

PERANCANGAN DAN IMPLEMENTASI SISTEM MONITORING DAN PEMBERI PAKAN KUCING OTOMATIS BERBASIS ANDROID

DESIGN AND IMPLEMENTATION OF AUTOMATIC CAT MONITORING AND FEEDER SYSTEM BASED ON ANDROID

Muhammad Rizky Imam Pamungkas¹, Sony Sumaryo², Agung Surya Wibowo³

^{1,2,3}Prodi S1 Teknik Elektro, Fakultas Teknik Elektro, Universitas Telkom
¹rzkyimamp@gmail.com, ²sonysumaryo@telkomuniversity.ac.id, ³agungsw@telkomuniversity.ac.id

Abstrak

Kucing merupakan salah satu hewan peliharaan yang paling populer di dunia. Bagi pemilik hewan peliharaan seperti kucing biasanya memiliki kendala dalam pemberian pakan hewan mereka secara rutin dikarenakan rutinitas mereka di luar rumah, pemberian pakan hewan yang tidak teratur serta hewan lapar dalam jangka waktu yang lama akan menyebabkan hewan akan memiliki penyimpangan substansial dalam parameter fisik dan perilaku yang terkait dengan stress atau tertekan.

Pada tugas akhir ini, penulis bertujuan untuk merancang dan membuat sebuah prototype sistem pemantauan dan pemberi pakan otomatis kucing dengan menggunakan algoritma penjadwalan *non-preemptive* dan mikrokontroler *nodeMCU* yang terintegrasi dengan *smartphone android* menggunakan fitur *IoT*.

Keluaran yang didapatkan dari tugas akhir ini yaitu dapat mengontrol dan memonitoring pakan kucing dari jauh yang bisa diakses dimana saja dan kapan saja melalui *smartphone android*.

Kata Kunci : *Android, IoT, NodeMCU, Non-preemptive, Smartphone*

Abstract

Cat is one of the most popular pets in the world. Cat owners usually having constraints in feeding the cat regularly due to their outdoor activities. Irregular feeding and also starvation will cause the animal will have a substantial deviation in physical parameters and behaviors associated with stress or depressed.

In this final project, the author aims to designs and creates a prototype of monitoring and automatic cat feeding system using scheduling non-preemptive algorithm and nodeMCU microcontroller integrated with android smartphone using the IoT feature.

The output obtained from this final project is to control and monitor the cat feeding from far away that can be accessed anywhere and anytime through the android smartphone.

Keywords: *Andorid, IoT, NodeMCU, Non-preemptive, Smartphone*

1. Pendahuluan

Kucing merupakan salah satu hewan peliharaan yang paling populer di dunia. Bahkan kucing peliharaan sudah biasa dianggap seperti anggota keluarga sendiri bagi sang pemilik. Memelihara seekor kucing memerlukan sebuah kedisiplinan dan ketekunan baik dalam merawat kebersihan kandang, dan juga dalam pemberian pakan.

Bagi pemilik hewan peliharaan seperti kucing, biasanya memiliki kendala didalam pemberian pakan hewan mereka secara rutin dikarenakan rutinitas kegiatan mereka diluar rumah, sehingga pakan kucing di setiap harinya tidak dapat terkontrol dengan baik dan dapat membuat kucing jadi lebih mudah terserang penyakit.

Pemberian pakan hewan yang tidak teratur serta hewan lapar dalam jangka waktu yang lama dapat menyebabkan hewan memiliki penyimpangan sustansial dalam parameter fisik dan perilaku yang terkait dengan stress atau tertekan[1]. Untuk dapat memberi pakan hewan secara teratur tanpa harus mengganggu aktivitas pemilik setiap hari maka dibutuhkan sebuah alat pemberi pakan otomatis yang dapat memberikan pakan hewan secara teratur dan dapat juga di kontrol dari jarak jauh oleh sang pemilik.

