

USULAN ALOKASI PENYIMPANAN PRODUK BERDASARKAN KEBIJAKAN CLASS BASED STORAGE DAN PENENTUAN METODE ROUTING MENGGUNAKAN ALGORITMA DYNAMIC PROGRAMMING UNTUK MEMINIMASI DELAY TIME PADA PROSES ORDER PICKING DI GUDANG SEMI FINISHED GOOD PT. XYZ

DESIGN OF STORAGE ALLOCATION BASED ON CLASS BASED STORAGE POLICIES AND DETERMINING ROUTING METHOD USING DYNAMIC PROGRAMMING ALGORITHM TO MINIMIZE DELAY TIME OF PICKING PROCESS IN SEMI FINISHED GOOD WAREHOUSE (STUDY CASE : PT.XYZ)

Dicky Abrian Setiyanto¹, Agus Kusnayat², Widia Juliani³

^{1, 2, 3} Program Studi Teknik Industri, Fakultas Rekayasa Industri, Telkom University

dickyabrians@student.telkomuniversity.ac.id, aguskusnayat17@gmail.com, widiajuliani@yahoo.com

Abstrak

PT. XYZ merupakan sebuah perusahaan tekstil yang memproduksi kain serta menjadi penyedia global *fashion* dan *corporate apparel*. PT. XYZ memiliki gudang *semi finished good* atau gudang *greige*. Pada gudang *semi finished good*, produk yang disimpan berupa kain polos yang telah melalui proses *weaving*. Lokasi PT. XYZ berada di Kota Cimahi. Sistem penyimpanan produk pada gudang *semi finished good* PT. XYZ masih dilakukan secara acak dengan cara mencari tempat penyimpanan yang kosong tanpa memperhatikan karakteristik dari masing-masing produk kain yang akan disimpan. Hal ini berdampak pada tingginya waktu siklus dan tidak tercapainya waktu target yang telah ditetapkan oleh perusahaan itu sendiri terutama pada proses *storing* dan *picking*. Delay itu disebabkan oleh tingginya waktu pencarian lokasi kain.

Untuk menangani masalah *delay* yang terjadi pada gudang *semi finished good* PT. XYZ, diusulkan dengan merancang usulan alokasi penyimpanan produk berdasarkan kebijakan *class based storage* serta penentuan *slotting* dan zonafikasi. Langkah awal dari penelitian ini adalah dengan mengidentifikasi penyebab *delay* menggunakan diagram *fishbone*. Setelah itu mengklasifikasikan produk menggunakan *FSN analysis*. Tahap akhir dari penelitian ini adalah melakukan simulasi *picking list* serta menentukan rute terpendek menggunakan algoritma *dynamic programming* untuk membandingkan kondisi eksisting dengan kondisi usulan setelah penelitian ini selesai.

Berdasarkan hasil perbandingan antara kondisi eksisting dan kondisi usulan, didapatkan penurunan waktu *delay* sebesar 351,33 detik lebih rendah dari kondisi eksisting.

Kata Kunci : *FSN Analysis, Warehouse Slotting, Dynamic Programming*.

Abstract

PT. XYZ is a company that produces fabrics and being a global supplier of fashion and corporate apparel. PT. XYZ has a semi finished good warehouse or greige warehouse as they said. In semi finished good warehouse, the product that stored in the warehouse is a plain fabric that has been through the weaving process. PT. XYZ has located in Cimahi. Product storage system in semi finished good warehouse is still done randomly by searching for empty storage without considering the characteristic of the product that will be stored. And the impact is the cycle time has being high as well and the company cannot reach the target of the company, especially in the storing and picking process. Delay was caused by the high search of product location time.

To solve the problem of delay that occurs in PT. XYZ's semi finished good warehouse, is proposed by designing proposed product storage allocations based on class based storage policies as well as determination of slotting and zonafication. The first step of this research is identifying the cause of delay using value stream mapping. After that, classify the product using FSN analysis. The final step is to simulate the picking list and determine the shortest route using dynamic programming algorithm to compare actual condition with the proposed condition after this research is done.

Based on comparasion of actual condition and proposed condition, the delay time has decreased by 351,33 seconds lower than the actual condition.

