

Autoparser Application of Automation Weighing System at PT. Flextronics Technology Indonesia

Agung Riyadi*, Ryan Binsar Sergius Pasaribu**

Multimedia and Network Engineering, Batam State Polytechnic

agung@polibatam.ac.id, rastalex11@gmail.com

Article Info

Article history:

Received Jun 12th, 2022

Revised Jun 20th, 2022

Accepted Jun 26th, 2022

Keyword:

Weighing Automation System

Auto Parser Application

Waterfall Method

Box Unit

ABSTRACT

Weighing Automation System is a system that already exists at PT. Flextronics Technology Indonesia. Weighing Automation System is an automatic scale that functions to display the weight of box unit that are complete with components. However, the results of the scales are compared manually by the operator by comparing the results of the weighing scale with the standard range weight on the PC monitor. Based on this, the author created an auto parser application with the aim of producing a desktop application as a component validation system missing in the box unit, and produce a track record of Weighing Automation System. Based on the objectives to be achieved, the problems to be discussed are how to design and build an auto parser application, how to connect the software with the Weighing Automation System and the auto parser application logic to validate the weighing results.

This research uses the waterfall method, information gathering, development design, application creation and testing, system installation and maintenance. Through this research, an auto parser application that can be connected to software and a Weighing Automation System with logic can validate the weighing results automatically. With this research, it is expected to reduce human errors during the component validation process in the box unit. The conclusion that the auto parser application can help the validation process for missing components in the box unit.

Copyright © 2022 Journal of Applied Multimedia and Networking

All rights reserved.

1. INTRODUCTION

Dalam proses produksi memerlukan tingkat ketelitian tinggi agar hasil produksi tidak cacat, salah satu kecacatan dalam suatu unit yaitu tidak adanya suatu komponen yang semestinya ada (missing component), menyebabkan unit tersebut menjadi cacat (Not Good). Sehingga apabila unit yang cacat tersebut sampai ke tangan konsumen, maka akan menimbulkan masalah, di PT. Flextronics Technology Indonesia dalam proses produksinya masih menggunakan tenaga operator untuk melakukan proses packaging atau proses memasukkan part-part ke dalam box unit yang diproduksi, yang mana setelah proses packaging tersebut, box unit akan masuk ke mesin pengeleman (taping machine) tanpa ada dilakukan proses pengecekan terlebih dahulu, sehingga ini menyebabkan potensi terjadinya missing component.

Sehingga dirancanglah suatu sistem timbangan otomatis yang mampu mendeteksi berat suatu unit, dengan menggunakan data matrix dari 100 box unit (full part) yang ditimbang, akan di dapat nilai berat minimum dan maksimum dari box unit tersebut sehingga menghasilkan nilai range, yang mana nilai range tersebut akan di jadikan patokan dalam sistem timbangan otomatis, apabila hasil penimbangan menunjukkan berat di bawah nilai berat minimum maka, sistem akan menahan box unit agar tidak masuk ke taping machine, di saat itulah dilakukan proses pengecekan terhadap box unit apakah box unit benar-benar terjadi missing component atau tidak, jika iya maka operator akan melengkapi part yang tidak ada di dalam box unit tersebut di Station Rework, jika sudah sistem akan membuka akses untuk box unit masuk ke taping machine.

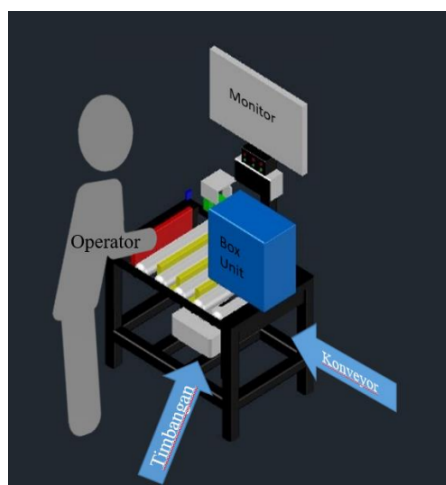
Terdapat beberapa masalah didalam penelitian ini sehingga adanya perumusan masalah sehingga didalam pembahasan penelitian ini lebih terfokus. Rumusan masalah tersebut yaitu: 1) Bagaimana merancang dan membangun aplikasi auto parser untuk Weighing Automation System. 2) Bagaimana menghubungkan perangkat lunak dengan Weighing Automation System. 3) Bagaimana aplikasi auto parser memvalidasi hasil timbang. Tujuan penelitian yaitu: 1) Menghasilkan aplikasi auto parser untuk Weighing Automation System. 2) Aplikasi auto parser bisa terhubung dengan Weighing Automation System. 3)

Aplikasi auto parser bisa memvalidasi hasil timbang. Serta batasan masalah dalam penelitian yaitu: 1) Bahasa pemrograman yang akan digunakan adalah C#. 2) Laporan penelitian ini tidak menjelaskan detail dari Weighing Automation System. 4) Weighing Automation System adalah sistem yang dibangun di PT. Flextronics Technology Indonesia. 5) Aplikasi auto parser akan digunakan oleh operator produksi yang telah di training. 6) Laporan ini menggunakan metode Waterfall. 7) Penelitian ini menggunakan Blackbox Testing untuk pengujian.

