools* dari Ms. Excel. Secara otomatis perangkat lunak akan melakukan rekap tergantung jumlah data yang ada berdasarkan stasiun kerja terhadap masing-masing jenis *waste*.



Gambar 13. Grafik *Waste Delay*

5.2 Analisis

Terdapat analisis mengenai fungsi perangkat lunak dan analisis mengenai penggunaan perangkat lunak, yaitu:

1. Sistem perangkat lunak dapat dikatakan valid untuk digunakan karena hasil yang diperoleh dari perhitungan perangkat lunak dengan hasil perhitungan secara manual tidak berbeda jauh, tetapi masih perlu dilakukan validasi lebih lanjut terhadap sistem untuk lebih memastikan tingkat keakuratan dari sistem perangkat lunak.
2. Proses *sorting* dan *filtering* lebih mudah karena sudah terotomasi oleh perangkat lunak
3. Rekapitulasi data *report* hanya membutuhkan waktu yang sangat singkat dibandingkan dengan rekapitulasi secara manual.
4. Data perhitungan *waste* lebih rinci dengan menampilkan data pemborosan dalam menit. Sedangkan dengan cara manual hanya menampilkan pemborosan dalam satuan hari.

6. KESIMPULAN

Berdasarkan hasil penelitian didapatkan beberapa kesimpulan sebagai berikut:

1. Penelitian menghasilkan prototipe perangkat lunak dengan menerapkan *lean manufacturing* untuk mengidentifikasi pemborosan yang terdapat pada *area bonding composite* PT Dirgantara Indonesia yang memiliki sistem *job order*.
2. Prototipe dikhususkan untuk mengidentifikasi 5 jenis pemborosan, yaitu *defect, delays, overprocess, transportation, dan inventory*.
3. Prototipe perangkat lunak ini dapat mempermudah perusahaan untuk melakukan pengolahan data untuk mengidentifikasi pemborosan serta membantu dalam mengidentifikasi stasiun kerja yang memiliki pemborosan. Prototipe mampu melakukan perekapan data dalam waktu yang singkat dan menampilkan hasil identifikasi pemborosan dalam bentuk diagram batang berdasarkan masing-masing stasiun kerja dan pemborosan.
4. Dengan adanya prototipe perusahaan dapat lebih *up to date* terhadap data pemborosan yang terjadi di stasiun kerja sehingga dapat lebih cepat dalam menangani masalah pemborosan yang ada.

REFERENSI

Gaspersz, Vincent. 2007. *Lean Six Sigma For Manufacturing And Service Industries*. PT Gramedia Pustaka Utama, Jakarta.

Haridi, Saif. "7. Modules and Interfaces". 31 Maret 2015. <http://www.mozart-oz.org/documentation/index.html>

Ladjamudin. 2005. *Analisis Dan Desain Sistem Informasi*. Graha Ilmu. Yogyakarta.

Microsoft. 2009. *Microsoft Application Architecture Guide, 2nd Edition*. Microsoft Pattern & Practice Developer. USA.

Phil, Jones. 2001. *Visual Basic: A Complete Course Letts Higher Education List Series*. Cengage Learning EMEA

Raharjo, Budi. 2011 *Belajar Otodidak Membuat Database Menggunakan MySQL*. IF INFORMATIKA. Bandung.

Yakub. 2012. *Pengantar Sistem Informasi*. Graha Ilmu. Yogyakarta.