OPTIMASI PENGATURAN STEAM UAP MENGGUNAKAN ALGORITMA GREEDY UNTUK MENDUKUNG PROSES PEREBUSAN KELAPA SAWIT

Sunanto[1)], Faural Abadi [2)]

[1]Fakultas Ilmu Komputer, Universitas Muhammadiyah Riau
email: sunanto@umri.ac.id
[2]Fakultas Ilmu Komputer, Universitas Muhammadiyah Riau
email: faural.abadi09@gmail.com

Abstract

The sterilizer station is the core station in the processing of palm fruit. Where in the process of boiling oil palm fresh fruit bunches must be in the applicable boiling standard. In order to obtain high yields. To optimize the work of the boiling station, especially in horizontal boiling vessels to avoid trouble or problems that arise during boiling, an automation system is made that can regulate the amount of steam pressure that enters the boiling vessel with a predetermined time so that it can adjust the inlet valve, condensate valve, exhaust valve. Where this automation system utilizes the greedy algorithm method. And in this research, the implementation, testing and results of the automation system will be carried out, on a prototype that utilizes the NodeMCU microcontroller (esp32), the Web Thinger IO, and the pressure sensor.

Keywords: *sterilizer, greedy algorithm, NodeMCU ,esp32*

Abstrak

Stasiun sterilizer (perebusan) merupakan stasiun inti dalam pengolahan buah sawit. Dimana dalam proses perebusan tandan buah segar kelapa sawit harus dalam standar perebusan yang berlaku. Agar didapat hasil rendemen yang tinggi. Untuk mengoptimalkan kerja stasiun perebusan tersebut khususnya pada bejana rebusan horizontal agar terhindar dari trouble atau masalah-masalah yang muncul saat perebusan dibuatlah sebuah sistem automatisasi yang dapat mengatur besaran tekanan uap yang masuk kedalam bejana rebusan dengan waktu yang sudah di tentukan sehingga dapat mengatur valve inlet, valve kondensate, valve exhause. Dimana sistem automatisasi ini memanfaatkan metode algoritma greedy. Dan pada penelitian ini akan di lakukan implementasi, pengujian dan hasil sistem automasisasi, pada sebuah prototype yang memanfaatkan mikrokontroler NodeMCU (esp32), Web Thinger IO, dan sensor tekanan.

Keywords: *sterilizer, algoritma greedy, NodeMCU ,esp32*

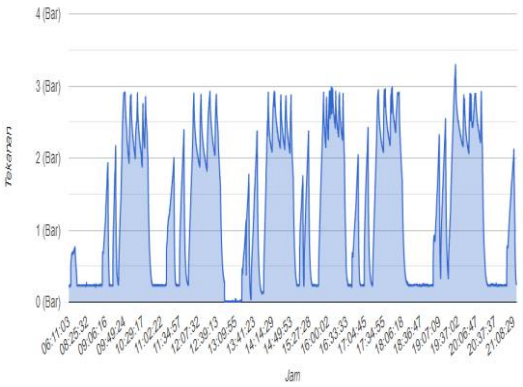
PENDAHULUAN

Kelapa sawit merupakan tanaman perkebunan penting penghasil minyak makanan, minyak industri dan bahan bakar nabati *(biodisel)*. Perkebunan kelapa sawit merupakan salah satu komoditas perkebunan yang berperan penting dalam perekonomian Indonesia. Menurut penelitian yang dilakukan oleh [1] Pada tahun 2020 PT.Perkebunan Nusantara V, anak perusahaan *Holding* perkebunan nusantara yang berlokasi di Provinsi Riau, berhasil meningkatkan produksi Tandan Buah Segar (TBS) kelapa sawit, mencapai rata-rata 23,9 ton per *hektare*. Tingkat produksi Tandan Buah Segar (TBS) pada posisi tersebut ialah tertinggi. Peningkatan produksi juga mempengaruhi jumlah produksi nasional Indonesia menurut penelitian yang dilakukan oleh [2].