Pada tugas akhir ini penulis akan membuat sebuah sistem pemberi pakan kucing secara otomatis menggunakan algoritma *Non-preemptive* dan mikrokontroler *NodeMCU* yang dapat di kontrol dari jarak jauh menggunakan *smartphone* melalui sebuah aplikasi berbasis *Android* yang memiliki fitur-fitur seperti memonitoring sisa pakan yang ada, mengontrol persediaan pakan dan juga dapat memberikan pakan kepada kucing peliharaan kita secara otomatis dan berkala kapanpun dimanapun dan dari jarak yang jauh sekalipun.

2. Dasar Teori

2.1 Pakan Kucing

Kucing mempunyai pola dan perilaku makan yang sangat spesifik. Kucing makan sekitar 12-18 kali sehari, dengan total waktu makan sekitar 30 menit/hari. Kucing hanya makan beberapa gram makanan setiap kali makan [2].

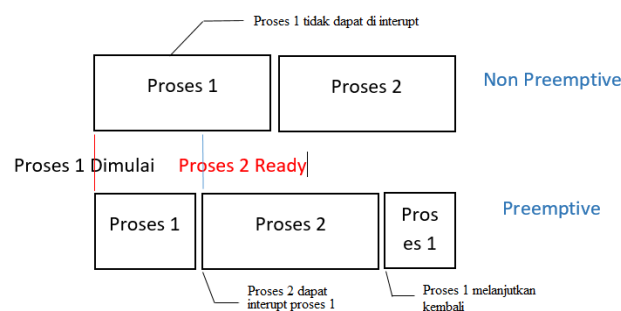
Indera pengecap menjadi sangat penting dan merupakan mekanisme hewan memilih pakan untuk memenuhi kebutuhan hidupnya. Penelitian Hewson-Hughes *et al* (2011) memberikan gambaran yang sangat jelas mengenai kebutuhan nutrisi kucing, khususnya makronutrisi protein, lemak dan karbohidrat. Dalam sehari seekor kucing membutuhkan sekitar 26 gram protein, 9 gram lemak dan 8 gram karbohidrat yang mana unsur tersebut setara dengan kebutuhan kalori sebanyak 52% dari protein, 36% dari lemak dan 12% dari karbohidrat.

Beberapa penelitian juga menuturkan bahwa seekor kucing yang hidup di dalam rumah hanya menghabiskan sedikit energy. Kucing dengan berat badan ± 3.5 kg, hanya memerlukan asupan normal sekitar 50 gram makanan kering/harinya[2].

2.2 Algoritma Non Preemptive Jenis FIFO

Non preemptive adalah gaya multitasking computer dimana sistem operasi tidak pernah memulai *context switch* dari proses yang sedang berjalan ke proses lain. Sistem seperti ini biasa digunakan pada penjadwalan statis, paling sering pada sistem periodic. Ketika *non preemptive* digunakan, sebuah proses yang menerima sumber daya tersebut tidak dapat terganggu sampai selesai. Penjadwalan *non preemptive* telah menerima cukup kurang perhatian di komunitas riset dibandingkan dengan *preemptive*. Namun, penjadwalan *non preemptive* banyak digunakan dalam industri praktek, dan itu mungkin lebih baik daripada penjadwalan *preemptive* untuk beberapa alasan[3].

Dalam komputasi, *context switch* adalah proses penyimpanan dan memulihkan keadaan (konteks) dari suatu proses atau *thread* sehingga eksekusi yang dapat dilanjutkan dari titik yang sama dilain waktu. Penjadwalan *non preemptive* ialah penjadwalan dimana sistem operasi tidak melakukan *context switch* dari proses yang sedang berjalan ke proses lain (proses yang berjalan tidak bisa di-*interrupt*).



Gambar II-1 Perbedaan Non Preemptive dengan Preemptive dalam chart

Jenis penjadwalan *non preemptive* yang dipakai adalah penjadwalan jenis FIFO yaitu *First In First Out* merupakan penjadwalan yang tidak berprioritas. Proses penjadwalan FIFO memiliki ketentuan-ketentuan yang sederhana, proses-proses diberi jatah waktu yang diurutkan berdasarkan waktu kedatangan proses-proses itu ke sistem. Begitu proses mendapatkan jatah waktu pemrosesan, proses dijalankan sampai selesai.