Keywords : *FSN Analysis, Warehouse Slotting, Dynamic Programming*

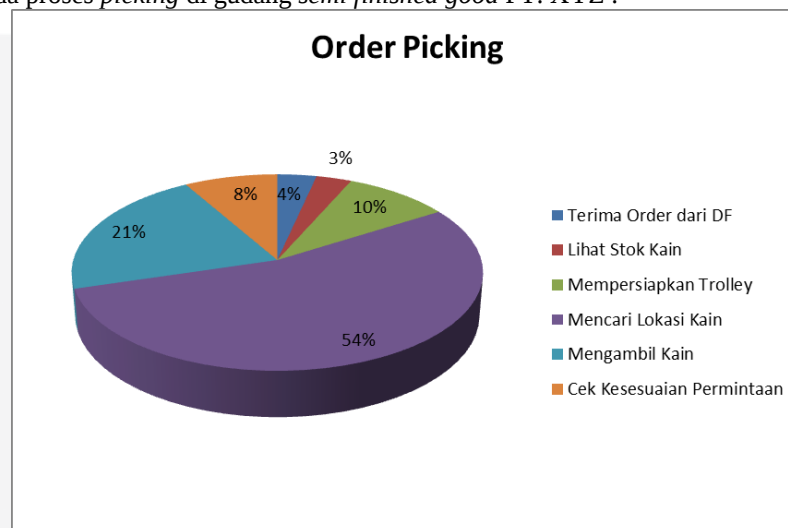
1. Pendahuluan

PT. XYZ merupakan sebuah perusahaan tekstil yang memproduksi kain serta menjadi penyedia global *fashion* dan *corporate apparel*. Penyimpanan secara *random* atau produk disimpan berdasarkan *slot* yang kosong menyebabkan waktu siklus setiap proses yang terjadi di gudang menjadi rendah jika dibandingkan dengan waktu standar yang telah ditentukan oleh perusahaan. Hal tersebut dapat dibuktikan dengan data waktu proses penyimpanan dan pengambilan barang pada gudang *semi finished good* di bawah ini :

Proses	<i>Receiving</i>	<i>Storing</i>	<i>Picking</i>	<i>Shipping</i>
Waktu Siklus (detik)	302,00	413,67	773,50	210,87
Target Waktu (detik)	240	360	600	200

Gambar 1. Perbandingan Waktu Siklus dan Waktu Standar

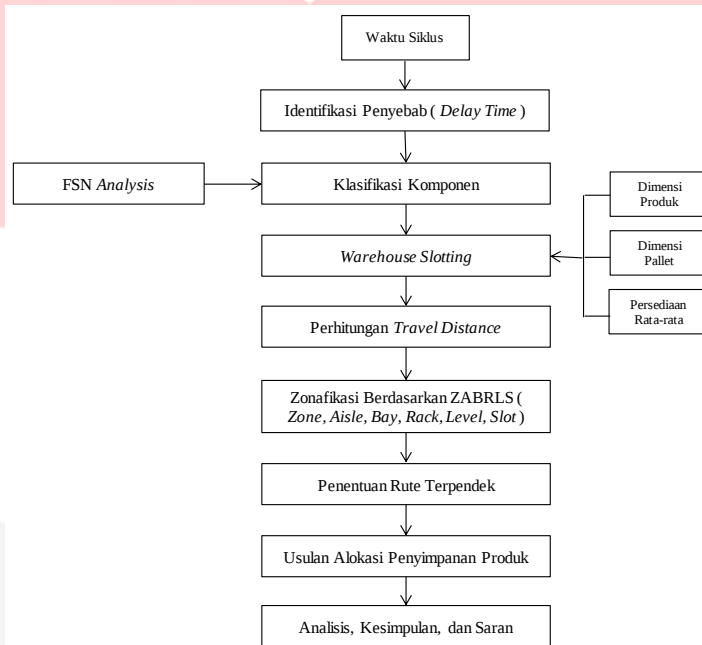
Dari data diatas juga dapat disimpulkan bahwa ada masalah pada proses *storing* dan *picking* karena memiliki waktu siklus yang lebih tinggi dari waktu standar yang telah ditentukan perusahaan dan proses *picking* adalah proses dengan gap waktu tertinggi antara waktu siklus dan waktu standarnya yaitu sebesar 226,04 detik. Untuk mengetahui setiap rincian aktivitas yang memberikan kontribusi terbesar pada aktivitas *picking*, berikut dijabarkan secara rinci beberapa aktivitas yang ada pada proses *picking* di gudang *semi finished good* PT. XYZ :



Gambar 2. Data Kontribusi Pada Proses Order Picking

Pada Gambar diatas dapat kita lihat bahwa proses pencarian lokasi produk memiliki presentase kontribusi tertinggi yaitu sebesar 58% dari keseluruhan proses pengambilan. Aktivitas pencarian lokasi produk ini disebabkan oleh kebijakan perusahaan dalam penyimpanan barang yang dilakukan secara *random* dan dapat mempengaruhi operator dalam proses pencarian lokasi produk.