Stasiun perebusan adalah stasiun inti dari pengolahan tandan buah segar (TBS) kelapa sawit. Stasiun perebusan atau sering disebut dengan *Sterilizer stastion* merupakan proses yang sangat menentukan untuk mendapatkan minyak sawit mentah / *crude palm oil* (CPO) yang baik dengan kualitas *rendemen* yang tinggi. Faktor yang mempengaruhi kesempurnaan proses perebusan adalah kondisi buah dan sistem perebusannya. Apabila dalam perebusan tidak memperhatikan tekanan, waktu dan *temperature* perebusan maka kehilangan minyak akan semakin besar [3]

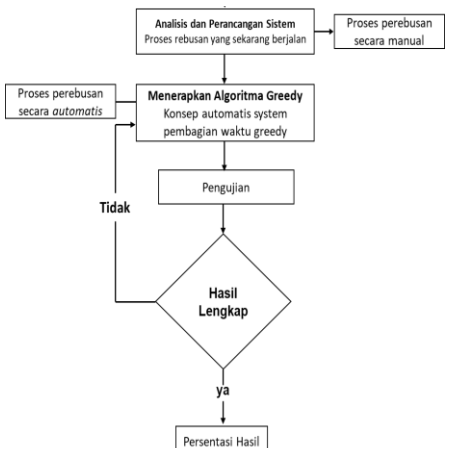
Proses perebusan saat ini masih dilakukan secara menual dengan mengatur valve inlet dan condesat untuk melakukan sterilisasi/perebusan tandan buah segar kelapa sawit. Kendala yang sering dihadapi adalah pembagian steam ke dalam tiga bejana rebusan belum maksimal. Dimana pengaturan pembagian steam ini hanya melalui satu pipa steam yang berasal dari sisa pembuangan turbin uap. Pada proses penentuan tekanan (*steam*) tidak sesuai dengan *procedure* perebusan masih menggunakan perasaan, jadi tekanan (*steam*) tidak dapat digunakan dengan maksimal. Banyaknya tenaga kerja yang sudah menjajaki pensiun maka sistem perebusan sering didapati beroperasi tidak sesuai standarnya. Sehingga mengakibatkan kehilangan minyak "losses" yang tinggi. Hal ini terjadi akibat kurangnya komitmen waktu dan pembagian tekanan (*steam*) dalam hal perebusan. Pengaturan steam uap yang tidak optimal akan menghasilkan loosis yang tinggi menurut penelitian yang dilakukan oleh [4]

Guna mengatasi permasalahan yang terjadi pada tidak komitmennya pembagian tekanan steam untuk mendukung system rebusan dengan triple peak pada 3 bejana rebusan maka akan diterapkan suatu formula Algoritma Greedy untuk mendapatkan solusi optimum untuk membagi steam pada tiga bejana rebusan. Algoritma *Greedy* merupakan salah satu metode yang popular untuk memecahkan persoalan optimasi. Yang di maksud dengan memecahkan persoalan optimasi adalah mencari solusi paling optimum dari segala kemungkinan yang ada.[5]



METODE PENELITIAN

Metode penilitian yang dilakukan untuk membuat sistem automasi rebusan tandan buah segar (TBS) kelapa sawit, menggunakan kerangka kerja yang sering dinamakan dengan diagram alur. Kerangka kerja ini dapat digunakan untuk *roadmap* penelitian yang akan dilakukan oleh peneliti.



Gambar 1 Framework Penelitian

1. Analisis Dan Perancangan Sistem automatis

Pada tahap perancangan sistem *automatis* kontrol tertanam *(Embedded System)* Arduino IDE berbasis algoritma *Greedy*, sistem perebusan tandan buah segar (TBS) mengunakan proses perebusan tiga puncak (*tripple peak*). Berikut merupakan grafik tekanan rebusan *steam* uap pada stasiun rebusan Sei Pagar PTPN V.