2. THEORETICAL BASIS

2.1. Weighing Automation System

Salah satu sistem yang ada di PT. Flextronics Technology Indonesia adalah *Weighing Automation System*. Sistem ini adalah sebuah mekanikal timbangan yang telah dimodifikasi dengan memasang timbangan di dalam konveyor berjalan.



Gambar 1. Sketsa *Weighing Automation System*

Pada Gambar 1 mendeskripsikan konveyor berjalan dari arah kanan ke kiri serta operator berdiri menghadap ke arah monitor. *Box unit* datang dari arah kanan dan berhenti ketika berada di atas timbangan, lalu *box unit* ditimbang dan hasil timbangan akan tampil di monitor dan divalidasi oleh operator. Jika “Pass” maka operator menggeser *box unit* kearah kiri untuk menuju proses *packaging*, jika tidak maka operator akan menimbang kembali dengan menekan tombol timbang yang berada dibawah konveyor. Jika sudah 3x “Fail” maka *box unit* akan diambil dan diarahkan ke *Rework Station*. *Weighing Automation System* berfungsi menimbang *box unit* dan menampilkan hasil berat. Operator melakukan validasi manual dengan membandingkan hasil berat unit box yang telah ditimbang dengan range berat standar box unit. Jika tidak standar, operator melakukan pengecekan *part-part* yang ada di dalam box. Setelah itu *box unit* ditimbang lagi sampai berat *box unit* berada di dalam range standar. Range standar adalah berat rata-rata dari hasil ujicoba dari 100 *box unit full component* dan diberikan jarak berat dari *part* yang paling ringan.

2.2. Aplikasi Auto Parser

Autoparser application (aplikasi *auto parser*) adalah aplikasi berbasis dekstop yang digunakan untuk melakukan *parsing* secara otomatis. *Parsing* adalah suatu cara memecah-mecah suatu rangkaian masukan atau input yang akan menghasilkan suatu pohon uraian (*parse tree*) yang akan digunakan pada tahap kompilasi berikutnya yaitu analisis semantic [1].

2.3. Bahasa Pemrograman C#

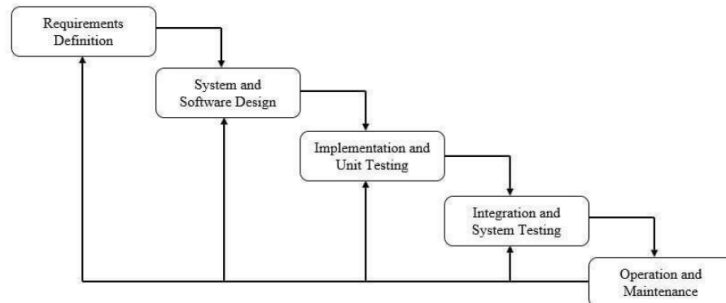
C# (*C sharp*) adalah “sebuah bahasa pemrograman berbasis objek yang didukung oleh *Microsoft .NET framework*” (Handoyo, 2019). *Microsoft .NET Framework* adalah fitur yang dimiliki *Microsoft* yang berfungsi untuk menyelaraskan bahasa pemrograman yang didukung dengan sistem operasi yang digunakan oleh komputer.

2.4. MySql (*My Structured Query Language*)

“MySQL merupakan sebuah program pembuat basis data yang bersifat *open source*. Selain itu MySQL dapat berjalan stabil pada berbagai sistem operasi seperti Windows, Linux, FireBSD, Mac OS X Server, Solaris, Amiga, dan masih banyak lagi [8]. Modul yang sudah termasuk yaitu *command-line* (perintah: *mysql* dan *mysql admin*). Dapat diunduh dari situs MySQL yaitu sebuah modul berbasis grafik (GUI): MySQL Administrator dan MySQL Query. Selain itu ada juga sebuah perangkat lunak gratis untuk administrasi basis data MySQL berbasis web yang sangat populer yaitu *phpMyAdmin*. Untuk perangkat lunak untuk administrasi basis data.

2.5. Waterfall Method

Pembangunan pada suatu sistem secara keseluruhan dilakukan dengan beberapa tahapan di dalam prosesnya. Metode pembangunan perangkat lunak dengan istilah lain *Software Development Life Cycle* (SDLC). Metode *waterfall* (air terjun) merupakan metode tertua dan pendekatan SDLC paling awal yang digunakan dalam pengembangan perangkat lunak dikarenakan sifatnya yang natural.



