RANCANG BANGUN SISTEM KENDALI SUPERVISI BERBASIS *IoT* PADA *PROTOTYPE OVERHEAD CRANE*

Muhammad Khoirul Imam[1], Elang Derdian Marindani[2], Muhammad Saleh[3]
[1,2]Fakultas Teknik, Universitas Tanjungpura, Pontianak
[1]khoiruimam64@gmail.com

Abstrak

Overhead crane merupakan salah satu peralatan vital dalam industri manufaktur yang berfungsi sebagai alat angkat dan pemindah material. Pengoperasian *overhead crane* memiliki risiko kecelakaan kerja yang tinggi terutama akibat ayunan beban yang tidak terkendali dan kesalahan operator. Penelitian ini bertujuan untuk merancang dan membangun sistem kendali posisi dan peredam ayunan beban pada *prototype overhead crane* menggunakan pengendali *PID* serta membangun aplikasi monitoring dan kendali supervisi berbasis Android menggunakan *MIT App Inventor*. Sistem menggunakan Arduino Mega 2560 sebagai pengendali utama, ESP32 sebagai modul komunikasi *IoT*, dan Arduino Pro Micro untuk pembacaan sensor MPU6050. Kendali posisi menggunakan *encoder* motor JGA25-370 sebagai umpan balik dengan nilai konversi 0,0085 cm/pulse untuk sumbu X dan 0,0098 cm/pulse untuk sumbu Y. Komunikasi antara aplikasi dan sistem menggunakan protokol *MQTT* dengan *broker* Eclipse *Mosquitto*. Hasil pengujian menunjukkan bahwa kendali *PID anti-swing* berhasil meredam ayunan beban secara signifikan yang dibuktikan dari perbandingan mode otomatis *PID ON* dan *PID OFF* dimana *MAE* posisi sumbu Y pada kondisi *PID ON* sebesar 3,31 cm sedangkan pada kondisi *PID OFF* meningkat drastis menjadi 27,23 cm. Aplikasi Android yang dikembangkan berhasil menampilkan data monitoring dan memungkinkan operator melakukan kendali supervisi perpindahan beban dari jarak jauh menggunakan smartphone.

Kata Kunci: *Overhead crane*, *PID*, *Anti-swing*, *IoT*, *MQTT*, *MIT App Inventor*.

Article History
Received: July 2026
Reviewed: July 2026
Published: July 2026

Plagiarism Checker No 237.u873247.fh8374
Copyright : Author
Publish by : Kohesi

This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License

Abstract

Overhead crane is one of the vital equipment in the manufacturing industry that functions as a material lifting and transferring tool. The operation of overhead cranes carries a high risk of work accidents, particularly due to uncontrolled load swinging and operator errors. This study aims to design and build a position control system and load swing damper ON an overhead crane prototype using a PID controller, as well as to develop an Android-based monitoring and supervisory control application using MIT App Inventor. The system uses Arduino Mega 2560 as the main controller, ESP32 as the IoT communication module, and Arduino Pro Micro for MPU6050 sensor reading. Position control uses the JGA25-370 motor encoder as feedback with a conversion value of 0.0085 cm/pulse for the X-axis and 0.0098 cm/pulse for the Y-axis. Communication between the application and the system uses the MQTT protocol with the Eclipse Mosquitto broker. The test results show that the PID anti-swing control successfully dampened load swinging significantly, as evidenced by the comparison of automatic mode PID ON and PID OFF conditions where the MAE of Y-axis position in PID ON condition was 3.31 cm while in PID OFF condition it increased drastically to 27.23 cm. The developed Android application successfully displayed monitoring data and allowed operators to perform supervisory control of load transfer remotely using a smartphone.

***Keywords**: Overhead crane, PID, Anti-swing, IoT, MQTT, MIT App Inventor.*

1. PENDAHULUAN

Latar Belakang

Industri manufaktur Indonesia dituntut meningkatkan produktivitas sekaligus menjamin keselamatan dan kesehatan pekerja. Kewajiban tersebut sejalan dengan Undang-Undang Nomor 1 Tahun 1970 tentang Keselamatan Kerja, termasuk dalam penggunaan mesin, peralatan, instalasi, dan proses produksi yang berpotensi menimbulkan bahaya.

Overhead crane digunakan dalam industri manufaktur dan konstruksi untuk memindahkan material berkapasitas besar melalui gerakan horizontal dan vertikal pada struktur *bridge* dan *trolley*. Peralatan ini memiliki kapasitas angkat besar, pergerakan cepat, dan biaya operasional relatif efisien. Meskipun demikian, pengoperasiannya mengandung risiko kecelakaan. Kurniawan (2023) mengidentifikasi faktor manusia, metode kerja, dan manajemen sebagai penyebab utama kecelakaan *lifting* menggunakan overhead crane. Analisis Pareto menunjukkan bahwa sekitar 80% permasalahan kecelakaan dipengaruhi oleh 20% faktor dominan, terutama kesalahan manusia.