Gambar 2 Rebusan *triple peak*

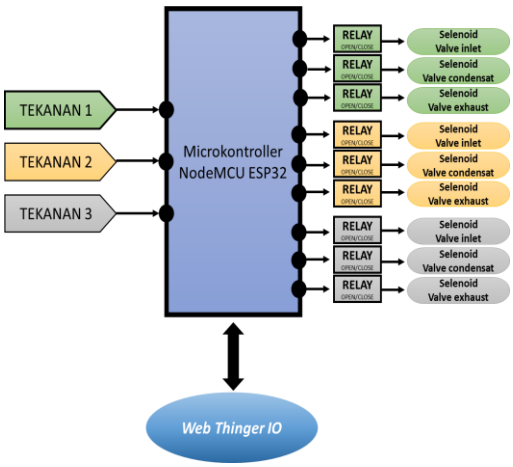
Pada proses puncak pertama waktu yang di butuhkan untuk memenuhi tekanan dari 0kg/cm^2 sehingga menjadi 1,5kg/cm^2 dalam waktu 8 menit, puncak kedua dari tekanan 0kg/cm^2 – 2,5kg/cm^2, dalam waktu 10 menit. Puncak ketiga dari tekanan 0kg/cm^2 – 3,0kg/cm^2, selama 13 menit. dengan pembagian waktu yang di tentukan 1, 3, 5, 10. Berapa jumlah minimum pembagian waktu tersebut sehingga mendapatkan solusi optimal?. Menggunakan tabel berikut ini

Tabel 1 Sistem Perebusan

Bejana Rebusan	Waktu Mulai Perebusan	Lama Proses Perebusan			Total Waktu Perebusan
		Proses Puncak I	Proses Puncak II	Proses Puncak III	
		(0-1,5 Kg/cm)	(0-2,5 Kg/cm)	(0-3,0 Kg/cm)	
Rebusan 1	07.00 WIB	07.00 - 07.10 WIB (10 Menit)	07.10 - 07.23 WIB (13 Menit)	07.23 - 08.30 WIB 13menit +54menit	90 Menit
Rebusan 2	07.30 WIB	07.25 - 07.35 WIB (10 Menit)	07.35 - 07.48 WIB (13 Menit)	07.48 - 08.55 WIB 13menit +54menit	90 Menit
Rebusan 3	08.00 WIB	07.50 - 08.00 WIB (10 Menit)	08.00 - 08.13 WIB (13 Menit)	08.13 - 09.20 WIB 13menit +54menit	90 Menit

2. Konsep Sistem Automatis

sistem otomatis menggunakan *board* mikrokontroler *NodeMCu Esp32*. Kelebihan menggunakan mikcrokontroler *NodeMCU esp32* adalah *prototype* yang kita buat dapat langsung terhubung dengan internet. Dimana pada microkontroler ini sudah terdapat *baord* IoT (esp32) yang akan menyambungkan data kita ke laman web ataupun *aplikasi home smart google*. Cara kerjanya sistem di atas yaitu akan menjalankan seluruh perintah proses rebusan yang telah di program, lalu data hasil proses akan tampilkan pada *laman web thinger io*. sensor tekanan 1, 2, dan 3 akan mendeteksi besar tekanan *steam* uap antara 0 – 3,0kg/cm^2 dan menghasilkan tegangan *output* sebesar 0 *volt* – 3*volt* DC. Tegangan *output* sensor tekanan, akan menjadi *input* atau masuk ke microkontroler untuk di proses oleh program yang sudah tertanam sistem perhitungan algoritma *greedy*, kemudian akan memerintahkan *relay* untuk bekerja *on/off*. Konsep system kendali automatis yang akan diterapkan disajikan pada gambar 2. Sistem kendali yang dilakukan dengan pengaturan dapat menggunakan algoritma greedy seperti yang dilakukan oleh [6]