Gambar 2. Tahapan *Waterfall Method*

Tahap-tahap yang dilakukan dalam penggunaan *waterfall method* adalah:

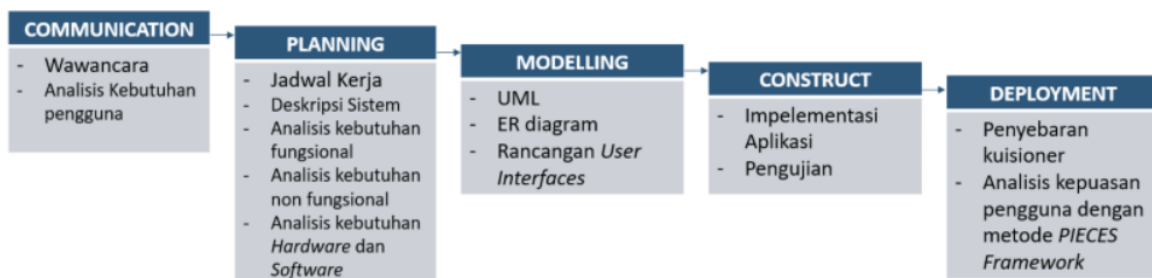
1. *Requirements Analysis and Definition*
Mengumpulkan informasi yang diperoleh dengan cara diskusi, survei dan wawancara. Kumpulan informasi yang diperoleh akan diolah dan dilakukan analisa untuk kebutuhan pengguna aplikasi *auto parser*.
2. *System and Software Design*
Ditahap ini melakukan pembangunan data berat box unit, mendesign perangkat lunak, perancangan interface, perancangan fungsi internal dan eksternal serta detail dari seluruh algoritma prosedural.
3. *Implementation and Unit Testing*
Melakukan uji fungsi aplikasi *auto parser* dan uji koneksi ke *weighing automation system*. Jika terdapat *error/bug* maka akan dilakukan perbaikan dan kembali melakukan uji fungsi.
4. *Integration and System Testing*
Seluruh unit atau modul perangkat lunak diintegrasikan ke dalam sistem secara keseluruhan dan dilakukan pemeriksaan dan pengujian sistem secara keseluruhan jika terdapat kesalahan atau kegagalan (*error*).
5. *Operational and Maintenance*
Pemeliharaan oleh pengembang adalah sebagai bentuk tanggung jawab meliputi perbaikan atas kesalahan yang belum terdeteksi sebelumnya, dan penambahan modul-modul baru untuk pengembangan aplikasi.

2.6. Black Box Testing

Black Box Testing merupakan teknik pengujian perangkat lunak yang berfokus pada spesifikasi fungsional dari perangkat lunak. *Black box testing* memungkinkan pengembang *software* untuk membuat himpunan kondisi input yang akan melatih seluruh syarat-syarat fungsional suatu program [4]. Setelah dilakukan pengujian sistem, hasil pengujian akan dievaluasi dan melakukan validasi hasil telah sesuai dengan proses yang diharapkan atau tidak.

3. RESEARCH METHOD

Metode yang digunakan untuk merancang dan membangun sistem perangkat lunak ini adalah metode *waterfall*. Digambarkan seperti pada gambar 3.



Gambar 3. Fase-fase *Waterfall Method*

3.1. Communication

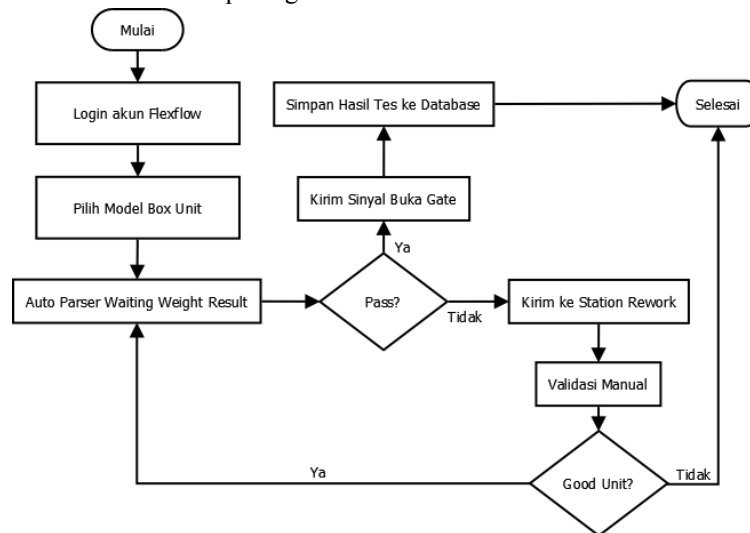
Berdasarkan hasil wawancara, dinyatakan *Weighing Automation System* belum tersambung ke *Main System* untuk menyimpan riwayat hasil tes mesin mekanikal. Selama ini yang dilakukan adalah manual inspeksi oleh operator produksi dengan membandingkan hasil timbang dengan data yang telah ada di *Work Instruction* yaitu berat standar dari satu box unit. Serta sering terjadi *missing component* di dalam box unit seperti aksesoris yang terdapat didalam box unit tidak sesuai yang menjadikan box unit (*Not Good*) ketika akan dipacking, harus dibongkar kembali dan menyebabkan banyak box unit menjadi *scrap* (dibuang). Dengan semakin sering terjadi *human error* maka diperlukan sebuah sistem yang dapat menyimpan riwayat tes dari *Weighing Automation System* serta dapat melakukan validasi hasil timbang dengan berat standar yang telah ditentukan