2. Metodologi Penelitian



Gambar 3. Metode Konseptual

Waktu baku dan proses bisnis *existing* akan digunakan untuk mencari akar masalah yang ada di gudang *Semi Finished Good* PT. XYZ. Untuk mencari akar masalah digunakan *Value-Stream Mapping (Current State)* pada gudang *Semi Finished Good* PT. XYZ.

Setelah akar masalah ditemukan, maka langkah selanjutnya adalah mengklasifikasikan setiap produk berdasarkan *FSN Analysis*. Kelas F untuk produk *fast moving*, kelas S untuk produk *slow moving*, dan kelas N untuk produk *non moving*. Untuk melakukan *FSN Analysis* dibutuhkan data *average stay* dan *consumption rate* yang didapatkan dari data *delivery order* dan data stok produk.

Penataan dan pengalokasian produk dilakukan setelah produk diklasifikasikan dalam kelasnya masing-masing. Selanjutnya adalah penentuan *slot* untuk setiap kelasnya untuk menentukan kapasitas produk dalam satu *slot* dan menghitung jumlah *slot* yang dibutuhkan untuk menyimpan produk.

Setelah menentukan *slot* untuk tiap kelasnya, langkah selanjutnya yaitu menentukan *travel distance* pada setiap lokasi penyimpanan yang fungsinya untuk menentukan *slot* yang memiliki waktu pengambilan paling rendah sehingga prioritas tertinggi akan diletakkan pada *slot* dengan waktu pengambilan paling rendah.

Selanjutnya adalah pengaturan zonafikasi produk yang mengatur dimana kelas tersebut akan disimpan. Pengaturan zonafikasi ini membutuhkan data dimensi rak, jumlah rak, layout rak, dimensi *slot*, dan jumlah *slot*.

Setelah adanya usulan perbaikan, peneliti melakukan perbandingan antara kondisi *existing* dengan kondisi usulan dengan cara menghitung waktu *delay* antara kondisi *existing* dengan kondisi usulan. Pembuatan rancangan *Value Stream Mapping (Future State)* juga dibuat untuk mengetahui apakah masalah pada kondisi *existing* sudah diselesaikan. Setelah itu, untuk menentukan alokasi yang optimal dilakukan dengan algoritma *dynamic programming* sehingga rata-rata *travel time* menjadi minimum.

Terakhir adalah penarikan kesimpulan dan saran dari hasil penelitian yang telah peneliti lakukan.

2.1 Analisis FSN

FSN analysis bertujuan untuk mengelompokkan item berdasarkan gerakan dari persediaan. Item diklasifikasikan menjadi *fast moving* (F), *slow moving* (S), dan *non-moving* (N) dengan mempertimbangkan *average stay* dan *consumption rate*. Item dapat diklasifikasikan ke dalam *fast moving* jika memiliki *average stay* yang rendah dan *consumption rate* yang tinggi.

Langkah-langkah FSN :

- Menghitung *Average Stay*

$$\text{Average Stay} = \frac{\text{Inventory Holding Balance}}{(\text{Opening Balance} + \text{Total Reciept})}$$

- Menghitung *Consumption Rate*

$$\text{Consumption Rate} = \frac{\text{Total Issue Quantity}}{(\text{Total Period Duration})}$$

- Bandingkan hasil *Average Stay* dengan *Consumption Rate*

Tahap terakhir dalam mengklasifikasikan produk berdasarkan FSN *analysis* yaitu, menggabungkan klasifikasi FSN berdasarkan *average stay* dengan klasifikasi FSN berdasarkan *consumption rate* berdasarkan matriks FSN yang dapat dilihat pada Tabel dibawah.

