

Pengembangan Aplikasi *Trucking Haulage* untuk Meningkatkan Produktivitas Pengangkutan Bongkar Kontainer di PT. X Surabaya

Claudia Febby Siemawijaya¹, Iwan Halim Sahputra²

Abstract: Logistics is a business that will continue to grow, because it is needed in the business world to deliver goods at the right time and quantity, and at low costs. PT. X is a logistics and shipping company. One of the activities that run in the logistics and shipping business is container haulage. The haulage process is considered still less effective and efficient. The current level of productivity as measured by the number of ritase per day is low. This is caused by the haulage process that can not be detected in real time, impractical communication media, queues, and unavailability of trucks. The trucking haulage application is created and developed so non-value added activities will be minimized or eliminated, so the haulage productivity can be increased. This study discusses the development of a trucking haulage application in phase 1.1. The implementation result of the application shows that the haulage productivity increased from 7,36 ritases per day to 7,93 ritases per day. Two sample t-test shows that the p-value obtained is 0,033 ($< 0,05$), so it can be concluded that the trucking haulage application can increase the productivity of container discharge at PT. X Surabaya.

Keywords: logistic and shipping; application; haulage; productivity; ritase

Pendahuluan

PT. X adalah perusahaan yang bergerak di bidang logistik dan pelayaran. Alat transportasi yang digunakan dalam logistik dan pelayaran untuk menjalankan proses pengangkutan adalah truk, dimana truk digunakan untuk mengangkut kontainer dari pelabuhan dan dibawa menuju ke depo untuk dibongkar. Namun, proses pengangkutan kurang efektif dan efisien. Tingkat produktivitas pengangkutan saat ini tergolong rendah, yaitu 7,36 ritase per hari. Proses pengangkutan seringkali selesai melebihi standar waktu proses (1 jam), karena tidak bisa dideteksi secara *real time*. Terdapat 31,43% aktivitas diselesaikan lebih dari 1 jam. Media komunikasi yang digunakan adalah *Whatsapp* (WA) dan HT. Adanya penyaluran informasi dari satu entitas ke entitas lain menyebabkan adanya *non-value added activity*. Selain itu, *dispatcher* tidak mengetahui kondisi pelabuhan. Jika pelabuhan ramai, *driver* harus antri. Hal ini juga menyebabkan ketidakseimbangan jumlah pesanan dan ketersediaan truk. Terkadang

tidak ada truk yang tersedia saat ada pesanan yang harus dijalankan. Beberapa permasalahan tersebut, menyebabkan rendahnya tingkat produktivitas pengangkutan kontainer, sehingga perusahaan memiliki ide untuk mengembangkan aplikasi *trucking haulage* yang bertujuan untuk meminimalkan atau mengeliminasi aktivitas-aktivitas pengangkutan yang tidak memberikan nilai tambah.

Metode Penelitian

Metode yang digunakan dalam penelitian ini adalah *Lean Thinking Model*. *Lean Thinking Model* adalah metode untuk meminimalkan sumber daya melalui kegiatan perbaikan yang dilakukan secara berkesinambungan dan menitikberatkan pada identifikasi, serta eliminasi aktivitas yang tidak memberikan nilai tambah pada proses kerja (Nur [1]). Urutan fase *Lean Thinking Model* adalah sebagai berikut.

Mendefinisikan Nilai

Mendefinisikan nilai (*define value*) adalah tahap menentukan sesuatu yang dapat memberikan nilai dari suatu jasa atau produk (Syawalluddin [2]). Pada identifikasi masalah dan studi literatur sudah digali

^{1,2} Fakultas Teknologi Industri, Program Studi Teknik Industri, Universitas Kristen Petra. Jl. Siwalankerto 121-131, Surabaya 60236. Email: claudiafebby9@gmail.com, iwanh@petra.ac.id

masalah-masalah dan teori-teori pendukung yang akan memberikan nilai tambah pada proses pengangkutan kontainer.

Memetakan Proses Awal

Memetakan proses awal (*map value stream*) adalah gambaran dari proses yang terjadi saat ini. Fase ini menggambarkan *business process* awal aktivitas pengangkutan bongkar. *Business process* awal akan dianalisa untuk mengetahui aktivitas-aktivitas yang tidak memberikan nilai tambah dan rencana perbaikan yang dapat dilakukan.

Membuat Alur Proses Perbaikan

Membuat alur proses perbaikan (*create flow*) adalah gambaran sederhana dari alur proses yang akan dibuat. Berdasarkan perencanaan perbaikan, dilakukan perencanaan pengembangan untuk aplikasi *trucking haulage* yang sekaligus menjadi MVP aplikasi. *Business process* perbaikan dibuat untuk menggambarkan proses pengangkutan dengan menggunakan aplikasi.

Merencanakan Sistem

Merencanakan sistem (*establish pull*) adalah gambaran sistem yang akan dikembangkan. Fase ini meliputi perencanaan kebutuhan teknis yang menjelaskan kebutuhan dari sudut pandang perusahaan (*business requirements*), pengguna (*user requirements*), dan fungsi aplikasi (*functional requirements*), serta perencanaan *data flow*, perencanaan akuisisi data, dan standarisasi proses dan data.