2.3 Mikrokontroler

Mikrokontroler adalah sebuah sistem mikroprosesor dimana di dalamnya sudah terdapat CPU, ROM (Read Only Memory), RAM (*Random Acces Memory*), antar muka *input-output (I/O interface)*, *clock*, dan peralatan internal lainnya yang sudah saling terhubung dan terorganisasi dengan baik oleh pabrik pembuatnya dan dikemas dalam satu chip yang siap pakai. Dengan demikian dapat langsung deprogram isi ROM sesuai aturan penggunaan oleh pabrik yang membuatnya (Bejo,2008) [5].

2.4 IoT

Internet of Things (IoT) adalah jaringan yang terdapat pada perangkat fisik, kendaraan, peralatan rumah tangga, dan barang-barang lainnya yang dilengkapi dengan sistem elektronik, perangkat lunak, sensor, actuator, dan konektivitas yang memungkinkan saling terjadinya pertukaran data. IoT dapat menciptakan banyak peluang untuk integrasi langsung dunia fisik ke dalam sistem berbasis komputer, yang menghasilkan peningkatan efisiensi, keuntungan ekonomi, dan pengurangan tenaga manusia[7].

2.5 Android

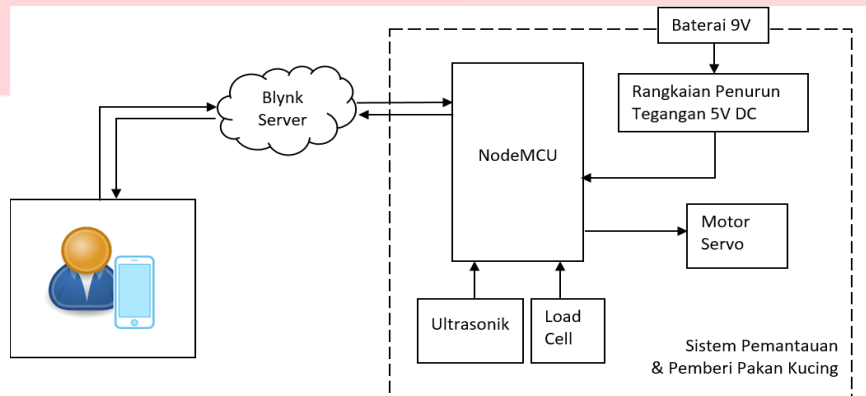
Android adalah platform *open source* yang komprehensif dan dirancang untuk *mobile devices*. Dikatakan komprehensif karena Android menyediakan semua tools dan frameworks yang lengkap untuk pengembangan aplikasi pada suatu *mobile device*. Sistem Android menggunakan database untuk menyimpan informasi penting yang diperlukan agar tetap tersimpan meskipun *device* dimatikan[6].

Dikembangkan bersama Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, NVIDIA yang tergabung dalam OHA (Open Handset Alliance) dengan tujuan membuat sebuah standar terbuka untuk perangkat bergerak[4].

3. Perancangan Sistem

3.1. Desain Sistem

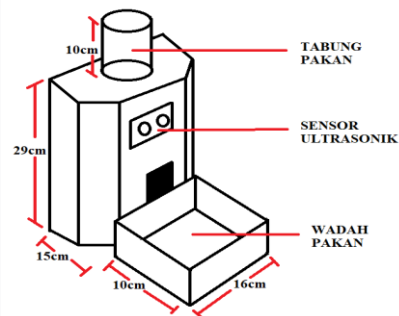
Pada tugas akhir ini akan dirancang sebuah sistem yang bertujuan untuk memonitoring dan memberi pakan otomatis pada kucing peliharaan di rumah yang dapat dimonitoring dan di kontrol menggunakan Android.