Dari hasil wawancara tersebut, penulis dapat memahami beberapa hal yang dibutuhkan dalam pengerjaan Tugas Akhir ini. Data yang diperlukan yaitu:

1. Dibutuhkan aplikasi perangkat lunak yang dapat melakukan validasi hasil timbang *Weighing Automation System* dengan berat standar.
2. Media akses untuk koneksi dari *Weighing Automation System* ke *desktop*.
3. *Autoparser application* dapat diakses di *desktop*.

3.2. Planning

Kebutuhan pengguna yang dianalisis meliputi: deskripsi sistem, analisis kebutuhan fungsional dan non fungsional, serta analisis kebutuhan perangkat lunak.



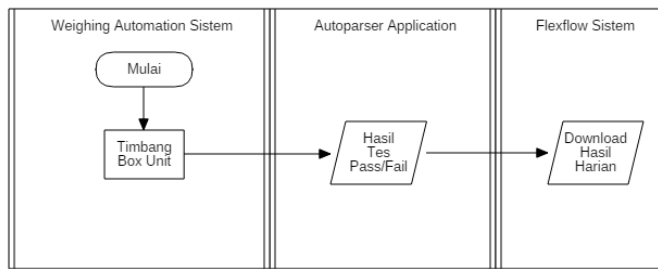
Gambar 4 Deskripsi Alur Proses Perangkat Lunak yang Akan Dibangun

Aplikasi yang dibangun merupakan aplikasi berbasis desktop yang dioperasikan oleh *user* (operator) dan berjalan otomatis yang dideteksi menggunakan sensor dan mekanikal.

Alur sistem perangkat lunak adalah sebagai berikut:

1. Operator login menggunakan akun yang telah terdaftar.
2. Operator memilih jenis model yang sedang berjalan.
3. Operator mengarahkan box unit ke timbangan.
4. Hasil timbangan divalidasi sistem dan menampilkan serial number yang dari box unit yang ditimbang.
5. Hasil uji (*Pass/Fail*) disimpan ke basis data *Main System*.

Operator yang diperbolehkan menggunakan aplikasi adalah operator yang telah di latih (*training*) oleh tim trainer dan telah dinyatakan lulus. Aplikasi membantu untuk ambil data dari *Weighing Automation System* dan menyimpan data ke sistem utama (*Flexflow System*). *Flexflow System* adalah sistem utama yang menyimpan semua rekaman hasil setiap *station* untuk ditampilkan di monitor *dashboard* sebagai persentasi produksi yang sedang berjalan. Dikarenakan kebijakan privasi dari perusahaan PT. Flextronics Technology Indonesia, *Flexflow System* tidak bisa dijelaskan lebih lanjut. Deskripsi sistem tersebut secara umum, yaitu sebagai berikut:



Gambar 5 Deskripsi Umum *Main System*

Weighing Automation System adalah mekanikal timbangan digital yang berfungsi untuk menimbang box unit untuk melakukan validasi *missing component* berdasarkan berat produk dan aksesoris yang terdapat di dalam box unit. *Auto parser* yang akan dibangun adalah aplikasi yang mengambil hasil tes dari *Weighing Automation System* berupa data terkomputerisasi dan disimpan didalam basis data *Flexflow System (Main System)*. Untuk mengunduh ataupun menampilkan seluruh hasil tes dari *Weighing Automation System* yang telah disimpan dapat diakses di *Flexflow System*. Laporan ini hanya akan menjelaskan perancangan *auto parser application* yang akan menyimpan hasil tes *Weighing Automation System* ke dalam basis data *Flexflow System*.

Kebutuhan fungsional yang dibutuhkan untuk pembangunan *autoparser application* yang akan dibangun adalah sebagai berikut.