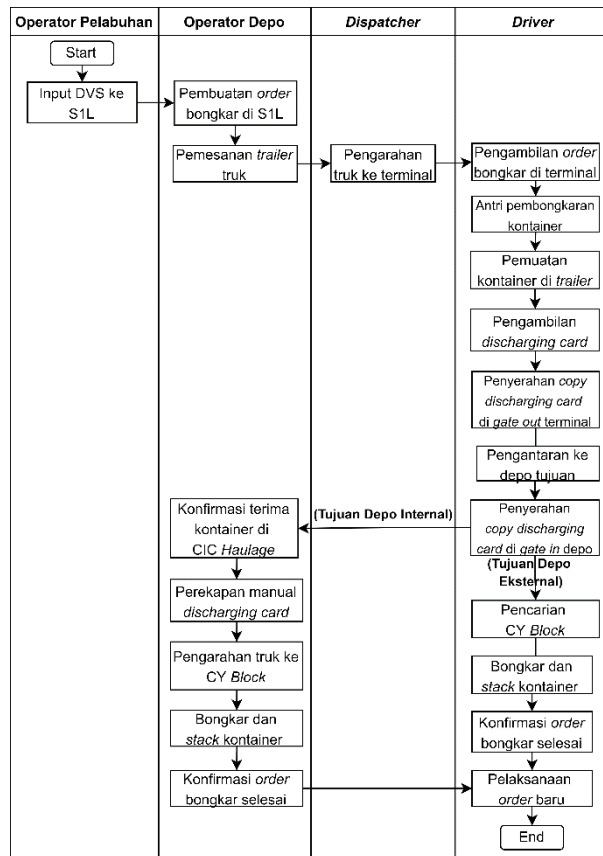
Mencapai Hasil

Mencapai hasil (*persuit perfection*) adalah tahap pembuatan produk dan evaluasi yang nantinya akan diimplementasi untuk mengatasi permasalahan yang ada. Fase ini meliputi perencanaan prototipe dengan menggunakan *Lean Startup Model*. Prototipe aplikasi akan diuji hingga semua fungsinya dapat berjalan dengan baik dan kemudian diimplementasi.

Hasil dan Pembahasan

Business Process Awal

Business process awal dibuat untuk aktivitas bongkar kontainer. *Business process* awal bertujuan untuk mengetahui proses dari aktivitas bongkar saat belum menggunakan aplikasi *trucking haulage*. Aktivitas pengangkutan bongkar melibatkan 4 entitas, yaitu: operator pelabuhan, operator depo, dispatcher, dan driver.



Gambar 1. *Business process* awal bongkar

Operator pelabuhan memasukkan *Daily Vessel Schedule* (DVS) ke dalam S1L (sistem informasi perusahaan yang berisi seluruh data operasional perusahaan). Operator depo membuat pesanan bongkar di S1L dan melakukan pemesanan *trailer* menggunakan HT atau WA. *Dispatcher* memberikan arahan ke *driver* untuk menuju pelabuhan juga melalui HT atau WA. Kemudian, *driver* menuju pelabuhan untuk mengambil pesanan. *Driver* antri dan mengambil kontainer yang dituju. *Driver* akan mengambil *discharging card*. *Driver* dapat melihat informasi pesanan, termasuk lokasi awal dan lokasi tujuan pada *discharging card*. Kemudian, *driver* menyerahkan *copy discharging card* di *gate out* pelabuhan.

Driver mengantarkan kontainer ke depo dan menyerahkan *copy discharging card* di *gate in depo*. Depo tujuan terbagi menjadi 2 jenis, yaitu: depo internal (milik perusahaan) dan depo eksternal (bukan milik perusahaan). Jika tujuan depo internal, operator depo akan melakukan konfirmasi terima kontainer di *CIC Haulage* (portal perusahaan yang digunakan untuk operasional kegiatan pengangkutan) dan merekap *discharging card* secara manual di sebuah form. Lalu, operator depo mengarahkan *driver* ke *CY Block*. Operator depo melakukan bongkar kontainer dan *stack* barang-barang tersebut. Operator depo konfirmasi *order*

bongkar selesai melalui HT atau WA dan *driver* akan menjalankan pesanan baru. Jika tujuan depo eksternal, *driver* akan langsung mencari *CY Block* sesuai yang tertera di *discharging card*. Kemudian, kontainer dibongkar dan *distack*. *Driver* melakukan konfirmasi *order* bongkar melalui HT atau WA dan pesanan baru akan dijalankan.

Business process tersebut dianalisa dan dibuat rencana perbaikannya. Analisa pertama adalah proses pemesanan *trailer* truk dan pengarahannya ke terminal. Penggunaan WA membutuhkan waktu untuk operator mencari aplikasi WA di telepon genggamnya, mencari nama kontak yang dituju, dan mengetik isi pesan, dimana dibutuhkan perhatian lebih karena isi pesan mencakup data-data pesanan yang berisi angka dan kode, lalu pesan tersebut baru dikirimkan. Tidak menutup kemungkinan terjadi kesalahan ketik dan adanya gangguan koneksi. Penggunaan HT juga membutuhkan waktu untuk operator menyalakan HT tersebut, mencari saluran yang dituju, juga memungkinkan terjadinya kesalahan pendengaran, karena ketidakjelasan dari pelafalan yang diucapkan dan kondisi lingkungan kerja tidak menentu. *Dispatcher* harus menunggu pesan tersebut, baru dapat mengarahkan *driver* untuk memulai pekerjaan. Proses tersebut menyebabkan adanya *waste of waiting*. Adanya penyaluran informasi dari satu entitas ke entitas lain juga menyebabkan *waste of overprocessing*. Rencana perbaikan yang dilakukan adalah pembuatan *order* bongkar di SIL dapat langsung dilihat oleh *driver*, sehingga *driver* dapat langsung menjalankan pesanan.