AVERAGE STAY	CONSUMPTION RATE	FINAL FSN
F	F	F
F	S	F
F	N	S
S	F	S
S	S	S
S	N	N
N	F	S
N	S	N
N	N	N

- *FSN Threatment*

Untuk masing-masing kelas dalam FSN, akan dilakukan threatment berbeda berdasarkan tingkat consumption rate. Misal, meskipun terdapat dua SKU yang memiliki kelas yang sama, kedua SKU tersebut akan mendapatkan perlakuan yang berbeda-beda dengan melihat consumption rate. Contoh perhitungan FSN threatment dapat dilihat pada tabel di bawah ini:

SKU	AVERAGE STAY	CONSUMPTION RATE	FSN CLASIFICATION
70608-TR	F	N	S
85255 modif	N	N	N
HP-03122	S	S	S
HP-03128	S	S	S
HP-03129	F	N	S
HP-03131	F	F	F
HP-03144	F	F	F
HP-03148	F	F	F
HP-03160	F	F	F
HP-03197	F	S	F
.....			
LP-93652	N	N	N
T-70427-TR	F	N	S
T-70435-TR	S	N	N
T-70436-TR	F	N	S
T-70604-TR	N	N	N
T-70608-TR	N	N	N
T-70610-TR	S	N	N
T-70613-TR	S	N	N
T-70614-TR	N	N	N
T-70615-TR	N	N	N

2.2 Warehouse Slotting

Pada usulan perancangan perbaikan, digunakan klasifikasi produk berdasarkan FSN. Produk-produk yang telah diklasifikasikan kemudian dilakukan slotting berdasarkan kelasnya. Setiap slot jenis produk yang berbeda-beda. Berikut merupakan contoh perhitungan kapasitas untuk produk kategori food untuk SKU 70608-TR yaitu:

- Menghitung jumlah unit yang disimpan tiap periode

$$\text{Total Kain Disimpan Per Periode} = \frac{\text{Total Inventory Closing Balance}}{\text{Total Periode}}$$

$$= \frac{8650}{12}$$

$$= 720,83 \approx 721 \text{ meter/periode}$$

- Menghitung jumlah kain yang disimpan pada setiap slot

$$\text{Qty A} = \frac{\text{Luas Pallet}}{\text{Luas Produk}}$$

$$\text{Qty A} = \frac{(180\text{cm} \times 180\text{cm})}{(180\text{cm} \times 10)}$$

$$= 18 \text{ roll kain}$$

$$\text{Qty B Level 1} = \frac{\text{Tinggi Maksimum Tumpukan Pallet}}{\text{Tinggi Produk}}$$

$$\text{Qty B} = \frac{168\text{cm}}{14\text{cm}}$$

$$= 12 \text{ roll kain}$$

$$\text{Qty B Level 2} = \frac{\text{Tinggi Maksimum Tumpukan Pallet}}{\text{Tinggi Produk}}$$

$$\text{Qty B} = \frac{154\text{cm}}{14\text{cm}}$$

$$= 11 \text{ roll kain}$$

$$\text{Qty B Level 3} = \frac{\text{Tinggi Maksimum Tumpukan Pallet}}{\text{Tinggi Produk}}$$

$$\text{Qty B} = \frac{140\text{cm}}{14\text{cm}}$$

$$= 10 \text{ roll kain}$$

$$\text{Qty B Level 4} = \frac{\text{Tinggi Maksimum Tumpukan Pallet}}{\text{Tinggi Produk}}$$