Tabel 1 Kebutuhan Fungsional

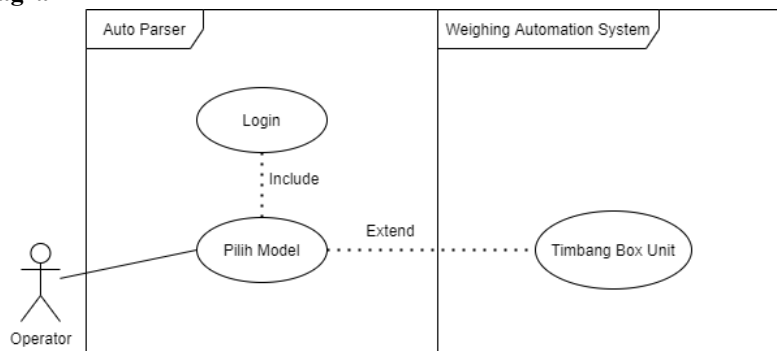
ID	Fungsi
F001	Pengguna melakukan login
F002	Pengguna pilih model <i>box unit</i>
F003	<i>Autoparser</i> menampilkan dan menyimpan hasil <i>Pass</i>
F004	<i>Autoparser</i> menampilkan dan menyimpan hasil <i>Fail</i>

Sedangkan kebutuhan non fungsional adalah sebagai berikut.

- Autoparser* dapat berjalan di komputer dekstop dengan sistem operasi Windows 10 dan versi di atasnya, serta telah terinstall *NET.Framework 3.0*.
- Autoparser* dapat terintegrasi dengan *Main System* menggunakan wifi lokal yang telah diberi akses.
- Autoparser* mempunyai fitur login pengguna yang memiliki keamanan *password*.
- Autoparser* memiliki antar muka yang mudah dipahami.
- Work Instruction* penggunaan sistem telah disediakan oleh tim *trainer*.

3.3. Modelling

A. Use Case Diagram

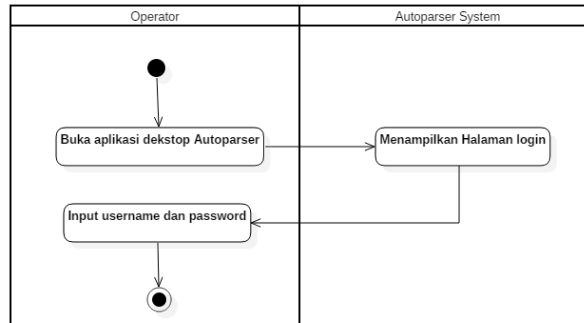


Gambar 6 Use Case Diagram

Operator melakukan login dan memilih model yang sedang berjalan atau yang akan ditimbang. Setelah itu, *Autoparser* akan melakukan validasi data secara otomatis berdasarkan hasil timbang yang dikirim dari *Weighing Automation System*. Proses tersebut dilakukan secara berulang setiap box unit akan ditimbang berdasarkan model dan *quantity production ordered*.

B. Diagram Aktivitas

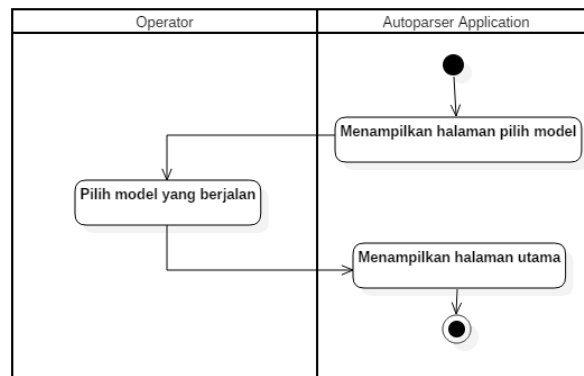
1. Diagram Aktivitas Login



Gambar 7 Diagram Aktivitas Login

Gambar 7 menggambarkan aktifitas login yang dilakukan operator produksi. Login hanya akan berhasil untuk akun *Flexflow* yang memiliki *group user* operator dan telah diberi akses oleh tim *trainer* berdasarkan hasil tes yang menyatakan operator telah lulus dan layak untuk mengoperasikan *station Weighing Automation System*.