Selain itu, operator depo memesan truk sesuai dengan pesanan yang sudah dibuat pada hari tersebut. *Dispatcher* mengikuti pesanan tersebut dan langsung mengarahkan *driver* menuju pelabuhan. *Dispatcher* tidak memperhitungkan jumlah pesanan yang ada dan jumlah truk yang tersedia. Hal tersebut menyebabkan terkadang tidak ada truk yang tersedia saat ada pesanan yang harus dijalankan, sehingga pesanan tidak dapat diproses. Rencana perbaikan yang dilakukan adalah *dispatcher* melakukan perencanaan pengangkutan harian.

Analisa kedua adalah proses pengambilan *order* bongkar. Terdapat beberapa kejadian, dimana waktu proses aktual melebihi waktu proses standar. Hal tersebut disebabkan karena aktivitas pengangkutan tidak dapat dideteksi secara *real time*. Perusahaan tidak mengetahui keadaan dan kondisi yang dialami oleh *driver* selama perjalanan, seperti: kondisi jalan ramai, terjadi kecelakaan, truk mogok, atau *driver* yang sengaja tidak menjalankan pekerjaan dengan baik. Rencana perbaikan yang dilakukan adalah

driver melakukan konfirmasi saat pesanan bongkar diambil dan saat kontainer selesai dikerjakan. *Driver* juga dapat melihat peta perjalanan (*Google Maps*) untuk melihat keadaan jalan dan memperkirakan waktu perjalanan, sehingga *dispatcher* juga bisa memantau aktivitas *driver* secara *real time*.

Analisa ketiga adalah proses antri pembongkaran kontainer. *Dispatcher* tidak bisa memantau keadaan di pelabuhan. Jika antrian di terminal panjang, *driver* harus menunggu antrian. Hal tersebut menyebabkan waktu proses bertambah dan *waste of waiting*. Rencana perbaikan yang dilakukan adalah terdapat *dispatcher* yang bertugas untuk memonitor antrian di pelabuhan, sehingga *dispatcher* dapat memutuskan apakah *driver* dapat berangkat ke lokasi atau tidak. Analisa keempat adalah proses konfirmasi terima kontainer di *CIC Haulage* dan analisa kelima adalah proses perekapan manual *discharging card*. Setelah konfirmasi terima kontainer di *CIC Haulage*, operator depo melakukan perekapan *discharging card*. Operator memasukkan data yang sama pada dua proses tersebut. *CIC Haulage* sering mengalami *error*, sehingga perekapan manual digunakan sebagai data cadangan. Hal tersebut menyebabkan adanya *waste of overprocessing*. Rencana perbaikan yang dilakukan adalah *CIC Haulage* tidak digunakan lagi. Konfirmasi terima kontainer dilakukan melalui SSaya (aplikasi perusahaan yang secara aliran data hampir sama dengan *CIC Haulage*, namun memiliki fitur yang lebih lengkap dan lebih mudah diaplikasikan di lapangan). Perekapan *discharging card* juga tidak perlu dilakukan lagi.

Analisa keenam adalah proses konfirmasi *order* bongkar selesai. Konfirmasi *order* bongkar selesai dilakukan melalui HT atau WA, sehingga menyebabkan adanya *waste of waiting*, *waste of overprocessing*, dan memungkinkan adanya kesalahan ketik dan pendengaran. Rencana perbaikan yang dilakukan adalah untuk tujuan depo internal, konfirmasi terima kontainer di SSaya sekaligus menjadi konfirmasi *order* selesai.

Perencanaan Pengembangan Aplikasi

Penelitian membahas pengembangan fase 1.1, sehingga tidak semua rencana perbaikan dilakukan. Perencanaan pengembangan terbagi menjadi pengembangan di dalam *business process* dan di luar *business process*. Pengembangan di dalam *business process* adalah pengembangan yang berhubungan langsung dengan *business process*. Pengembangan di luar *business process* adalah pengembangan yang dilakukan terhadap hal-hal lain yang masih berhubungan dengan jalannya aplikasi.