Salah satu tantangan dalam pengoperasian *overhead crane* adalah ayunan beban akibat gaya inersia ketika motor mengalami percepatan atau perlambatan. Ayunan tersebut dapat mengurangi kestabilan beban dan memengaruhi ketepatan posisi peletakan. Rusli dan Isdayani mengembangkan sistem kendali posisi *overhead crane* dengan motor AC yang dilengkapi pengendali PID sebagai mekanisme peredam ayunan. Pengendali PID bekerja berdasarkan galat antara nilai aktual dan nilai yang diinginkan. Komponen proporsional mempercepat respons sistem, komponen integral mengurangi galat pada kondisi tunak, sedangkan komponen derivatif

menekan *overshoot* dan meningkatkan kestabilan respons. Atas dasar tersebut, pengendali PID digunakan dalam penelitian ini untuk mengendalikan posisi sekaligus menjaga ayunan beban selama proses perpindahan.

Sistem juga mengintegrasikan *Internet of Things* melalui protokol MQTT untuk komunikasi data, monitoring, dan kendali supervisi menggunakan aplikasi smartphone. Penelitian terdahulu telah membahas sistem overhead crane, tetapi integrasi kendali posisi berbasis encoder, peredam ayunan PID, dan komunikasi MQTT masih dinyatakan terbatas.

Beberapa penelitian sebelumnya telah mengembangkan sistem kendali crane dengan pendekatan yang berbeda. Pratama dan Sianturi merancang *overhead crane* otomatis berbasis Arduino yang digerakkan menggunakan motor *stepper* dan dikendalikan melalui Bluetooth. Putra, Herlambang, dan Nurdiansari mengembangkan sistem kendali *hoist crane* melalui aplikasi Android menggunakan MIT App Inventor. Parenrengi menerapkan kendali posisi berbasis encoder, fitur *anti-sway* menggunakan sensor IMU, serta *anti-collision* berbasis sensor LIDAR pada *rubber tyred gantry crane*. Rusli dan Isdayani menggunakan pengendali PID sebagai mekanisme peredam ayunan pada sistem kendali posisi *overhead crane*. Arrifai dan rekan-rekan membandingkan pengendali PID dan PD untuk sistem *anti-swing* pada miniatur *gantry crane* menggunakan MATLAB dan LabVIEW.

Penelitian-penelitian tersebut telah membahas otomatisasi, kendali berbasis aplikasi, pengendalian posisi, dan peredaman ayunan, tetapi masih diterapkan secara terpisah. Sistem otomatis berbasis Bluetooth belum mengintegrasikan monitoring IoT dan pengendalian ayunan. Sistem berbasis Android lebih berfokus pada antarmuka kendali, sedangkan penelitian *anti-swing* belum menggabungkan kendali posisi otomatis berbasis koordinat dengan komunikasi MQTT. Penelitian ini mengisi kesenjangan tersebut melalui integrasi kendali posisi berbasis encoder, pengendali PID untuk stabilisasi beban, protokol MQTT dengan broker Eclipse Mosquitto, serta aplikasi Android untuk monitoring dan kendali supervisi. Sistem dirancang agar dapat menerima koordinat *pickup* dan *drop*, menjalankan pemindahan beban secara otomatis, dan menampilkan kondisi crane melalui perangkat bergerak.

Rumusan Masalah

1. Bagaimana merancang sistem kendali posisi berbasis PID yang mampu memindahkan beban secara stabil sekaligus meredam ayunan?
2. Bagaimana mengimplementasikan MQTT untuk kendali supervisi dan monitoring prototype overhead crane?

Tujuan Penelitian

Penelitian ini bertujuan merancang sistem kendali posisi dan peredam ayunan berbasis PID serta membangun aplikasi monitoring dan kendali supervisi berbasis Android menggunakan MIT App Inventor.

Batasan Masalah

Prototype berdimensi horizontal 70 cm, vertikal 100 cm, dan tinggi 70 cm. Beban uji bermassa 47 gram dengan ukuran 15 × 5 × 7 cm serta bagian atap berukuran 25 × 9 cm. Sistem beroperasi otomatis menuju koordinat tertentu, menggunakan tuning PID manual, serta menganalisis akurasi posisi dan kestabilan ayunan. Keamanan jaringan tidak dibahas secara mendalam.

Sistematika Penulisan

Bab I memuat pendahuluan; Bab II tinjauan pustaka; Bab III metodologi penelitian; Bab IV pengujian dan analisis; serta Bab V kesimpulan dan saran.

2. TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Sejumlah penelitian mengenai sistem kendali *crane*, *anti-swing*, dan *Internet of Things* (IoT) digunakan sebagai dasar pengembangan penelitian ini. Ringkasan penelitian terdahulu disajikan pada Tabel 1.