$$\text{Qty B} = \frac{126\text{cm}}{14\text{cm}}$$

$$= 9 \text{ roll kain}$$

$$\text{Total Qty Level 1} = \text{Qty A} \times \text{Qty B}$$

$$= 13 \text{ roll} \times 12 \text{ roll}$$

$$= 156 \text{ roll kain}$$

$$\text{Total Qty Level 2} = \text{Qty A} \times \text{Qty B}$$

$$= 13 \text{ roll} \times 11 \text{ roll}$$

$$= 143 \text{ roll kain}$$

$$\text{Total Qty Level 3} = \text{Qty A} \times \text{Qty B}$$

$$= 13 \text{ roll} \times 10 \text{ roll}$$

$$= 130 \text{ roll kain}$$

$$\text{Total Qty Level 4} = \text{Qty A} \times \text{Qty B}$$

$$= 13 \text{ roll} \times 9 \text{ roll}$$

= 117 roll kain

2.3 Zonafikasi Rak Penyimpanan

Proses ini merupakan proses perbaikan yang bertujuan untuk memudahkan operator dalam melaksanakan proses operasional pada gudang PT XYZ. Pada proses ini dilakukan penentuan area penyimpanan pada rak, dalam menentukan area tersebut diperlukan zonafikasi yang bertujuan untuk menyimpan produk berdasarkan pendekatan analisis yang telah dilakukan yaitu analisis FSN. Pemberian label atau kodefikasi dilakukan berdasarkan ZABRLS yaitu singkatan dari Zone, Aisle, Bay, Rack, Level, Slot. Berikut merupakan proses zonafikasi pada rak untuk penyimpanan SKU 70608-TR

1 2 3 4 5 6
GR-3-4-6-3-04

Gambar 4. Kode Penyimpanan Produk

2.4 Penentuan Rute Terpendek Menggunakan Algoritma *Dynamic Programming*

Algoritma *Dynamic Programming* adalah teknik yang sangat ampuh untuk menyelesaikan atau memecahkan masalah tertentu seperti *travelling salesman problem* pada proses *travelling* karena *dynamic programming* memiliki rumusan pendekatan yang elegan dan pemikiran yang sederhana sehingga metode ini sangat mudah untuk digunakan¹. *Dynamic programming* merupakan sebuah model matematis yang dirancang secara prosedural dengan tujuan dapat melakukan membagi-bagi masalah hingga dalam bentuk yang lebih kecil. Inti dari pemikiran ini adalah jika anda telah memecahkan masalah dengan masukan yang diberikan, maka simpan hasilnya untuk referensi di kemudian hari agar anda tidak menyelesaikan masalah yang sama lagi. Jika masalah yang diberikan dipecahkan dalam sub-masalah yang lebih kecil dan sub-masalah yang lebih kecil ini akan tetap menjadi sub-masalah yang lebih kecil lagi. Dalam proses ini, jika anda mengamati beberapa sub-masalah yang banyak, maka itu akan menjadi petunjuk besar untuk menggunakan *dynamic programming* ini. Selain itu, solusi optimal untuk sub-masalah akan berkontribusi pada solusi optimal dari keseluruhan masalah yang diberikan.

Untuk menghasilkan waktu pencarian lokasi kain usulan yang diperbaiki menggunakan algoritma *dynamic programming* adalah dengan melakukan simulasi order picking menggunakan LINGO dari 3 buah picking dengan masing-masing titik picking yang telah ditentukan dengan simulasi Monte Carlo. Berikut merupakan hasil dari simulasi menggunakan LINGO.

```
Global optimal solution found.
Objective value:                186.9000
Objective bound:                186.9000
Infeasibilities:                0.000000
Extended solver steps:          0
Total solver iterations:        10
```

Variable	Value	Reduced Cost
N	4.000000	0.000000
U(1)	0.000000	0.000000
U(2)	3.000000	0.000000
U(3)	2.000000	0.000000
U(4)	1.000000	0.000000

Gambar 5. Hasil Simulasi Menggunakan LINGO

3. Pembahasan

3.1 Klasifikasi FSN

FSN *analysis* bertujuan untuk mengklasifikasikan produk berdasarkan kelas dari masing-masing produk. Metode ini mengklasifikasikan produk menjadi 3 kelas yaitu F (*fast moving*), S (*slow moving*), N (*non moving*). Berikut merupakan hasil dari FSN *analysis*.

Kelas	Jumlah SKU	Keterangan
F	24	Kelas F atau <i>fast moving</i> merupakan kelas yang diisi oleh SKU dengan <i>consumption rate</i> terbesar dan <i>average stay</i> terkecil. SKU yang masuk kedalam kelas F akan ditempatkan pada lokasi yang mudah dijangkau oleh operator dalam waktu yang singkat
S	54	Kelas S atau <i>slow moving</i> akan ditempatkan pada lokasi yang lumayan mudah dijangkau oleh operator dalam waktu yang lumayan singkat
N	111	Kelas N atau <i>non moving</i> merupakan kelas yang diisi oleh SKU dengan <i>consumption rate</i> terkecil dan <i>average stay</i> terbesar sehingga SKU ini akan ditempatkan pada lokasi yang susah dijangkau oleh operator dan dapat ditempuh dalam waktu yang paling lama

3.2 Analisis Kondisi Aktual Dengan Kondisi Usulan

Dengan melakukan rancangan usulan perbaikan, waktu *delay* dapat dikurangi karena disebabkan oleh pengurangan waktu pencarian lokasi pada aktivitas *storing* dan *order picking*. Berikut merupakan perbandingan waktu proses di gudang *semi finished good* PT. XYZ.