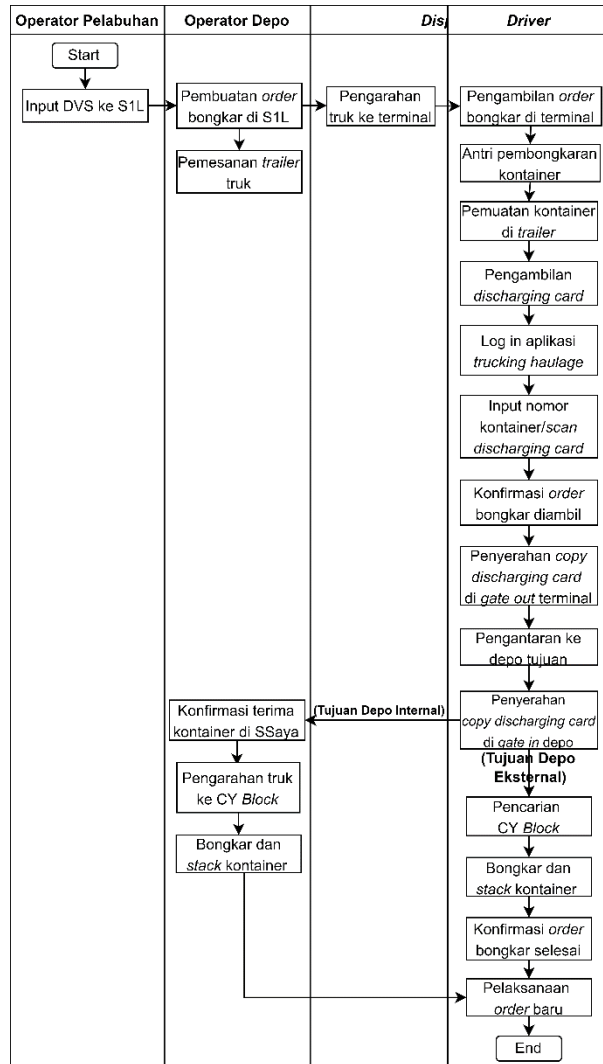
Terdapat 4 rencana pengembangan di dalam *business process*. Rencana pengembangan pertama adalah proses pembuatan *order* bongkar di S1L, dimana S1L akan diintegrasikan dengan aplikasi. Rencana pengembangan kedua adalah pengambilan *order* bongkar. *Driver* dapat *log in* aplikasi dengan memasukkan nomor kontainer atau *scan discharging card*. Setelah itu, *driver* konfirmasi *order* diambil dengan memperbarui status pesanan di aplikasi menjadi '*pick up*'. Rencana pengembangan ketiga adalah konfirmasi terima kontainer. Operator depo melakukan konfirmasi terima kontainer melalui SSaya yang juga diintegrasikan dengan aplikasi. Rencana pengembangan keempat adalah konfirmasi *order* bongkar selesai. Jika tujuan depo internal, konfirmasi terima kontainer di SSaya akan memperbarui status pesanan di aplikasi menjadi '*finish*' secara otomatis. Jika tujuan depo eksternal, *driver* melakukan konfirmasi dengan memperbarui status pesanan di aplikasi menjadi '*finish*'.

Terdapat 4 rencana pengembangan di luar *business process*. Rencana pengembangan pertama adalah setiap *driver* terdaftar di aplikasi *trucking haulage*. *Dispatcher* harus membuat akun untuk setiap *driver* melalui *myBackend* (portal perusahaan yang berisi data-data dari seluruh entitas yang terlibat di dalam proses bisnis). Rencana pengembangan kedua adalah nomor kontainer yang sudah dimasukkan oleh satu *driver* tidak dapat dimasukkan oleh *driver* lain untuk menghindari dobel pesanan.

Saat pembuatan *order* di S1L, operator depo memasukkan lokasi depo tujuan. Lokasi depo tujuan juga digunakan untuk membedakan jenis konfirmasi penyelesaian *order* di akhir proses. Rencana pengembangan ketiga adalah *dispatcher* perlu memperbarui daftar depo pada *Master Container Yard* di *CIC Haulage*, sehingga aplikasi dapat berjalan sesuai dengan fungsinya. Rencana pengembangan keempat adalah *driver* dapat mengganti nomor lambung sesuai dengan truk yang digunakan. *Dispatcher* harus memperbarui *Master Truck* di *CIC Haulage*, sehingga semua nomor lambung truk yang ada dapat muncul di aplikasi dan *driver* dapat menemukan nomor lambung tersebut.

Business Process Perbaikan

Business process perbaikan bertujuan untuk mengetahui proses dari aktivitas bongkar setelah menggunakan aplikasi *trucking haulage*. Aktivitas bongkar melibatkan 4 entitas, yaitu: operator pelabuhan, operator depo, *dispatcher*, dan *driver*. *Business process* perbaikan dapat dilihat pada Gambar 2.



Gambar 2. *Business process* perbaikan bongkar

Perbedaan pada *business process* awal dan *business process* perbaikan adalah pembuatan *order* bongkar di S1L, dimana S1L akan diintegrasikan dengan aplikasi. Setelah *driver* mengambil *discharging card*, *driver* akan *log in* aplikasi, kemudian memasukkan nomor kontainer atau *scan discharging card*. *Driver* konfirmasi *order* bongkar diambil dengan memperbarui status pesanan menjadi '*pick up*' di aplikasi. Setelah sampai di depo, jika tujuan depo internal, operator depo konfirmasi terima kontainer melalui SSaya, dimana SSaya juga akan diintegrasikan dengan aplikasi. Konfirmasi terima kontainer ini akan memperbarui status pesanan di aplikasi menjadi '*finish*' secara otomatis. Jika tujuan depo eksternal, *driver* konfirmasi *order* selesai dengan memperbarui status pesanan menjadi '*finish*' di aplikasi.