Tabel 1. Penelitian Terkait

No.	Peneliti dan Tahun	Judul dan Isi Penelitian
1	Panji Bintang Pratama dan Intan Sianturi (2025)	*Rancang Bangun Overhead Crane Automatic Berbasis Arduino*. Penelitian ini mengembangkan *crane* otomatis yang digerakkan motor *stepper* dan dioperasikan melalui *remote* Bluetooth.
2	Rivo Restu Ananda Putra, Sri Mulyanto Herlambang, dan Henna Nurdiansari (2025)	*Rancang Bangun Pengendali Hoist Crane Berbasis Jaringan Menggunakan Aplikasi Android*. Sistem peraga *crane* dikembangkan dengan antarmuka Android melalui MIT App Inventor.
3	Andi Muhammad Farid Parenrengi (2024)	*Analisis dan Implementasi Sistem Kontrol Posisi, Anti-Sway, dan Anti-Collision pada Rubber Tyred Gantry Crane*. Penelitian menerapkan kendali posisi berbasis *encoder* dan *joystick*, *anti-collision* menggunakan sensor LIDAR, serta *anti-swing* menggunakan sensor IMU.
4	Thahirah Luthfiani Rusli dan Isdayani (2024)	*Sistem Kontrol Posisi Menggunakan Feedback Control pada Overhead Crane*. Penelitian mengembangkan kendali posisi pada *crane* bermotor AC dengan pengendali PID sebagai peredam ayunan.
5	Muhammad Yusuf Arrifai, Endryansyah, Puput Wanarti Rusimamto, dan Muhammad Syariffudin Zuhrie (2021)	*Desain Anti-Swing pada Miniature Gantry Crane Menggunakan Pengendali PID dan PD Berbasis MATLAB*. Penelitian membandingkan kinerja sistem dengan dan tanpa pengendali. MATLAB digunakan untuk *tuning* PID dan PD, sedangkan LabVIEW digunakan untuk mengidentifikasi fungsi alih sistem.

Sumber: Data peneliti pribadi

Perbedaan penelitian ini terletak pada integrasi kendali posisi otomatis berbasis koordinat, stabilisasi beban menggunakan pengendali PID, serta IoT melalui protokol MQTT dan *broker* Eclipse Mosquitto. Sistem dirancang agar *crane* dapat mengambil beban dari lokasi awal, memindahkannya ke lokasi tujuan dengan ayunan minimal, serta dipantau dan dikendalikan secara supervisi melalui aplikasi *mobile*.

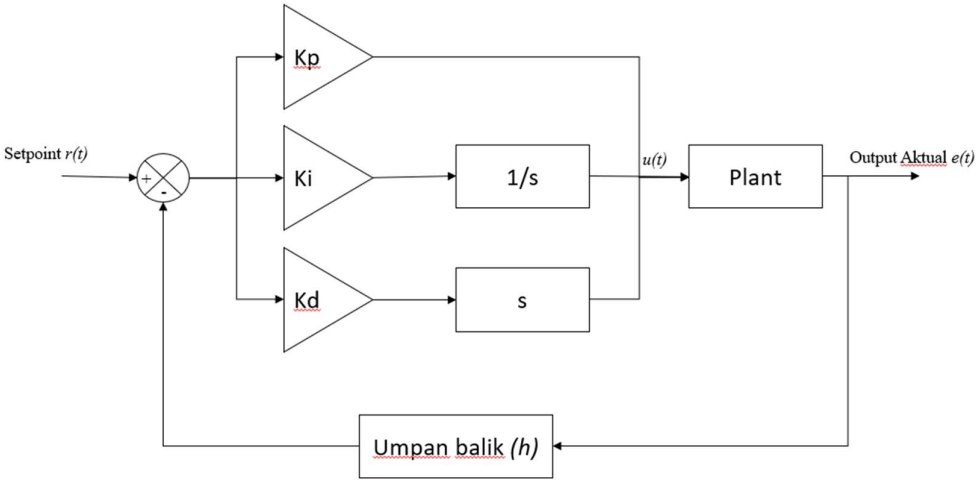
2.2 Landasan Teori

a) Overhead Crane

Overhead crane merupakan sistem pemindah material yang digunakan pada industri manufaktur, galangan kapal, dan pembangkit listrik. Mekanismenya terdiri atas *bridge* yang bergerak pada rel tetap dan *trolley* yang bergerak sepanjang *bridge*. *Trolley* menjadi titik suspensi beban yang terhubung dengan mekanisme *hoisting* menggunakan tali atau kabel baja. Pengangkatan dan penurunan beban digerakkan motor melalui sistem katrol. Kombinasi gerakan *bridge*, *trolley*, dan beban memungkinkan perpindahan dalam bidang

dua dimensi, tetapi dapat menimbulkan ayunan beban pada ruang tiga dimensi. Skematik mekanisme tersebut ditampilkan pada Gambar 1.