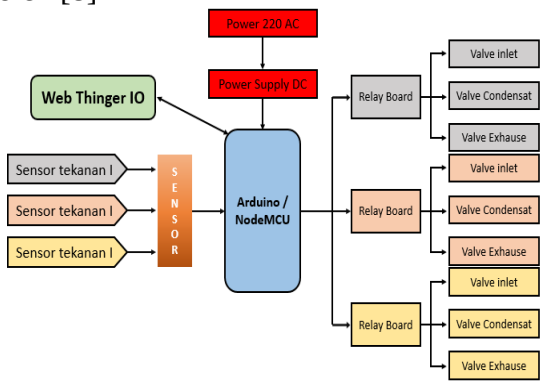
Gambar 2 Konsep I/O Esp 32

HASIL DAN PEMBAHASAN

Permasalahan yang di ambil pada penelitian ini yaitu tentang optimasi pengaturan *steam* uap pada perebusan buah kelapa sawit. Pengaturan *steam* uap ini memiliki tiga proses untuk sekali perebusan, setiap proses memiliki waktu dan tekanan yang sudah di tentukan sesuai *standar* perebusannya. Sehingga menghasilkan perebusan yang optimal. Dari permasalahan tersebut dibuatlah *prototipe system automatis* untuk mensimulasikan pembagian waktu dan tekakan *steam* uap. Kode pemrograman yang dibuat di jalankan menggunakan Arduino IDE. Algoritma yang digunakan adalah algoritma *Greedy*. Esp-32 adalah mikrokontroler 32 bit yang dapat deprogram dan menjalankan program dengan kecepatan sama dengan satu unit computer seperti penelitian yang dilakukan oleh [7] Esp-32 juga sangat memungkinkan digunakan untuk kendali jarak jauh berbasis SCADA.

1. Konfigurasi Pengujian

Pengujian sistem otomasi dilakukan dengan mengukur tegangan *output* pada setiap sensor tekanan. Serta menghubungkan antara *Web Thinger IO* dengan microkontroler. Web Thinger I/O adalah sebuah aplikasi berbasis web yang dapat digunakan untuk memantau kondisi variable yang terjadi pada pin ESP-32 seperti yang dilakukan oleh [8]



Gambar 3. Wiring diagram ESP32

2. Penentuan Solusi Optimum

Penentuan solusi optimum dilakukan untuk meminimalisir terjadinya dua energi (*Steam*) dalam satu waktu adapun perhitungan nya menggunakan Knapsack Problem menggunakan tabel berikut. Penelitian menggunakan Greedy dengan mempertimbangkan knapsack problem juga dilakukan oleh [9] yang memberikan kesimpulan yaitu algoritma greedy lebih tidak sesuai untuk dynamic programing namun untuk solusi optimum algoritma greedy baik.

Tabel 2 Knapsack Promblem

Solusi ke:	X1	X2	X3	∑ Wi Xi	∑ Pi Xi	status/solusi
1	0	0	0	90	0	tidak optimal
2	0	0	1	90	3	tidak optimal
3	0	1	0	90	2,5	tidak optimal
4	0	1	1	90	5,5	tidak optimal
5	1	0	0	90	1,5	tidak optimal
6	1	0	1	90	4,5	tidak optimal
7	1	1	0	90	4	tidak optimal
8	1	1	1	90	7	optimal