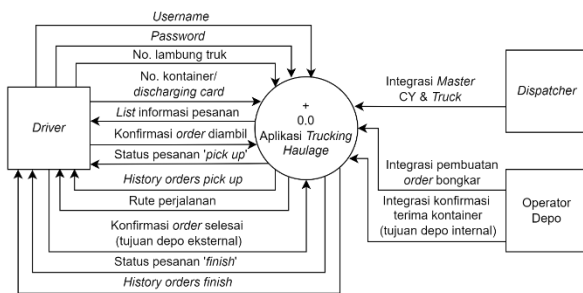
Perencanaan Kebutuhan Teknis

Business requirements yaitu meminimalkan atau mengeliminasi aktivitas yang tidak memberikan nilai tambah, sehingga dapat meningkatkan

produktivitas ritase. *User requirements* yaitu menjalankan pesanan bongkar lebih mudah dan praktis. *Functional requirements* meliputi beberapa menu di dalam aplikasi, yaitu: cari kontainer, *scan QR Code*, *combo*, kontainer lain, *history*, dan profil. Aplikasi juga terintegrasi dengan S1L, SSaya, dan *CIC Haulage*. Terdapat juga hal-hal lain, seperti: setiap *driver* harus memiliki akun yang terdaftar untuk bisa *log in* ke aplikasi, tampilan *list* informasi terkait kontainer yang bersangkutan, tombol '*pick up*' dan '*finish*' untuk memperbarui status pesanan, tampilan *Google Maps* saat pekerjaan berjalan, dan nomor kontainer yang sudah dimasukkan oleh seorang *driver* tidak dapat dimasukkan oleh *driver* lain.

Data Flow Diagram

Data Flow Diagram (DFD) bertujuan untuk menunjukkan aliran data yang terjadi di dalam aplikasi. Terdapat 3 entitas yang terlibat, yaitu: *driver*, operator depo, dan *dispatcher*.



Gambar 3. *Data flow diagram*

Terdapat integrasi *Master CY* dan *Truck CIC Haulage* dari *dispatcher*, serta pembuatan pesanan S1L dari operator depo. *Driver log in* aplikasi dengan memasukkan *username* dan *password*. *Driver* juga dapat mengganti nomor

lambung truk. Kemudian, *driver* menjalankan pesanan dan memasukkan nomor kontainer atau *scan discharging card*. Aplikasi akan memunculkan *list* informasi pesanan. *Driver* melanjutkan pesanan dengan melakukan konfirmasi bahwa pesanan diambil. Aplikasi mengubah status pesanan menjadi '*pick up*' dan nomor kontainer tersimpan di *history orders pick up*. Aplikasi akan menampilkan rute yang ditempuh *driver* selama perjalanan. Jika tujuan depo internal, konfirmasi terima kontainer di SSaya dari operator depo akan menjadi pemicu pada status pesanan, sehingga status pesanan berubah menjadi '*finish*'. Jika tujuan depo eksternal, *driver* melakukan konfirmasi *order* bongkar selesai. Aplikasi akan memunculkan status pesanan '*finish*' dan nomor kontainer tersimpan di *history orders finish*.

Perencanaan Akuisisi Data

Perencanaan akuisisi data bertujuan untuk mengetahui data-data yang dibutuhkan aplikasi, beserta dengan sumber datanya. Perencanaan akuisisi data dapat dilihat pada Tabel 1. Pembuatan *order* bongkar membutuhkan integrasi data *Master CY* dan *truck* dari *CIC Haulage* dan data pesanan bongkar dari S1L. *Log in* aplikasi membutuhkan data *username*, *password*, dan nomor lambung truk dari *driver* yang bersangkutan. Pengambilan kontainer membutuhkan data nomor kontainer/*discharging card* dan konfirmasi *order* diambil yang akan dimasukkan oleh *driver* ke aplikasi. *List* informasi pesanan, status pesanan '*pick up*', dan *history orders pick up* akan diproses oleh aplikasi. Pengiriman kontainer membutuhkan data rute perjalanan yang akan diproses oleh aplikasi. Konfirmasi kontainer diterima membutuhkan integrasi konfirmasi terima kontainer dari SSaya. Penyelesaian *job order* membutuhkan data konfirmasi *order* selesai dari *driver*, serta status

Tabel 1. Perencanaan akuisisi data

No.	Proses	Data yang Dibutuhkan	Sumber Data
1.	Pembuatan <i>order</i> bongkar	Integrasi <i>Master CY</i> dan <i>Truck</i> Integrasi pembuatan <i>order</i> bongkar	<i>Data Entry</i> <i>Data Entry</i>
2.	<i>Log in</i> aplikasi	<i>Username</i> <i>Password</i> Nomor lambung truk	<i>Data Entry</i> <i>Data Entry</i> <i>Data Entry</i>
3.	Pengambilan kontainer	Nomor kontainer/ <i>discharging card</i> <i>List</i> informasi pesanan Konfirmasi <i>order</i> diambil Status pesanan ' <i>pick up</i> ' <i>History orders pick up</i>	<i>Data Entry</i> <i>Trucking Haulage</i> <i>Data Entry</i> <i>Trucking Haulage</i> <i>Trucking Haulage</i>
4.	Pengiriman kontainer	Rute perjalanan	<i>Trucking Haulage</i>
5.	Konfirmasi kontainer diterima	Integrasi konfirmasi terima kontainer (tujuan depo internal)	<i>Data Entry</i>
6.	Penyelesaian <i>job order</i>	Konfirmasi <i>order</i> selesai (tujuan depo eksternal) Status pesanan ' <i>finish</i> ' <i>History orders finish</i>	<i>Data Entry</i> <i>Trucking Haulage</i> <i>Trucking Haulage</i>

pesanan *'finish'* dan *history orders finish* yang akan diproses oleh aplikasi.