Gambar 1. Skema Sistem Kendali PID



Sumber: Data peneliti pribadi

Sistem *hoisting* dapat menggunakan satu atau beberapa tali dan dilengkapi *hook*. Peralatan *below-the-hook* dipilih sesuai dengan karakteristik beban. Salah satunya adalah *grab*, *yang* dibedakan menjadi *grab* mandiri, *grab* dengan sumber tenaga eksternal, dan *grab* permanen. *Grab* mandiri dipasang langsung pada *hook* tanpa tenaga tambahan. *Grab* eksternal menggunakan penggerak listrik atau hidrolik, sedangkan *grab* permanen menjadi bagian tetap *crane* dan biasanya dilepas hanya untuk perawatan. Berdasarkan fungsinya, *grab* bekerja dengan menjepit, mengait, atau melilit beban.

b) Pengendalian PID

Pengendali *Proportional-Integral-Derivative* (PID) merupakan metode kendali umpan balik yang membandingkan keluaran aktual dengan *setpoint*. Selisih keduanya menjadi *error* yang digunakan untuk menghasilkan sinyal kendali. Menurut Ogata, ketiga aksi pengendalian *tersebut* bekerja bersama untuk membentuk respons yang stabil, cepat, dan akurat. Skema kendali ditampilkan pada Gambar 1, sedangkan persamaan umumnya adalah:

$$u(t) = K_p e(t) + K_i \int_0^t e\,(t)\,dt + K_d \frac{de(t)}{dt} \qquad (1)$$

dengan $u(t)$sebagai sinyal kendali, $e(t)$sebagai *error*, serta K_p, K_i, dan $K_d$sebagai konstanta proporsional, integral, dan derivatif.

1. Aksi Proporsional (P)

Aksi proporsional dinyatakan sebagai:

$$u_P(t) = K_p e(t) \qquad (2)$$

Nilai $K_p$yang kecil menghasilkan respons lambat, sedangkan nilai yang terlalu besar dapat memicu osilasi dan ketidakstabilan. Aksi ini mengurangi *rise time*, tetapi tidak sepenuhnya menghilangkan *steady-state error* [6].

2. **Aksi Integral (I)**
   Aksi integral dinyatakan sebagai:

$$u_I(t) = K_i \int_0^t e\,(t)\,dt \qquad (3)$$

Komponen integral mengakumulasi *error* sampai penyimpangan pada kondisi tunak hilang. Namun, nilai $K_i$yang terlalu besar dapat meningkatkan *overshoot* dan memperpanjang *settling time*.

3. **Aksi Derivatif (D)**
   Aksi derivatif dinyatakan sebagai:

$$u_D(t) = K_d \frac{de(t)}{dt} \qquad (4)$$

Komponen ini merespons laju perubahan *error*. Ketika *error* berubah cepat, aksi derivatif menahan perubahan sebelum penyimpangan membesar sehingga *overshoot* menurun dan stabilitas transien meningkat. Komponen derivatif tidak menghilangkan *steady-state error*, tetapi efektif digunakan pada sistem yang rentan terhadap osilasi.

4. **Pengaruh Parameter Pengendali PID terhadap Respons Sistem**
   Ogata merangkum pengaruh setiap parameter PID terhadap karakteristik respons sistem sebagaimana disajikan pada Tabel 2.

Tabel 2. Pengaruh Parameter Pengendali PID

Parameter	*Rise Time*	*Overshoot*	*Settling Time*	*Steady-State Error*
K_p	Menurun	Meningkat	Perubahan kecil	Menurun
K_i	Menurun	Meningkat	Meningkat	Hilang
K_d	Perubahan kecil	Menurun	Menurun	Perubahan kecil

Sumber: Data pribadi peneliti

Tidak ada satu parameter yang dapat memperbaiki seluruh karakteristik respons sekaligus. Karena itu, K_p, K_i, dan $K_d$perlu disetel secara seimbang sesuai dengan kebutuhan sistem. Tabel 2.2 digunakan sebagai acuan dalam proses *tuning* kendali posisi *crane* yang dibahas pada Subbab 4.4.

c) *Internet of Things*

Internet of Things merupakan infrastruktur jaringan yang menghubungkan perangkat fisik maupun virtual melalui protokol komunikasi yang interoperabel. Setiap perangkat memiliki identitas unik dan dapat bertukar data secara otomatis melalui internet. Karakteristik dinamis dan *self-configuring* memungkinkan perangkat menyesuaikan operasi, membentuk jaringan, serta melakukan konfigurasi dengan intervensi pengguna yang terbatas.