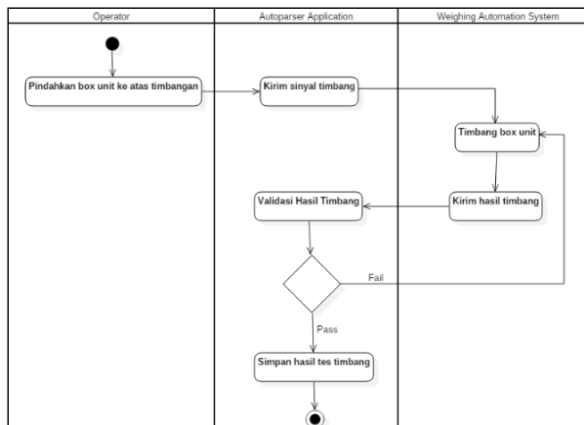
2. Diagram Aktivitas Pilih Model



Gambar 8 Diagram Aktivitas Pilih Model

Gambar 8 adalah deskripsi aktifitas pilih model yang dilakukan oleh operator produksi. Model yang dipilih berdasarkan model *box unit* yang akan ditimbang ataupun yang akan diproduksi. Setiap *Line Produksi* memiliki beberapa model dan berat dari setiap model berbeda. Oleh karena itu, Jika operator salah memilih model yang sedang berjalan *Weighing Automation System* akan memberikan sinyal *error* dan akan ditampilkan di aplikasi *auto parser* sebagai pemberitahuan kepada operator bahwa model yang dipilih tidak sesuai.

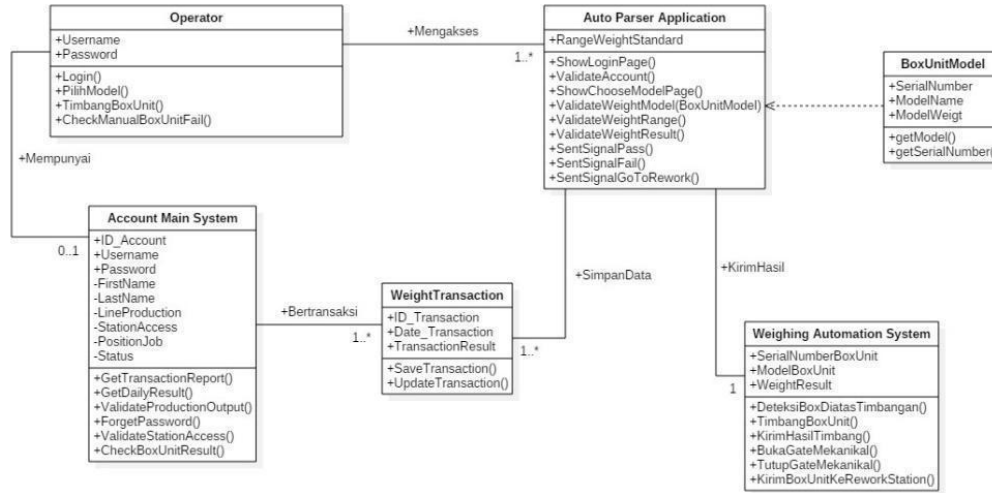
3. Diagram Aktivitas Timbang Box Unit



Gambar 9 Diagram Aktivitas Timbang *Box Unit*

Gambar 9 menggambarkan aktifitas timbang *box unit* yang dilakukan oleh *Weighing Automation System* secara otomatis setelah mekanikal mendeteksi terdapat *box unit* diatas mekanikal timbangan.

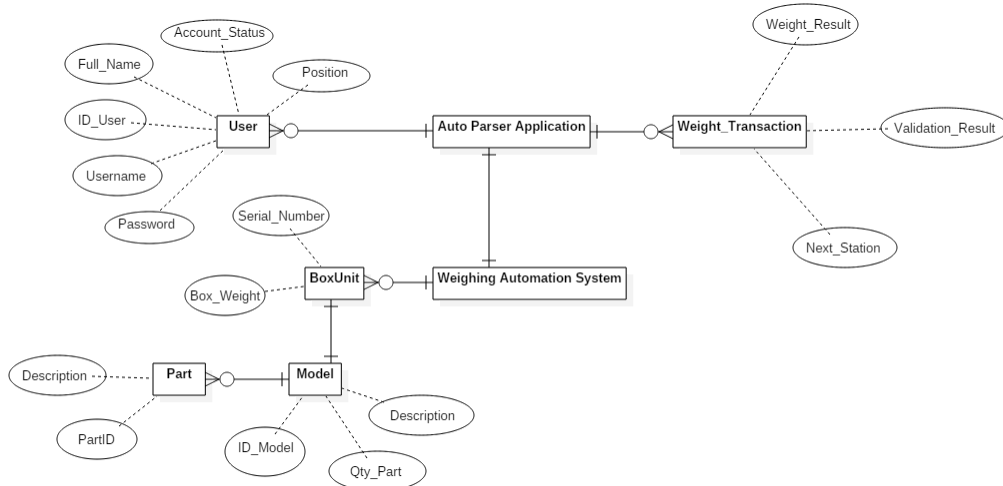
C. Class Diagram



Gambar 10 Class Diagram

Gambar 10 adalah deskripsi *class diagram* dari *auto parser* dan *weighing automation system*. Terdapat *class* operator, *auto parser application*, *box unit model*, *account main system*, *weight transaction* dan *weighing automation system*. Setiap operator mempunyai satu akun *main system*, operator bisa mengakses *auto parser application* dengan melakukan *login*. Sedangkan *class auto parser application* dimiliki oleh satu mekanikal *weighing automation system*. *Box unit* memiliki *attribute* serial number dan berat per model yang akan digunakan oleh *class auto parser application* sebagai validasi berat per model. Dan semua transaksi akan tersimpan di *class weight transaction*.