Standarisasi Proses dan Data

Aplikasi *trucking haulage* digunakan untuk proses pengangkutan di Surabaya dan Jakarta. Standarisasi proses dan data diperlukan untuk menyamakan proses dan data dari kedua kota tersebut.

Tabel 2. Standarisasi proses dan data

Deskripsi	Kontainer	Status Before	Status After	Nama Aktivitas
TL Bongkar Full – Stripping Luar	Full	FOB	FXD	Bongkar
TL Bongkar Full – Stripping Dalam	Full	FOB	FXD	Bongkar
TL Bongkar MTA	Empty	MTB	MTA	Bongkar
TL Bongkar Full - Ekspor	Full	FOB	FIT	Bongkar

Aktivitas *Truck Lossing* (TL) bongkar *full stripping* luar adalah aktivitas bongkar kontainer berisi yang berasal dari pelabuhan dengan status *Full on Board* (FOB). Kontainer ini dibawa ke depo dan akan dibongkar oleh penerima di gudangnya sendiri, sehingga status kontainer menjadi *Full at Discharge* (FXD). Aktivitas TL bongkar *stripping* dalam adalah aktivitas bongkar kontainer berisi yang berasal dari pelabuhan dengan status FOB untuk dibawa menuju ke depo. Kontainer ini akan dibongkar di dalam depo, sehingga status kontainer menjadi FXD. Aktivitas TL bongkar *Empty Available* (MTA) adalah aktivitas bongkar kontainer kosong yang berasal dari pelabuhan dengan status *Empty on Board* (MOB). Kontainer ini dibawa menuju ke depo yang membutuhkan, sehingga status kontainer menjadi MTA. Aktivitas TL bongkar *full ekspor* adalah aktivitas bongkar kontainer berisi yang berasal dari pelabuhan dengan status FOB untuk dibawa menuju ke depo untuk transit. Transit ini dibutuhkan, karena kontainer memiliki tujuan internasional, sehingga status kontainer menjadi *Full in Transit* (FIT). Keempat aktivitas tersebut akan digolongkan ke dalam jenis aktivitas bongkar.

Perencanaan Prototipe

Perencanaan prototipe dari aplikasi *trucking haulage* menggunakan pendekatan *lean startup model*. Terdapat 3 fase dalam *lean startup model*. Fase pertama adalah *build*. Berdasarkan beberapa permasalahan yang ada, perusahaan

akan mengembangkan aplikasi *trucking haulage* seperti pada perencanaan pengembangan yang sudah dijelaskan. Aplikasi bertujuan untuk meminimalkan atau mengeliminasi aktivitas-aktivitas pengangkutan bongkar yang tidak memberikan nilai tambah. Fase kedua adalah *measure*. Aplikasi ditunjukkan dalam bentuk prototipe. Prototipe akan diuji beberapa kali untuk mencoba semua fitur yang ada. Fase ketiga adalah *learn*. Hasil evaluasi pengujian prototipe akan diperbaiki hingga semua fungsi aplikasi berjalan dengan baik dan memenuhi MVP yang sudah ditentukan. Setelah prototipe memenuhi MVP, aplikasi diimplementasi pada proses pengangkutan bongkar dan dikumpulkan data ritase selama beberapa waktu untuk perhitungan produktivitas pengangkutan.

Gambaran Ide Aplikasi

Sebelum *driver* bisa menjalankan pekerjaan dengan aplikasi, operator depo membuat pesanan bongkar pada SIL. Pertama, *driver* masuk aplikasi dan menjumpai halaman *log in*. *Driver* harus memasukkan *username* dan *password*. *Driver* masuk ke halaman utama aplikasi dan mendapatkan notifikasi yang menginformasikan bahwa lokasi *driver* akan dicatat ke dalam sistem, walaupun aplikasi sedang tidak digunakan. Setelah itu, *driver* dapat mulai menggunakan aplikasi untuk menjalankan pesanan. Pada halaman utama, terdapat beberapa pilihan menu, yaitu: cari kontainer, *scan QR Code*, *history*, kontainer lain, *combo*, dan profil. *Driver* dapat memasukkan nomor kontainer pada bagian cari kontainer atau melakukan *scan QR Code* yang terdapat pada *discharging card*. Setelah nomor kontainer tertera di aplikasi, *driver* menekan tombol cari. Aplikasi menampilkan informasi pesanan berupa nomor kontainer, kode kapal, status kontainer, pengirim, dan status pesanan. Status pesanan yang tertera masih kosong, sehingga *driver* harus menekan tombol lanjut *pick up*. Aplikasi akan memunculkan notifikasi untuk memastikan bahwa *driver* melakukan pembaharuan status pesanan. *Driver* menekan tombol OK dan status pesanan berubah menjadi *pick up*.