Identitas seperti alamat IP atau URL mendukung proses identifikasi, pemantauan, dan pengendalian perangkat dari jarak jauh. Data yang dihimpun dari berbagai perangkat selanjutnya dapat dianalisis dan digunakan untuk mendukung pengambilan keputusan sistem.

d) *Message Queuing Telemetry Transport*

MQTT (*Message Queuing Telemetry Transport*) merupakan protokol komunikasi ringan berbasis *publish/subscribe*. Protokol ini bersifat terbuka, mudah diterapkan, dan mendukung komunikasi banyak perangkat melalui satu server. MQTT sesuai digunakan pada

jaringan dengan *bandwidth* terbatas, latensi tinggi, serta perangkat yang memiliki kapasitas memori dan pemrosesan rendah.

Keunggulan MQTT mencakup ukuran *header* pesan yang kecil, komunikasi *machine-to-machine*, dukungan beberapa tingkat *Quality of Service* (QoS), serta mekanisme penyimpanan atau pengiriman ulang pesan ketika koneksi antara *client* dan server terganggu.

2.3 Komponen yang Digunakan

a) Arduino Mega 2560

Arduino Mega 2560 merupakan papan mikrokontroler berbasis ATmega2560 yang dipilih karena menyediakan banyak jalur *input/output* [9]. Papan ini memiliki 54 pin digital, 15 pin PWM, 16 input analog, empat UART, kristal 16 MHz, SRAM 8 kB, *flash memory* 256 kB termasuk *bootloader* 8 kB, dan EEPROM 4 kB. Tegangan operasinya 5 V dengan masukan yang disarankan 7–12 V, keluaran 5 V dan 3,3 V, serta arus 20 mA per pin. Pemrograman dilakukan melalui USB Type-B atau ICSP dengan antarmuka UART, I2C, dan SPI.

b) ESP32

ESP32 merupakan mikrokontroler berdaya rendah yang dikembangkan oleh Espressif Systems dengan dukungan komunikasi Wi-Fi dan Bluetooth terintegrasi untuk pengembangan sistem *embedded* dan *Internet of Things*. Mikrokontroler ini menggunakan prosesor Xtensa *dual-core* 32-bit LX6 dengan frekuensi kerja hingga 240 MHz. ESP32 juga menyediakan sejumlah pin GPIO serta antarmuka komunikasi SPI, I2C, dan UART. Dalam penelitian ini, ESP32 digunakan sebagai penghubung antara Arduino Mega 2560 dan *broker* MQTT melalui jaringan Wi-Fi

c) Arduino Pro Micro

Arduino Pro Micro merupakan papan mikrokontroler berbasis ATmega32U4 yang digunakan sebagai unit pemrosesan data sensor MPU6050. Modul ini dipilih karena memiliki antarmuka USB terintegrasi dan mendukung komunikasi I2C serta UART. Dalam sistem yang dikembangkan, MPU6050 terhubung dengan Arduino Pro Micro melalui pin SDA dan SCL, sedangkan hasil pembacaan sudut dikirimkan ke Arduino Mega 2560 melalui komunikasi serial. Arduino Pro Micro mengolah data *accelerometer* dan *gyroscope* menggunakan *complementary filter* sebelum data digunakan sebagai masukan kendali *anti-swing*

d) Motor DC Gearbox

Motor DC mengubah energi listrik searah menjadi energi mekanik melalui interaksi medan magnet pada *armature* dan stator. Motor yang digunakan memiliki *gearbox* 1:48, tegangan 3–6 V, arus tanpa beban 150 mA, kecepatan 90 RPM pada 3 V dan 200 RPM pada 6 V, torsi 0,15–0,60 Nm, serta *stall torque* 0,8 kg·cm. Spesifikasi tersebut disajikan pada Tabel 2.6.

e) Motor DC JGA25-370

Motor JGA25-370 digunakan sebagai aktuator pemindah beban dan telah dilengkapi *gearbox* serta *encoder*. Spesifikasinya meliputi tegangan 12 V, kecepatan nominal 130 RPM, arus kerja 60 mA, poros berbentuk D, rasio 1:47, dan torsi 1,3 kg·cm. *Encoder* bawaannya bekerja pada 3,3–5 V DC, menggunakan sensor *Hall effect* dua fasa atau *quadrature*, beresolusi 11 PPR, dan memiliki *channel* A serta B. Data *encoder* digunakan sebagai umpan balik *closed-loop* untuk menghitung posisi dan kecepatan motor.

f) Motor Servo SG90

Motor servo SG90 merupakan aktuator putar *closed-loop* yang terdiri atas motor DC, *gear*, rangkaian kendali, dan potensiometer. *Gear* menurunkan kecepatan serta meningkatkan torsi, sedangkan potensiometer membaca posisi sudut. Servo memiliki torsi 2,5 kg·cm, tegangan 4,8–6 V, berat 14,7 gram, kecepatan respons 0,1 detik/60°, dan rentang suhu kerja 0–55°C.