Gambar III- 1 Blok Diagram Sistem

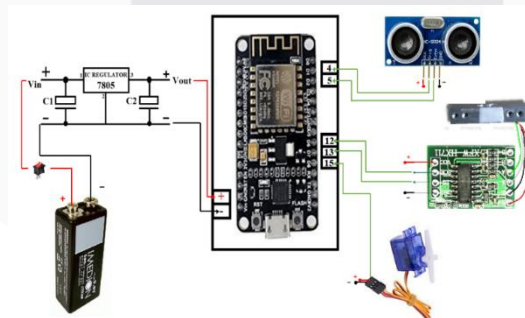
3.2. Desain dan Perancangan Perangkat Keras

Dalam pemilihan dan pembuatan mekanik sistem. Penulis memilih kayu sebagai bahan utama untuk pembuatan box dan *packaging* dari alat ini. Dengan pertimbangan sifatnya yang mudah dibentuk sesuai yang diinginkan, bersifat isolator, dan juga harga yang cukup terjangkau.



Gambar III- 2 Rancangan Alat Pemberi Pakan Kucing

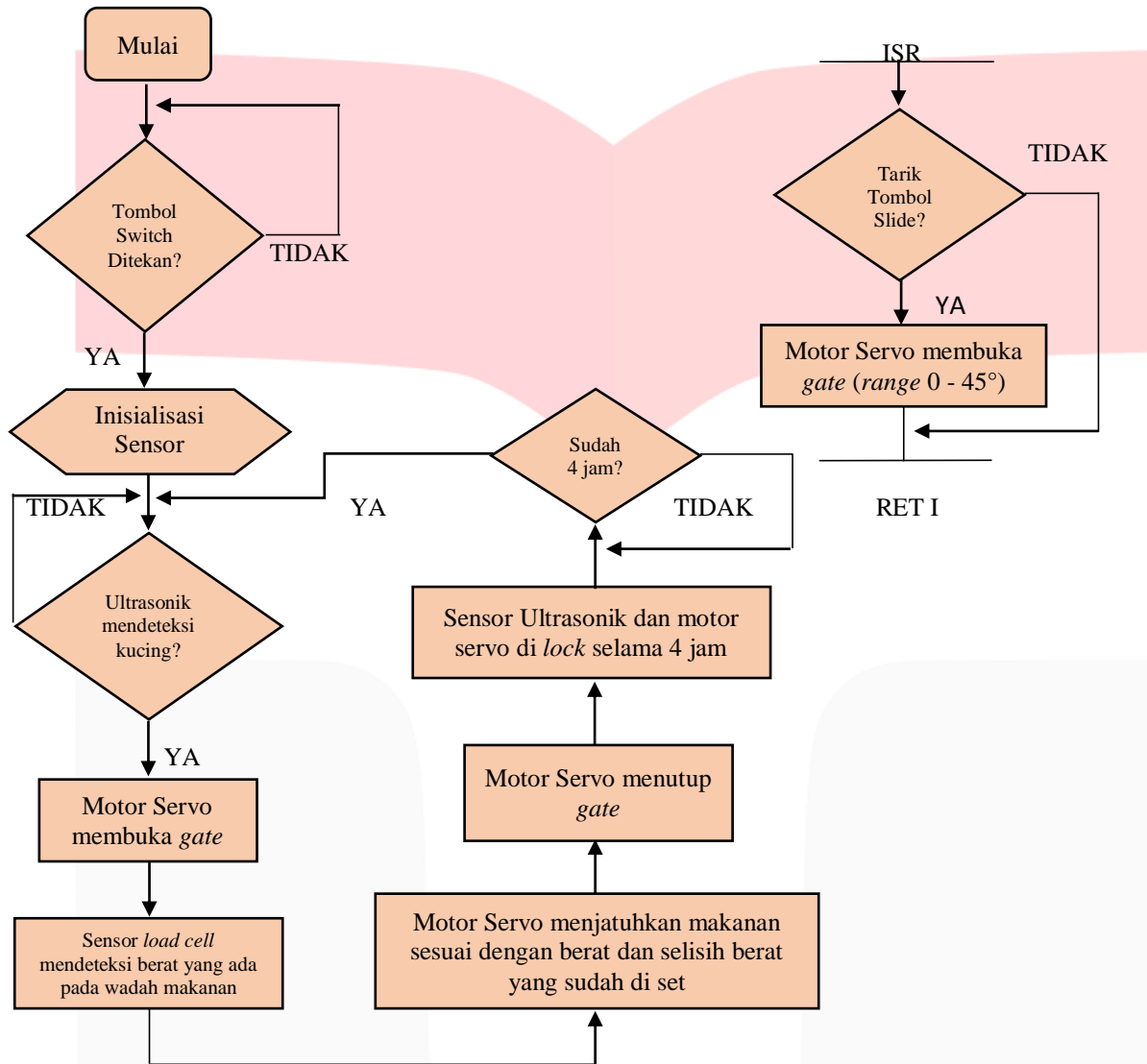
Adapun pada sub bab ini juga dilengkapi dengan skematik wiring komponen pada NodeMCU yang dapat dilihat pada Gambar III-3.