a. **Solusi ke 1:** $\sum W_i . X_i \leq M$

$= (W_1 \text{ x } X_1) + (W_2 \text{ x } X_2) + (W_3 \text{ x } X_3) \leq M$

$= (10 \text{ x } 0) + (13 \text{ x } 0) + (67 \text{ x } 0) \leq \mathbf{90}$

$= \mathbf{0 \leq 90}$

b. **Solusi ke 2:** $\sum W_i . X_i \leq M$

$= (W_1 \text{ x } X_1) + (W_2 \text{ x } X_2) + (W_3 \text{ x } X_3) \leq M$

$= (10 \text{ x } 0) + (13 \text{ x } 0) + (67 \text{ x } 1) \leq 90$

$= \mathbf{67 \leq 90}$

c. **Solusi ke 3:** $\sum W_i . X_i \leq M$

$= (W_1 \text{ x } X_1) + (W_2 \text{ x } X_2) + (W_3 \text{ x } X_3) \leq M$

$= (10 \text{ x } 0) + (13 \text{ x } 1) + (67 \text{ x } 0) \leq 90$

$= \mathbf{13 \leq 90}$

d. **Solusi ke 4:** $\sum W_i . X_i \leq M$

$= (W_1 \text{ x } X_1) + (W_2 \text{ x } X_2) + (W_3 \text{ x } X_3) \leq M$

$= (10 \text{ x } 0) + (13 \text{ x } 1) + (67 \text{ x } 1) \leq \mathbf{90}$

$= \mathbf{80 \leq 90}$

e. **Solusi ke 5:** $\sum W_i . X_i \leq M$

$= (W_1 \text{ x } X_1) + (W_2 \text{ x } X_2) + (W_3 \text{ x } X_3) \leq M$

$= (10 \text{ x } 1) + (13 \text{ x } 0) + (67 \text{ x } 0) \leq \mathbf{90}$

$= \mathbf{10 \leq 90}$

f. **Solusi ke 6:** $\sum W_i . X_i \leq M$

$= (W_1 \text{ x } X_1) + (W_2 \text{ x } X_2) + (W_3 \text{ x } X_3) \leq M$

$= (10 \text{ x } 1) + (13 \text{ x } 0) + (67 \text{ x } 1) \leq \mathbf{90}$

$= \mathbf{79 \leq 90}$

g. **Solusi ke 7:** $\sum W_i . X_i \leq M$

$= (W_1 \text{ x } X_1) + (W_2 \text{ x } X_2) + (W_3 \text{ x } X_3) \leq M$

$= (10 \text{ x } 1) + (13 \text{ x } 1) + (67 \text{ x } 0) \leq \mathbf{90}$

$= \mathbf{23 \leq 90}$

h. **Solusi ke 8:** $\sum W_i . X_i \leq M$

$= (W_1 \text{ x } X_1) + (W_2 \text{ x } X_2) + (W_3 \text{ x } X_3) \leq \mathbf{M}$

$= (10 \text{ x } 1) + (13 \text{ x } 1) + (67 \text{ x } 1) \leq \mathbf{90}$

$= \mathbf{90 \leq 90}$

Solusi optimal untuk perhitungan knapsack problem terdapat pada solusi ke delapan, yaitu proses pertama dijalankan, proses kedua dijalankan dan prosesketiga dijalankan. Jadi didapat hasil *visible solution* 90 dan fungsi *obyektif* 7. Proses perhitungan menggunakan cara matematika juga dilakukan oleh [10] yang memberikan kesimpulan perhitungan yang telah dilakukan yaitu untuk menyelesaikan permasalahan knapsack, maka dapat disimpulkan Knapsack problem dapat diselesaikan dengan cara matematika, kriteria greedy dan algoritma greedy.

3. Penerapan implementasi *Knapsack Problem*

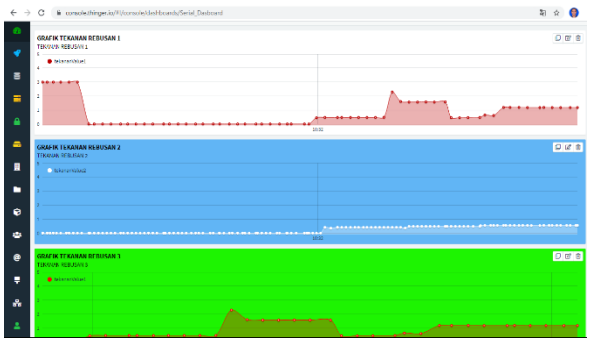
Implementasi menggunakan ESP32 guna mengendalikan valve inlet, Exhaust dan Condensate menggunakan penetuan waktu