g) Driver L298N

L298N adalah *dual full-bridge driver* untuk mengendalikan motor DC, motor *stepper*, *relay*, atau solenoid. Modul ini memungkinkan mikrokontroler mengatur arah melalui H-Bridge dan kecepatan melalui PWM. Spesifikasinya meliputi tegangan operasi 5–46 V, tegangan logika 5 V, arus maksimum 2 A per *channel*, dua *channel*, antarmuka TTL, dan proteksi termal.

h) Driver HX711

HX711 merupakan ADC 24-bit berpresisi tinggi untuk *load cell* dan dilengkapi *Programmable Gain Amplifier* berderau rendah. *Channel* A menyediakan *gain* 128 atau 64, sedangkan *Channel* B menggunakan *gain* 32. Modul bekerja pada 2,6–5,5 V, mengonsumsi arus kurang dari 1,5 mA, memiliki frekuensi *sampling* 10 atau 80 SPS, dan menggunakan antarmuka serial digital.

i) Limit Switch

Limit switch adalah sensor mekanis yang mengubah kondisi kontak ketika tuas mencapai batas tertentu. Sensor ini digunakan untuk mendeteksi posisi objek bergerak dan memiliki kontak *Normally Open* serta *Normally Closed*. Spesifikasinya mencakup tegangan maksimum 3,3–125 V, arus maksimum 5 A, penggerak tuas engsel, dan konfigurasi kontak SPDT.

j) MPU-6050

MPU-6050 adalah sensor MEMS yang mengintegrasikan *accelerometer* dan *gyroscope* tiga sumbu. Sensor ini mendeteksi percepatan, orientasi, kemiringan, dan kecepatan sudut melalui komunikasi I2C pada pin SDA dan SCL. Spesifikasinya meliputi tegangan 3–5 V, rentang *accelerometer* ±2g, ±4g, ±8g, dan ±16g, rentang *gyroscope* ±250, ±500, ±1.000, dan ±2.000°/s, ADC 16-bit, serta *Digital Motion Processor*.

k) Load Cell

Load cell strain gauge mengukur gaya berdasarkan perubahan resistansi akibat deformasi elemen logam. Karena sinyal keluarannya kecil, sensor memerlukan penguat dan pengondisi sinyal. Konfigurasi *Wheatstone bridge* digunakan untuk meningkatkan sensitivitas dan mengurangi pengaruh temperatur. *Load cell* memiliki kapasitas maksimum 5 kg, ketepatan 0,05%, dan *rated output* 1,0 ± 0,15 mV/V.

l) Power Supply dan LM2596

Power supply atau catu daya merupakan perangkat yang mengubah arus listrik AC menjadi DC untuk memasok kebutuhan daya pada rangkaian elektronik. Catu daya yang digunakan dalam penelitian ini menghasilkan tegangan 12 V DC dengan rentang keluaran 11–13,8 V dan arus maksimum 10 A. Perangkat ini dilengkapi perlindungan terhadap beban berlebih, hubungan singkat, dan tegangan berlebih. Tegangan keluaran selanjutnya dapat diturunkan menggunakan modul LM2596 agar sesuai dengan kebutuhan tegangan komponen pada sistem

m) Arduino IDE

Arduino IDE digunakan untuk menulis dan mengunggah *sketch* ke papan Arduino. Program diterjemahkan ke bahasa C dan dikompilasi menggunakan *avr-gcc*. Struktur void setup() dan void loop() menyederhanakan proses pemrograman mikrokontroler.

n) 2.3.14 MIT App Inventor

MIT App Inventor merupakan platform pengembangan aplikasi Android berbasis blok visual. Pengguna dapat merancang antarmuka dan fungsi aplikasi melalui metode *drag-and-drop* sehingga proses pengembangan menjadi lebih sederhana.

o) Eclipse Mosquitto

Eclipse Mosquitto merupakan *broker* MQTT *open-source* yang menjadi perantara komunikasi antara *publisher* dan *subscribe*. *Broker* ini mendukung MQTT versi 5.0, 3.1.1, dan 3.1 serta dapat dijalankan pada perangkat berspesifikasi rendah maupun server berskala besar.

METODOLOGI PENELITIAN

3.1 Waktu dan Tempat Penelitian

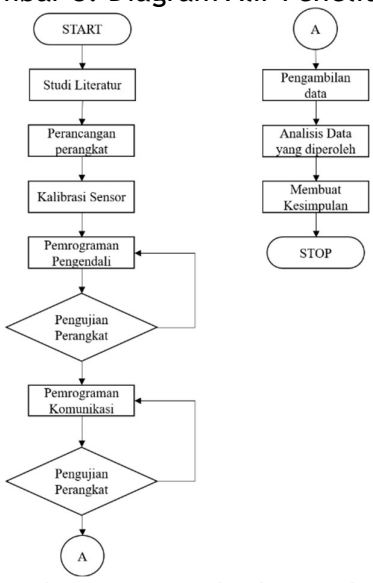
Penelitian dilaksanakan di Jalan 28 Oktober, Gang Sido Mukti, Pontianak, Kalimantan Barat, selama enam bulan, yaitu Februari–Juli 2026. Kegiatan penelitian mencakup studi literatur, perancangan sistem, pembuatan *prototype*, pemrograman, pengujian, pengambilan data, dan analisis hasil. Lokasi pembuatan serta pengujian sistem ditunjukkan pada Gambar 2.