D. EERD (Enhanced Entity Relational Diagram)



Gambar 11 Enhanced Entity Relational Diagram

Pada gambar 11 menggambarkan perancangan hubungan antar entitas sistem yang akan dibangun. *Entitas part* memiliki relasi banyak ke satu *entitas model*. *Entitas model* memiliki relasi satu ke satu dengan *entitas box unit* karena dalam satu *box unit* hanya ada 1 model. *Entitas user* memiliki relasi ke *entitas auto parser* yang merupakan sistem yang akan dibangun dan *entitas weight transaction* adalah hasil tes dari setiap transaksi timbang.

4. RESULTS AND ANALYSIS

4.1. Lingkungan Impementasi

Implementasi *auto parser application* yang dibangun berada di PT. Flextronics Technology Indonesia. PT. Flextronics Technology Indonesia yang beralamat Gg. Mangga No. Lot 515, Muka Kuning, Kec. Sei Beduk, Kota Batam, Kepulauan Riau, Indonesia.

4.2. Antarmuka

a). Antarmuka Menu Login



Gambar 12 Antarmuka Menu Login

Pada gambar 12 menggambarkan antarmuka yang telah dibangun. Terdapat *icon human* dan *key* sebagai petunjuk untuk login. Operator input *username (Badge No)* dan *password* dan klik tombol *ok* atau tombol *cancel* untuk keluar.

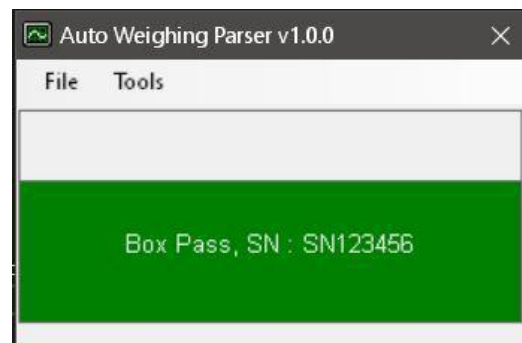
b). Antar Muka *Ready to Scale*



Gambar 13 Antar Muka *Ready To Scale*

Gambar 13 adalah antarmuka halaman awal aplikasi *auto parser*. Terdapat *alert "Waiting Scale Result"* yang berarti aplikasi sudah *stand by* dan menunggu hasil timbang dikirim dari *weighing automation system*.

c). Antarmuka Hasil *Pass*



Gambar 14 Antarmuka Hasil *Pass*

Gambar 14 adalah antarmuka hasil timbang *pass* yang menampilkan hasil validasi *box unit pass* dan serial number *box units* dengan *background* hijau.

d). Antarmuka Hasil *Fail*



Gambar 15 Antarmuka Hasil *Fail*

Gambar 15 menampilkan antarmuka hasil timbang *fail*, dengan informasi *box unit* yang ditimbang *fail* dan operator diarahkan untuk cek komponen secara manual. Serta diinformasikan juga berat standar dari *box unit* tersebut.

e). Antarmuka Pilih Model



Gambar 16 Antarmuka Pilih Model

Gambar 16 merupakan antarmuka halaman pilih model dan akan tampil daftar model yang bisa ditimbang dengan *klik dropdown* pilih model. Setelah dipilih operator (*user*) akan *klik* tombol ok.

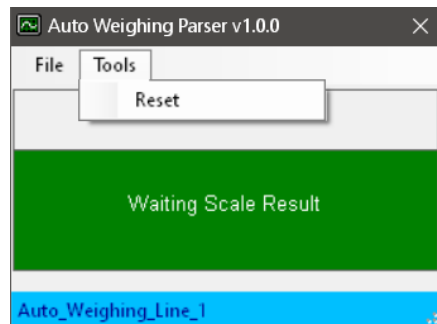
f). Antarmuka Menu File



Gambar 17 Antarmuka Menu File

Gambar 17 merupakan menu file yang dirancang untuk situasi *line production* akan mengganti model yang sedang berjalan. Sub menu *change model* telah disediakan untuk keluar dari *main menu* dan kembali ke pilih model, sedangkan submenu *logout* berfungsi sebagai fitur keluar dari aplikasi dan kembali ke halaman login.

g). Antarmuka Menu *Tools*



Gambar 18 Antarmuka Menu *Tools*

Gambar 18 merupakan antarmuka menu *tools* yang disediakan untuk *reset* proses jika terjadi malfungsi dari sistem ataupun mekanikal *weighing automation system*.