yang telah dilakukan perhitungan menggunakan *algorithm greedy* sebagai berikut

Tabel 3 Implementasi Knapsack

Proses Rebusan	PUNCAK I						PUNCAK II						PUNCAK III					
REBUSAN 1	Action Relay < 1.5 Bar			Action Relay = 1.5 Bar			Action Relay < 2.5 Bar			Action Relay = 2.5 Bar			Action Relay < 3 Bar			Action Relay = 3 Bar		
	V-In	V-Exh	V-Kon	V-In	V-Exh	V-Kon	V-In	V-Exh	V-Kon	V-In	V-Exh	V-Kon	V-In	V-Exh	V-Kon	V-In	V-Exh	V-Kon
	1	0	0	0	1	1	1	0	0	0	1	1	1	0	0	0	1	1
	Nyala	Mati	Mati	Mati	Nyala	Nyala	Nyala	Mati	Mati	Mati	Nyala	Nyala	Nyala	Mati	Mati	Mati	Nyala	Nyala
Waktu >= 25 menit Lanjut Perebusan 2																		
REBUSAN 2	Action Relay < 1.5 Bar			Action Relay = 1.5 Bar			Action Relay < 2.5 Bar			Action Relay = 2.5 Bar			Action Relay < 3 Bar			Action Relay = 3 Bar		
	V-In	V-Exh	V-Kon	V-In	V-Exh	V-Kon	V-In	V-Exh	V-Kon	V-In	V-Exh	V-Kon	V-In	V-Exh	V-Kon	V-In	V-Exh	V-Kon
	1	0	0	0	1	1	1	0	0	0	1	1	1	0	0	0	1	1
	Nyala	Mati	Mati	Mati	Nyala	Nyala	Nyala	Mati	Mati	Mati	Nyala	Nyala	Nyala	Mati	Mati	Mati	Nyala	Nyala
Waktu >= 50 menit Lanjut Perebusan 3																		
REBUSAN 3	Action Relay < 1.5 Bar			Action Relay = 1.5 Bar			Action Relay < 2.5 Bar			Action Relay = 2.5 Bar			Action Relay < 3 Bar			Action Relay = 3 Bar		
	V-In	V-Exh	V-Kon	V-In	V-Exh	V-Kon	V-In	V-Exh	V-Kon	V-In	V-Exh	V-Kon	V-In	V-Exh	V-Kon	V-In	V-Exh	V-Kon
	1	0	0	0	1	1	1	0	0	0	1	1	1	0	0	0	1	1
	Nyala	Mati	Mati	Mati	Nyala	Nyala	Nyala	Mati	Mati	Mati	Nyala	Nyala	Nyala	Mati	Mati	Mati	Nyala	Nyala

Simulasi grafik dan data menggunakan *Web Thinger I/O* sehingga dihasilkan bentuk grafik dari 3 rebusan itu berbasis web. Aplikasi kendali automatis berbasis algoritma Greedy adalah suatu aplikasi yang tertanam pada chip ESP-32 dan dapat bekerja secara automatis. Sedangkan aplikasi Web Thinger I/O hanya digunakan untuk melihat kondisi Valve dan Tekanan yang masuk pada bejana yang disimulasikan menggunakan potensio meter grafik yang ditampilkan pada pengujian ini disajikan pada gambar 3.



Gambar 3. Grafik dan Data ESP32

Pada proses penelitian dan pengujian penerapan Algoritma Greedy yang ditanamkan pada ESP32 dapat berjalan dengan normal. ESP32 sendiri memiliki processor 32Bit sehingga proses data lebih maksimal.