Gambar 2. Lokasi Pembuatan dan Pengujian Sistem

Sumber: https://www.google.com/maps

3.2 Metode Penelitian

Gambar 3. Diagram Alir Penelitian



Sumber: Data pribadi peneliti

Penelitian dilakukan melalui tahapan yang ditampilkan pada Gambar 3. Proses dimulai dengan studi literatur mengenai sistem *crane*, sensor, motor penggerak, mikrokontroler, IoT, serta metode kendali. Tahap berikutnya meliputi perancangan mekanik, rangkaian elektronika, pemilihan komponen, dan penentuan sistem transmisi.

Sensor kemudian dikalibrasi untuk memastikan pembacaan posisi, sudut, dan berat muatan sesuai dengan kebutuhan sistem. Arduino Mega 2560 diprogram sebagai pengendali utama untuk mengatur motor, membaca sensor, dan menjalankan algoritma kendali. Arduino Pro Micro digunakan untuk membaca MPU6050, memproses data sudut menggunakan *complementary filter*, lalu mengirimkannya ke Arduino Mega 2560.

Setelah perangkat keras dan program kendali diuji, pemrograman komunikasi dilakukan menggunakan ESP32 yang terhubung dengan *broker* Eclipse Mosquitto. Aplikasi Android yang dibuat melalui MIT App Inventor digunakan sebagai antar muka monitoring dan kendali supervisi. Sistem yang telah terintegrasi kemudian diuji dan diambil datanya untuk menganalisis akurasi sensor, kestabilan beban, respons kendali, waktu perpindahan, dan keberhasilan pemindahan beban.

Penelitian ini menggunakan metode rancang bangun yang disertai pengujian eksperimental. Metode rancang bangun digunakan untuk menghasilkan *prototype overhead crane* yang mengintegrasikan kendali posisi, peredam ayunan, dan komunikasi IoT. Pengujian eksperimental dilakukan untuk mengevaluasi fungsi sensor dan komponen, akurasi posisi, sudut ayunan beban, waktu perpindahan, serta kinerja sistem pada mode supervisi dan mode otomatis. Tahapan penelitian meliputi studi literatur, perancangan perangkat, kalibrasi sensor, pemrograman pengendali, pemrograman komunikasi, pengujian sistem, pengambilan data, dan analisis hasil

3.3 Perancangan Sistem

Sistem dirancang dengan mengintegrasikan sensor, aktuator, mikrokontroler, dan komunikasi IoT. Arduino Mega 2560 menjadi pengendali utama, sedangkan Arduino Pro Micro memproses data MPU6050 dan ESP32 menangani komunikasi MQTT. Antarmuka pengguna dikembangkan menggunakan MIT App Inventor.

MPU6050 membaca kemiringan dan ayunan beban. Data sensor diproses pada Arduino Pro Micro, kemudian dikirim melalui UART untuk mengurangi potensi gangguan akibat penggunaan kabel I2C yang panjang. *Hall-effect encoder* membaca posisi motor, *load cell* mengukur berat beban, sedangkan enam *limit switch* digunakan sebagai referensi posisi, pemicu pengambilan dan peletakan beban, serta pembatas gerak sumbu Z.

Arduino Mega mengirimkan sinyal kendali ke tiga *driver* L298N. Motor sumbu X mengatur gerak kanan–kiri, sumbu Y mengatur gerak maju–mundur, dan sumbu Z mengangkat atau menurunkan beban. Dua motor servo digunakan untuk menggerakkan *gripper*. Data sistem diteruskan oleh ESP32 ke *broker* Mosquitto agar dapat dipantau melalui smartphone.

a) Sistem Kendali Posisi dan Anti-Swing

Sistem menggunakan dua lapis kendali yang dijalankan setiap 20 ms. Kendali PID posisi mengarahkan *crane* menuju koordinat target, sedangkan kendali derivatif (*D only*) meredam ayunan beban. Kedua keluarannya dijumlahkan sebelum dikirim sebagai sinyal PWM ke L298N.

Koordinat dari aplikasi Android dikirim melalui MQTT ke ESP32 dan diteruskan ke Arduino Mega. Posisi aktual dibaca dari *encoder* sumbu X dan Y melalui *Interrupt Service Routine*. Selisih posisi target dan aktual menjadi *error* bagi kendali PID.