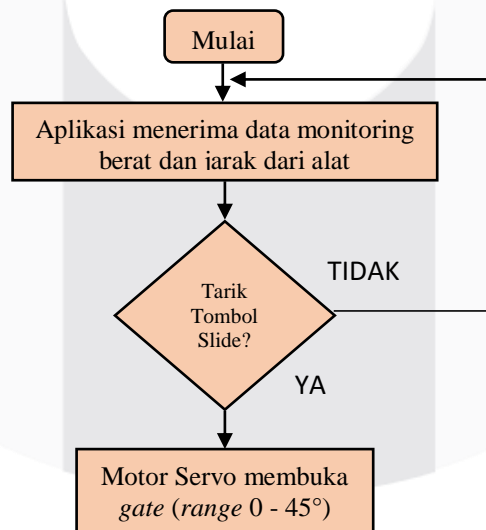
Gambar III-3. Wiring Komponen pada NodeMCU

3.3. Perancangan Perangkat Lunak

Berikut merupakan diagram alur pada sistem:



Gambar III- 4 Flowchart Sistem



Gambar III-5 Flowchart Aplikasi

4. Hasil Pengujian dan Analisis

4.1 Hasil Pengujian Pengukuran Berat Pada Sensor Load Cell

Sensor Load Cell merupakan sebuah sensor berat yang dapat mengukur berat pada benda apapun dan dapat di set satuannya sesuai yang dibutuhkan. Pengujian ini bertujuan untuk mendapatkan nilai akurasi dan presisi dengan cara membandingkan data yang diterima pada sensor Load Cell dengan timbangan digital. Pengujian ini dilakukan dengan cara memasukan makanan pada tabung penyimpanan makanan yang lalu di *trigger* oleh sensor ultrasonic yang membuat servo menjatuhkan makanan tepat didalam wadah makanan yang berada diatas sensor Load Cell. Setelah itu data nya dibandingkan dengan pengukuran berat pada timbangan digital.

Pada parameter pengujian ini akan dilakukan dua kali percobaan pengukuran berat yaitu pada berat 50gr dan pada berat 100gr.