4.3. Analisis Hasil Pengujian

Tabel 4 Modul Pengujian *User (Operator)*

No	Modul Pengujian	<i>Defect</i> yang ditemukan	Tipe Kesalahan	Jenis <i>Defect</i>	Keterangan
1	Modul Halaman Login	Operator salah input <i>username</i> atau <i>password</i>	<i>Failure</i>	<i>Human Error</i>	Jika 3 kali melakukan salah <i>password</i> , akun otomatis terkunci dan perlu <i>approval supervisor</i> jika ingin <i>unlock account</i>
2	Modul Halaman Login	-	-	-	Sukses
3	Modul Halaman Pilih Model	Operator salah pilih model	<i>Failure</i>	<i>Human Error</i>	<i>Auto parser application</i> akan menampilkan <i>alert</i> model yang dipilih tidak sesuai dengan unit yang ditimbang
4	Modul Halaman Pilih Model	-	-	-	Sukses
5	Modul Halaman Utama	-	-	-	Sukses
6	Modul <i>Pass Result</i>	-	-	-	Sukses menampilkan " <i>Alert</i> " <i>pass</i>
7	Modul <i>Fail Result</i>	<i>Weighing Automation System</i> mengirim hasil timbang yang berbeda	<i>Failure</i>	<i>Mechanical Error</i>	Timbangan perlu di kalibrasi setiap pengantian <i>shift</i> kerja
8	Modul <i>Fail Result</i>	-	-	-	Sukses menampilkan " <i>Alert</i> " <i>fail</i>
9	Modul logout	-	-	-	Sukses menampilkan halaman login setelah sub menu logout di klik
10	Modul <i>Change Model</i>	Model yang berjalan tidak tersedia didalam list pilihan	<i>Failure</i>	<i>Application Error</i>	Aplikasi tidak dapat menampilkan model yang belum di <i>approve Quality Engineer</i> .
11	Modul <i>Tools Menu Sub menu reset</i>	-	-	-	Sukses menampilkan halaman awal setelah sub menu di klik

5. CONCLUSION

Berdasarkan hasil penelitian yang telah penulis laksanakan di PT. Flextronics Technology Indonesia, maka dapat disimpulkan bahwa mekanikal Weighing Automation System dapat tersambung ke desktop PC dengan koneksi kabel serial port RS232 to USB. Auto parser application dibangun dengan bahasa pemrograman C# dan dapat melakukan validasi hasil timbang dengan membandingkan hasil timbang dengan range standar setiap model secara otomatis serta menyimpan hasil tes ke dalam basis data Main System. Trigger yang diberikan oleh aplikasi berbentuk sinyal voltase sebesar 5 volt melalui kabel serial port RS232 to USB.

Dengan menggunakan timbangan, komponen yang memiliki berat dibawah 1 gram tidak akan terdeteksi, penulis menyarankan perlu ditambahkan kamera vision diatas box unit sebagai tambahan module Weighing Automation System sebagai alat validasi komponen yang ringan. Selain itu di PT. Flextronics bersedia memberikan kesempatan bagi karyawan yang ingin mengembangkan dan membangun aplikasi baru di setiap station yang dibutuhkan.

REFERENCES

Autoparser Application of Automation Weighing System (Ryan Binsar Sergius Pasaribu)

- [1] Yulianto. (2015). Implementasi Penggunaan Sistem Aplikasi Web PDF Parser untuk Menampilkan Informasi Isi Dokumen. Tangerang.
- [2] Jaya, T. S. (2018). Pengujian Aplikasi dengan Metode Blackbox Testing Boundary Value Analysis. Jurnal Pengembangan IT (JPIT).
- [3] Agung Satria Alam, Hendi Matalata. Perancangan Alat Pengolahan Air Minum Otomatis Pada Proses Netralisasi Ph Dan Aerasi. Journal of Electrical Power Control and Automation (JEPCA). 2018.
- [4] Munawar. (2005). Pemodelan Visual Dengan UML. Yogyakarta: Graha Ilmu.
- [5] Kristanto, A. (2010). Kupas Tuntas PHP & MySQL, Cable Book. Klaten.
- [6] Furqan, A. A. (2016). Rancang Bangun Timbangan Beras Digital Dengan Keluaran Berat Dan Harga Berbasis Mikrokontroler. Makassar: Tugas Akhir, Universitas Alaudin.
- [7] Meifia, S. (2017). Perancangan Timbangan Digital Buah dengan Keluaran Massa dan Harga Menggunakan Sensor Load Cell Berbasis Atmega 328. Medan: Skripsi, Universitas Sumatera Utara.
- [8] Sasongko, B.H. (2012). Pemrograman Mikrokontroler dengan Bahasa C. Yogyakarta: Andi.