	<i>Receiving</i> (detik)	<i>Storing</i> (detik)	<i>Picking</i> (detik)	<i>Shipping</i> (detik)
Kondisi Eksisting	302,00	413,67	773,50	210,87
Kondisi Usulan	153,42	326,14	509,7	122,2
Gap	-	87,5267	263,803	-

Pada data diatas menjelaskan bahwa adanya perbedaan antara kondisi eksisting dan kondisi usulan. Terlihat bahwa terjadi perubahan pada proses *storing* dan *picking* yang disebabkan oleh adanya usulan perbaikan dimana pada proses *storing* memiliki penurunan sebesar 87,53 detik dan pada proses *picking* memiliki penurunan sebesar 263,8 detik. Hal ini terjadi karena waktu *delay* yang terjadi sebelumnya telah mengalami penurunan. Berikut merupakan perbandingan target waktu dan waktu siklus setelah usulan.

	<i>Receiving</i>	<i>Storing</i>	<i>Picking</i>	<i>Shipping</i>
Target Waktu (detik)	175	350	600	125
Waktu Siklus (detik)	153,42	326,14	509,7	122,2

Dari data diatas dapat dilihat bahwa adanya penurunan waktu *delay* sebesar 351,33 detik dan dengan adanya perubahan ini, dapat disimpulkan bahwa kinerja pada gudang *semi finished good* PT. XYZ dapat ditingkatkan.

4. Kesimpulan

Kesimpulan dari penelitian ini adalah penurunan *delay time* sebesar 351,33 detik dan kinerja pada gudang *semi finished good* PT. XYZ menjadi lebih optimal.

Referensi :

- Adhitya, A. V. (2014). *Troli Pengikut Otomatis Berbasis Mikrokontroler AVR*. Widya Teknik, 1.
- Fajriyanti, G. O., Ridwan, A. Y., Santosa, B. (2016). *Design of Storage Allocation at Machine Spare Part Warehouse Using Class Based Storage Policy to Reduce Delay Time at PT. EFG*. Bandung. Telkom University.
- Francis, R. L. (1992). *Facility Layout and Location, An Analytical Approach*. New Jersey: Prentice Hall.
- Frazelle, E. H. (2002). *World Class Warehousing and Material Handling*. New York: Mc-Graw-Hill.
- Hines, P. H., & Rich, N. (2004). *Learning to Evolve: A Review of Contemporary Lean Thinking*. International Journal of Operations & Production Management.
- Icun, H. Y. (2006). *Business Concept Implementation : In Sales & Distribution*. Jakarta: Elex Media.
- IOMA. (2002). *The IOMA Handbook of Logistics and Inventory Management*. New York: John Wiley & Sons, Inc.
- Marchelina, E., Ridwan, A. Y., Santosa, B. *Design of Storage Allocation to Reduce Delay Time For PUMA Program by Using Dedicated Storage at Tools Crib 500T PT. EFG*. Bandung. Telkom University
- Mulcahy, D E. (1994). *Warehouse Distribution and Operation Handbook*. McGraw-Hill Education.
- Nash, M. A., & Poling, S. R. (2008). *Mapping The Total Value Stream*. New York: Taylor & Francis Group.
- Richards, G. (2014). *Warehouse Management : A Complete Guide To Improving Efficiency And Minimizing Costs In The Modern Warehouse*. United States.
- Singhal, Dr. Abha (2016). *Travelling Salesman Problem By Dynamic Programming Algorithm*. International Journal of Scientific Engineering and Applied Science (IJSEAS)
- Sutalaksana, I. Z. (1997). *Perancangan Sistem Kerja*. Bandung: Departemen Teknik Industri ITB.
- Taha, H. A. (1996). *Operations Research*. New Jersey: Prentice Hall.
- Timilsina, P. dan Khanal, R. (2016). *Spare Parts Management in Heavy Equipment Division (HED), Department of Roads. European Journal of Advances in Engineering and Technology*, 119-124.