Tabel IV-1 Pengujian Berat 50gr

No	Timbangan Digital (gr)	Alat (gr)	Error (gr)	Error (%)	No	Timbangan Digital (gr)	Alat (gr)	Error (gr)	Error (%)
1	50	51	-1	2%	16	50	50	0	0%
2	50	50	0	0%	17	50	50	0	0%
3	50	51	-1	2%	18	50	53	3	6%
4	50	48	-2	4%	19	50	50	0	0%
5	50	45	-5	10%	20	50	51	1	2%
6	50	50	0	0%	21	50	51	1	2%
7	50	51	1	2%	22	50	47	-3	6%
8	50	50	0	0%	23	50	47	-3	6%
9	50	50	0	0%	24	50	50	0	0%
10	50	50	0	0%	25	50	49	-1	2%
11	50	54	4	8%	26	50	50	0	0%
12	50	50	0	0%	27	50	48	-2	4%
13	50	48	-2	4%	28	50	52	2	4%
14	50	50	0	0%	29	50	50	0	0%
15	50	47	-3	6%	30	50	49	-1	2%
Rata-Rata Nilai Error (%)					2				
Rata-Rata Nilai dari Alat (gr)					49.67				
Presentasi Error Maksimal					8%				
Varians					3.374				
Standar Deviasi					1.837				
Konstanta Variansi					3.766%				

Dari data pengujian pada Tabel IV-1 dapat dihitung nilai akurasinya sebesar:

$$\begin{aligned} \text{Akurasi} &= 100\% - \text{presentase nilai error} \\ &= 100\% - 2\% \\ &= 98\% \end{aligned}$$

Lalu nilai presisinya didapat dari nilai konstanta variasi sebesar 3.766% atau diatas 2% yang menandakan pengujian ini masih kurang presisi, namun walaupun dalam teori pengujian ini masih kurang presisi tetapi sama sekali tidak berpengaruh dengan jalannya alat dan masih dapat digunakan seperti yang diinginkan.

Tabel IV-2 Pengujian Berat 100gr

No	Timbangan Digital (gr)	Alat (gr)	Error (gr)	Error (%)	No	Timbangan Digital (gr)	Alat (gr)	Error (gr)	Error (%)
1	100	107	7	7%	16	100	102	2	2%
2	100	102	2	2%	17	100	97	-3	3%
3	100	100	0	0%	18	100	91	-8	8%
4	100	100	0	0%	19	100	105	5	5%
5	100	101	1	1%	20	100	100	0	0%
6	100	100	0	0%	21	100	99	-1	1%

7	100	99	-1	1%	22	100	99	-1	1%
8	100	104	4	4%	23	100	100	0	0%
9	100	96	-4	4%	24	100	102	2	2%
10	100	98	-2	2%	25	100	100	0	0%
11	100	100	0	0%	26	100	104	4	4%
12	100	100	0	0%	27	100	98	-2	2%
13	100	100	0	0%	28	100	99	-1	1%
14	100	101	1	1%	29	100	100	0	0%
15	100	100	0	0%	30	100	100	0	0%
Rata-Rata Nilai Error (%)					2				
Rata-Rata Nilai dari Alat (gr)					100.13				
Presentasi Error Maksimal					7%				
Varians					8.188				
Standar Deviasi					2.862				
Konstanta Variansi					2.858%				

Dari data pengujian pada Tabel IV-2 dapat dihitung nilai akurasi sebesar:

$$\begin{aligned} \text{Akurasi} &= 100\% - \text{presentase nilai error} \\ &= 100\% - 2\% \\ &= 98\% \end{aligned}$$

Lalu nilai presisinya didapat dari nilai konstanta variasi sebesar 2.856% atau diatas 2% yang menandakan pengujian ini masih kurang presisi, namun walaupun dalam teori pengujian ini masih kurang presisi tetapi sama sekali tidak berpengaruh dengan jalannya alat dan masih dapat digunakan seperti yang diinginkan.

4.2 Hasil Pengujian Monitoring Berat Pakan Didalam Kandang Kucing

Pada pengujian ini alat akan ditaruh dan diimplementasikan di dalam kandang kucing selama 8 jam perhari dan berat makanan pada alat akan dimonitoring setiap 30 menit sekali dalam 8 jam. Tujuan dari pengujian ini adalah agar mengetahui ketahanan alat jika ditaruh didalam kandang kucing dan untuk memonitoring berat. Pada pengujian ini alat akan ditaruh di dalam kandang kucing mulai pukul 8.00 pagi hingga pukul 16.00 sore.