Jika tujuan depo internal, operator depo melakukan konfirmasi terima kontainer melalui SSaya. Operator depo memasukkan nomor kontainer yang bersangkutan, lalu menekan tombol cari. SSaya menampilkan informasi kontainer. Jika informasi sudah benar, operator depo dapat menekan tombol *confirm* terima. SSaya akan menampilkan notifikasi untuk

memastikan bahwa operator depo melakukan konfirmasi terima kontainer. Setelah itu, SSaya menampilkan tulisan kontainer berhasil diterima dan status pesanan di aplikasi berubah menjadi *finish* secara otomatis. Jika tujuan depo eksternal, *driver* dapat menekan tombol lanjut *finish* di aplikasi. Aplikasi akan menampilkan notifikasi untuk memastikan bahwa *driver* melakukan pembaharuan status pesanan. *Driver* menekan tombol OK dan status pesanan berubah menjadi *finish*. *Driver* dapat menekan tombol kontainer lain untuk kembali ke halaman utama.

Menu combo digunakan untuk menjalankan pesanan dengan jumlah kontainer 2 buah. *Driver* dapat langsung memasukkan kedua nomor kontainer pada bagian cari kontainer atau melakukan *scan barcode* pada *discharging card* satu per satu. Setelah kedua nomor kontainer tertera di aplikasi, *driver* menekan tombol cari. Aplikasi menampilkan informasi pesanan dari kedua kontainer tersebut. *Driver* memilih kontainer pertama dan menekan tombol lanjut *pick up*, dilanjutkan dengan memilih kontainer kedua dan menekan tombol lanjut *pick up*. Status pesanan kedua kontainer berhasil diperbarui. Perubahan status pesanan menjadi *finish* juga dilakukan dengan cara yang sama, baik untuk tujuan depo internal maupun eksternal.

Menu *history* digunakan untuk melihat daftar kontainer yang sedang dikerjakan dan sudah selesai dikerjakan. Jika *driver* sudah melakukan perubahan status pesanan menjadi *pick up* dan tidak sengaja keluar dari aplikasi, *driver* dapat membuka menu *history* dan memilih nomor kontainer tersebut, lalu menekan tombol cari. Aplikasi akan menampilkan halaman terakhir sebelum *driver* keluar dari aplikasi. *Driver* dapat melanjutkan pekerjaannya hingga kegiatan kontainer selesai.

Namun, jika *driver* memilih kontainer dengan status pesanan *finish* pada menu *history*, aplikasi akan menampilkan notifikasi bahwa container tidak terdapat pada *list* bongkaran. Pada menu profil, terdapat beberapa pilihan, antara lain: *edit* nomor lambung, kata sandi, PIN, notifikasi, hapus riwayat kontainer, sertifikasi vaksin Covid-19, dan keluar. Jika *driver* melanjutkan pesanan dari *driver* sebelumnya, *driver* dapat mengubah nomor lambung pada akun aplikasinya dengan klik tombol *edit* yang terdapat di bawah tulisan nomor lambung dan nomor polisi.

Evaluasi Hasil Uji Aplikasi

Pengujian prorotipe dilakukan sebanyak 3 kali. Terdapat 5 evaluasi pada pengujian pertama. Evaluasi pertama terjadi pada tampilan notifikasi saat *driver* mengambil pesanan bongkar. Notifikasi yang muncul adalah 'Status SOPT/Container berhasil di *udpate*'. Tulisan 'SOPT/Container' dapat membingungkan *driver*, karena SOPT adalah istilah untuk aktivitas *dooring*. Notifikasi harus diperbaiki menjadi 'Status Container berhasil di *update*'. Evaluasi kedua terjadi pada menu *history* kontainer, dimana daftar riwayat nomor kontainer yang sudah *finish* tidak muncul di aplikasi. Perbaikan yang dilakukan adalah nomor kontainer yang sudah *finish* muncul pada menu *history*.

Evaluasi ketiga adalah kontainer dengan status '*pick up*' dapat diinput oleh *driver* lain. Perbaikan yang dilakukan adalah nomor kontainer akan terkunci pada nomor lambung truk yang menjalankan kontainer bersangkutan. Jika *driver* pertama hendak menyerahkan pesanan kepada *driver* kedua, *driver* kedua harus melakukan *log in* aplikasi dengan nomor lambung truk yang sama. Evaluasi *keempat* adalah kontainer *empty* tidak bisa dikerjakan. Perbaikan yang dilakukan adalah pembuatan pesanan kontainer *empty* di S1L dapat disimpan tanpa harus mengisi lokasi depo tujuan dan pesanan kontainer *full* wajib diisi lokasi depo tujuannya. Evaluasi kelima adalah status pesanan tidak berubah menjadi '*finish*' saat konfirmasi *order* selesai. Perbaikan yang dilakukan adalah Tim IT akan memperbaiki sistem aplikasi.

Terdapat 4 evaluasi pada pengujian kedua, yaitu: terdapat tulisan 'SOPT' pada notifikasi *history*, tidak bisa mengubah nomor lambung, kontainer *empty* tidak bisa dikerjakan, dan aplikasi *error* saat konfirmasi *order* selesai. Terdapat 1 evaluasi pada pengujian ketiga, yaitu: kontainer *empty* tidak bisa dikerjakan. Setelah pengujian ketiga, semua fungsi pada aplikasi sudah dapat berjalan dengan baik, sehingga aplikasi dapat diimplementasi pada proses pengangkutan bongkar kontainer.

Analisa Produktivitas Pengangkutan

Perhitungan produktivitas ritase pengangkutan bongkar awal menggunakan rata-rata ritase Februari 2021 sampai Januari 2022. Berdasarkan data tersebut, didapatkan produktivitas ritase bongkar awal adalah 7,36 ritase per hari.