SIMPULAN DAN SARAN

1. Simpulan

Komputasi pada Algoritma Greedy dapat diterapkan pada ESP 32 untuk mengendalikan proses kendali rebusan dengan perhitungan Knapsack problem pada optimasi menggunakan algoritma *greedy* didapat solusi optimal dengan "solusi ke 8" (90≤90), dimana semua objek dijalankan tanpa ada yang kurang ataupun berlebih dengan kapasitas knapsack yang sudah di tentukan

2. Saran

Meskipun pada simulasi prototype ESP32 dan Algoritma Greedy dapat digunakan untuk mendukung proses perebusan Tandan Buah Segar kelapa sawit namun perlu pertimbangan Fungsional untuk dapat menganti ESP32 menggunakan PLC yang sudah memiliki standart industry.

TERIMA KASIH

Ucapan terimakasih kepada seluruh karyawan/wati PT.Perkebunan Nusantara V terutama pada unit Bisnis pabrik dan kebun Sei Pagar. Ucapan terimakasih juga diucapkan kepada tim Kendali stasiun Rebusan (Sterilizer Controller).

DAFTAR PUSTAKA

[1] D. N. Pitriani ,H.Edison, "Jurnal Agri Sains Vol, 3 No.02, (28 Desember 2019)," no. 02, pp. 1–12, 2019.

[2] H. L. Nainggolan, C. K. Gulo, W. S. S. Waruwu, T. Egentina, and T. P. Manalu, "Strategi Pengelolaan Usahatani Kelapa Sawit Rakyat Masa Pandemi Covid-19 di Kecamatan STM Hilir Kabupaten Deli Serdang, Provinsi Sumatera Utara, Indonesia," *Agro Bali Agric. J.*, vol. 4, no. 2, pp. 260–275, 2021, doi: 10.37637/ab.v4i2.724.

[3] Saparudin and A. Fadlly, "Analisis Energi Strerilizer Dalam Proses Perebusan Kelapa Sawit Di PT Perkebunan Nusantara 1 PKS Tanjung Seumantoh," *Hadron J. Fis. dan Terap.*, vol. 1, no. 01, pp. 22–24, 2019.

[4] Sulaiman and R. Randa, "Pengaruh Temperatur Terhadap Efisiensi Sterilizer Dan Kualitas Minyak

Yang Dihasilkan," *Menara Ilmu*, vol. XII, no. 10, pp. 1–8, 2018.

[5] R. P. S. Rianti, H. Lubis, "Sistem Penunjang Keputusan Optimalisasi Baranga Dengan Algoritma Greedy Pada PT Sentralindo Teguh Gemilang," 2015.

[6] S. G. Wiratama, C. Christian, F. Johanna, J. Gonardi, and K. Purnomo, "Penerapan Algoritma Greedy Pada Pengaturan Shipping Buku Diknas PT. X," *J. Rekayasa Sist. Ind.*, vol. 5, no. 01, p. 23, 2018, doi: 10.25124/jrsi.v5i01.309.

[7] L. O. Aghenta and M. T. Iqbal, "Low-cost, open source IoT-based SCADA system design using thinger.IO and ESP32 thing," *Electron.*, vol. 8, no. 8, pp. 1–24, 2019, doi: 10.3390/electronics8080822.

[8] R. S. T. Atmojo, "Rancang Bangun Pemantauan Proses Dekomposisi Pupuk Kompos Berbasis Low Cost & Multi Point Modul Board," vol. 4, no. 1, pp. 174–179, 2019.

[9] G. I. Sampurno, E. Sugiharti, and A. Alamsyah, "Comparison of Dynamic Programming Algorithm and Greedy Algorithm on Integer Knapsack Problem in Freight Transportation," *Sci. J. Informatics*, vol. 5, no. 1, p. 49, 2018, doi: 10.15294/sji.v5i1.13360.

[10] D. Supriadi, "Indonesian Journal on Computer and Information Technology Vol 1 No 2 November 2016," *Indones. J. Comput. Inf. Technol.*, vol. 1, no. 2, pp. 91–99, 2016.