Tabel IV-3 Hasil Monitoring pada Hari Ke-1

Jam	Berat Makanan Pada Wadah (gr)	Jam	Berat Makanan Pada Wadah (gr)
8:00	0	12:30	53
8:30	48	13:00	52
9:00	32	13:30	52
9:30	30	14:00	46
10:00	27	14:30	40
10:30	23	15:00	40
11:00	20	15:30	40
11:30	26	16:00	34
12:00	22		

Tabel IV-4 Hasil Monitoring pada Hari Ke-2

Jam	Berat Makanan Pada Wadah (gr)	Jam	Berat Makanan Pada Wadah (gr)
8:00	0	12:30	52
8:30	41	13:00	51
9:00	39	13:30	42
9:30	30	14:00	38
10:00	34	14:30	39
10:30	33	15:00	32
11:00	36	15:30	32
11:30	30	16:00	27
12:00	24		

Pada pengujian hari ke-1 didapati 1 nilai error yaitu data ke 8 pada pukul 11.30. Berat pakan yang seharusnya tidak mengalami kenaikan dikarenakan masih dalam masa delay 4 jam. Selain itu pada data ke 10,11,12 pukul 12:30,13:00,13:30 juga terdapat error penimbangan sebesar 3gr,2gr,2gr. Dan pada pengujian di hari ke-2 didapati 3 nilai error yaitu data ke 5,7,12 pada pukul 10:00,11:00,14:30. Berat makanan yang seharusnya tidak mengalami kenaikan dikarenakan masih dalam masa delay 4 jam. Selain itu pada data ke 10 dan 11 pukul 12:30 dan 13:00 juga terdapat error sebesar 2gr dan 1gr.

4.3 Pengujian Delay

Pada pengujian ini akan diuji nilai delay yang ada pada saat pengontrolan Motor Servo di fitur *emergency slide* yang ada pada aplikasi di *smartphone*. Akan ada 2 pengujian dan 30 data delay yang diambil, yaitu 15 data delay pada saat *emergency slide* ditarik saat kondisi Motor Servo berada di sudut 0° - 45° (katup pakan terbuka) dan 15 data delay pada saat kondisi Motor Servo berada di sudut 45° - 0° (katup pakan tertutup).

Tabel IV-5 Data Hasil Pengujian Delay 0° - 45°

no	Waktu Geser	Waktu Terbuka	Delay/Selisih (s)
1	08:30:05:74	08:30:06:57	0,83
2	08:30:13:16	08:30:13:52	0,36
3	08:30:17:88	08:30:18:32	0,44
4	08:30:23:89	08:30:24:02	0,13
5	08:30:29:16	08:30:29:89	0,73
6	08:30:34:24	08:30:34:56	0,32
7	08:30:41:04	08:30:41:67	0,63
8	08:30:48:73	08:30:49:86	1,13
9	08:30:54:12	08:30:54:56	0,44
10	08:31:00:09	08:31:00:56	0,47
11	08:31:05:82	08:31:06:22	0,40
12	08:31:11:29	08:31:12:16	0,87
13	08:31:16:53	08:31:17:09	0,56
14	08:31:21:04	08:31:21:56	0,52
15	08:31:27:83	08:31:28:23	0,40
	Rata-Rata		0,54
	Max		1,13
	Min		0,13

Pada data diatas dapat disimpulkan bahwa nilai delay rata-rata pada saat android mengontrol Motor Servo saat sudut 0° - 45° memiliki nilai sebesar 0,54 detik yang berarti hanya dibutuhkan waktu setengah detik untuk mengontrol Motor Servo agar katup pakan terbuka secara penuh.