Tabel 3. Data produktivitas ritase bongkar awal

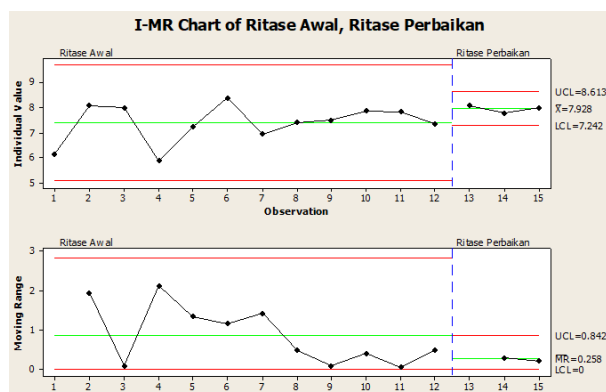
Waktu	Jumlah Ritase	Rata-rata Jumlah Truk	Ritase/Truk/Hari
Feb '21	7402	43	6,13
Mar '21	11303	45	8,06
Apr '21	10925	46	7,98
Mei '21	8030	44	5,87
Jun '21	9670	45	7,21
Jul '21	11077	43	8,34
Agst '21	9438	44	6,91
Sept '21	10098	45	7,40
Okt '21	10242	44	7,46
Nov '21	10360	44	7,85
Des '21	10449	43	7,81
Jan '21	9735	43	7,33
Rata-rata ritase			7,36

Perhitungan produktivitas ritase pengangkutan bongkar perbaikan menggunakan rata-rata ritase Maret 2022 sampai Mei 2022. Berdasarkan data tersebut, didapatkan produktivitas ritase bongkar awal adalah 7,93 ritase per hari.

Tabel 4. Data produktivitas ritase bongkar awal

Waktu	Jumlah Ritase	Rata-rata Jumlah Truk	Ritase/Truk/Hari
Mar '22	10865	44	8,06
Apr '22	10140	44	7,76
Mei '22	10954	44	7,97
Rata-rata ritase			7,93

Data ritase/truk/hari pada Tabel 3 dan Tabel 4 diuji normalitas. *P-value* data ritase awal adalah 0,256, sedangkan *p-value* ritase perbaikan adalah 0,400. *P-value* menunjukkan bahwa data berdistribusi normal, sehingga dapat digunakan untuk *two sample t-test*.

**Gambar 4.** Control chart ritase awal dan perbaikan

Control chart menunjukkan bahwa rata-rata ritase perbaikan lebih tinggi dari rata-rata ritase awal. Hasil *two sample t-test* dari ritase awal dan ritase perbaikan dengan H_0 adalah rata-rata ritase awal sama dengan rata-rata ritase

perbaikan dan H_1 adalah rata-rata ritase awal tidak sama dengan rata-rata ritase perbaikan menunjukkan bahwa *p-value* yang didapatkan adalah sebesar 0,033, dimana *p-value* lebih kecil dari tingkat signifikansi (α) sebesar 0,05 (tolak H_0). Hal tersebut menyimpulkan bahwa rata-rata ritase awal tidak sama dengan rata-rata ritase perbaikan, sehingga produktivitas ritase bongkar mengalami peningkatan secara signifikan.

Simpulan

PT. X adalah perusahaan yang bergerak di bidang logistik dan pelayaran. Salah satu alat transportasi yang digunakan dalam logistik dan pelayaran adalah truk yang digunakan untuk mengangkut kontainer dari pelabuhan ke depo untuk dibongkar. Namun, proses pengangkutan ini kurang efektif dan efisien. Tingkat produktivitas ritase pengangkutan saat ini tergolong rendah, karena proses pengangkutan yang tidak bisa dideteksi secara *real time*, media komunikasi yang tidak praktis, antrian, dan ketidakseimbangan antara jumlah pesanan dan truk yang tersedia.

Terdapat 6 proses yang tidak memberikan nilai tambah, sehingga dibuat rencana pengembangan aplikasi *trucking haulage*, baik di dalam maupun di luar *business process*. Prototipe aplikasi dibuat dan diuji. Hasil evaluasi digunakan sebagai bahan perbaikan sampai semua fungsi di dalam aplikasi berjalan dengan baik. Produktivitas ritase bongkar awal adalah 7,36 ritase per hari, sedangkan produktivitas ritase bongkar perbaikan adalah 7,93 ritase per hari. *Two sample t-test* menunjukkan bahwa *p-value* yang didapatkan sebesar 0,033, dimana *p-value* lebih kecil dari 0,05. Berdasarkan perhitungan tersebut, disimpulkan bahwa aplikasi *trucking haulage* dapat meningkatkan produktivitas bongkar kontainer di PT. X Surabaya.

Daftar Pustaka

- Nur, T., Implementasi Metode Lean Thinking untuk Mengurangi Non Value Adding Activity dan Meningkatkan Kinerja PT. XK, *Jurnal Teknologi Terapan*, 4(1), 2020, pp. 254-262.
- Syawalluddin, M. W., Pendekatan Lean Thinking dengan Menggunakan Metode Root Cause Analysis untuk Mengurangi Non Value Added Activities, *Jurnal PASTI*, 3(2), 2017, pp. 236-250.