Tabel IV-6 Data Hasil Pengujian Delay 45° - 0°

no	Waktu Geser	Waktu Terbuka	Delay/Selisih (s)
1	08:30:08:92	08:30:10:12	1,20
2	08:30:15:42	08:30:15:82	0,40
3	08:30:20:00	08:30:20:51	0,51
4	08:30:26:02	08:30:26:46	0,42
5	08:30:31:49	08:30:32:26	0,77
6	08:30:38:02	08:30:39:39	1,37
7	08:30:44:08	08:30:45:17	1,09
8	08:30:50:84	08:30:52:19	1,35
9	08:30:57:06	08:30:58:02	0,96
10	08:31:02:76	08:31:03:98	1,22
11	08:31:08:63	08:31:09:78	1,15
12	08:31:14:30	08:31:14:69	0,39
13	08:31:19:96	08:31:20:32	0,36
14	08:31:24:78	08:31:25:32	0,64
15	08:31:30:05	08:31:30:67	0,72
	Rata-Rata		0,83
	Max		1,37
	Min		0,36

Pada data diatas dapat disimpulkan bahwa nilai delay rata-rata pada saat android mengontrol Motor Servo saat sudut 45° - 0° memiliki nilai sebesar 0,83 yang berarti hanya dibutuhkan waktu kurang dari 1 detik untuk mengontrol Motor Servo agar katup pakan tertutup secara penuh.

5. Kesimpulan dan Saran

5.1. Kesimpulan

Berdasarkan pengujian dan analisa yang telah dilakukan, maka dapat diambil kesimpulan bahwa:

1. Mekanik dari alat dapat berfungsi dengan cukup baik.
2. Alat dapat dipantau dan dikontrol jarak jauh berat pakannya secara *real time* oleh *smartphone*.
3. Alat dapat menerapkan algoritma penjadwalan *Non Preemptive*.
4. Berdasarkan pengujian serta analisis dapat diketahui bahwa nilai akurasi sensor Load Cell pada berat 50gr dan 100gr sama-sama memiliki nilai akurasi sebesar 98%.
5. Pada pengujian monitoring di dalam kandang didapati 1 nilai error pada hari pertama pengujian dan 3 nilai error pada hari kedua pengujian.
6. Berdasarkan pengujian delay *emergency slide* pada sudut 0° - 45° hanya dibutuhkan waktu rata-rata pembukaan *gate* sebesar 0,54 detik dan pada pengujian delay sudut 45° - 0° hanya dibutuhkan waktu rata-rata penutupan *gate* sebesar 0,83 detik.

5.2 Saran

Berdasarkan pengujian dan analisa yang telah dilakukan, saran yang dapat diberikan diantaranya:

1. Untuk pengerjaan selanjutnya, dapat digunakan bahan *acrylic* agar alat bisa lebih kuat dan kokoh.
2. Menggunakan lebih dari satu sensor Load Cell agar penimbangan bisa lebih akurat dan merata.
3. Menggunakan kalman filter untuk pembacaan sensor ultrasonik agar menghilangkan *noise* pada pembacaan dan nilai yang fluktuatif

Daftar Pustaka

- [1] America Association of Feline Practitioners, Hill's. 2004. Feline Behavior Guidelines. Kansas
- [2] <https://royalcanin.co.id> [Diakses pada 12 November 2018]
- [3] K. Jeffay, D.F. Stanat, C.U. Martel On non-preemptive scheduling of periodic and sporadic tasks, in: Proceedings of the 12th IEEE Real-Time Systems Symposium (RTSS), 1991.
- [4] Mulyadi. *Membuat Aplikasi Untuk Android*. Yogyakarta: Multimedia Center Publishing, 2010.
- [5] R. T. AS SADAD, I. and . J. A. SADAD , "Implementasi Mikrokontroler Sebagai Pengendali Lift Empat Lantai," *JURNAL ILMIAH SEMESTA TEKNIKA* , vol. 14, no. 2, pp. 160-165, November 2011.
- [6] Setiawan, Iwan., Andjarwirawan, Justinus dan Handojo, Andreas., Aplikasi Makassar Tourism Pada Kota Makassar Berbasis Android, *Jurnal Infra*, 1 , 156, 2013.
- [7] Vermesan, Ovidiu; Friess, Peter (2013). *Internet of Things: Converging Technologies for Smart Environments and Integrated Ecosystems*. Aalborg, Denmark: River Publishers.