Parameter kendali posisi adalah sebagai berikut:

Parameter	Sumbu X	Sumbu Y
K_p	0,200	0,500
K_i	0,0008	0,0009
K_d	0,0020	0,0150
Toleransi posisi	±5 pulsa	±5 pulsa
Batas integral	±500	±500
Keluaran maksimum	165 PWM	105 PWM

Mekanisme *anti-windup* membatasi akumulasi integral pada rentang ±500. Sistem dinyatakan mencapai target ketika galat kedua sumbu berada dalam toleransi ±5 pulsa. Motor kemudian dihentikan dan sistem menunggu hingga sudut ayunan berada di bawah 3°.

Kendali *anti-swing* menggunakan data MPU6050 yang diproses dengan bobot 90% giroskop dan 10% akselerometer. Komponen proporsional dan integral tidak digunakan karena ayunan bersifat osilatoris dan bukan galat posisi. Parameter $K_d$ditetapkan sebesar 0,8 pada sumbu X dan 1,0 pada sumbu Y. Keluaran dibatasi masing-masing ±40 PWM dan ±25 PWM, dengan *deadband* 0,8° serta *low-pass filter* berfaktor alfa 0,3.

Sistem dinyatakan telah mencapai target ketika galat posisi sumbu X dan Y berada dalam toleransi ±5 pulsa secara bersamaan. Setelah posisi tercapai, motor dihentikan dan sistem menunggu hingga sudut ayunan beban berada di bawah 3°. Nilai tersebut digunakan sebagai ambang operasional dalam program untuk memastikan ayunan telah cukup mereda sebelum sistem melanjutkan ke tahapan berikutnya.

b) Sistem IoT

Arsitektur IoT terdiri atas Arduino Pro Micro, Arduino Mega 2560, ESP32, *broker* Mosquitto, dan smartphone. Arduino Mega mengirimkan data sistem ke ESP32 melalui UART. ESP32 meneruskan data tersebut menggunakan MQTT melalui jaringan Wi-Fi. Mosquitto berfungsi menerima dan mendistribusikan pesan antara perangkat dan aplikasi.

Sistem menggunakan empat topik *subscribe*, yaitu crane/command/task, crane/command/PID, crane/command/PID_toggle, dan crane/command/home, serta satu topik *publish*, yaitu crane/status. Pesan perintah menggunakan QoS 1, sedangkan status menggunakan QoS 0 karena diperbarui setiap 200 ms.

c) Perancangan Perangkat Keras

Tiga L298N mengendalikan motor pada sumbu X, Y, dan Z. Pin kendali motor sumbu X menggunakan pin 22–25 dengan PWM pin 5 dan 6. Sumbu Y menggunakan pin 26–29 dengan PWM pin 10 dan 11, sedangkan sumbu Z menggunakan pin 44–45 dan PWM pin 7.

Encoder sumbu X terhubung ke pin 2 dan 3, sedangkan sumbu Y menggunakan pin 18 dan 19. Servo *gripper* terhubung ke pin 46 dan 47, dengan posisi buka 130° dan tutup 85°. Modul HX711 menggunakan pin 30 sebagai data dan pin 31 sebagai *clock*.

Enam *limit switch* dipasang pada pin 34–39 dengan konfigurasi INPUT_PULLUP. Fungsinya mencakup referensi *home* sumbu X, Y, dan Z, pembatas bawah sumbu Z, serta pemicu *pickup* dan *drop*. Arduino Pro Micro berkomunikasi melalui Serial2, sedangkan ESP32 menggunakan Serial3. Keduanya bekerja pada 9.600 bps.

d) Perancangan Perangkat Lunak

Program Arduino Mega dijalankan secara *non-blocking* dengan pendekatan *state machine* sebanyak 28 *state*. Tahapan utamanya terdiri atas *homing*, *pickup*, *drop*, dan *return*. Posisi *encoder* dikonversi menggunakan faktor 0,0085 cm/pulsa pada sumbu X dan 0,0098 cm/pulsa pada sumbu Y.

Data status dikirim setiap 200 ms dalam format CSV yang memuat posisi X, posisi Y, status *gripper*, sudut X, sudut Y, berat, mode operasi, dan status PID. ESP32 menerima data tersebut, menampilkannya melalui Serial Monitor, dan memublikasikannya ke topik crane/status.

ESP32 menggunakan koneksi Wi-Fi dengan *timeout* 15 detik, percobaan koneksi ulang Wi-Fi setiap 10 detik, dan koneksi ulang MQTT setiap 5 detik. Perintah juga dapat dikirim secara manual melalui Serial Monitor untuk keperluan pengujian.

Arduino Pro Micro menggunakan ATmega32U4 untuk membaca MPU6050 melalui I2C. Sensor diatur pada rentang akselerometer ±8 G, giroskop ±500°/s, dan *bandwidth* 21 Hz. Kalibrasi menggunakan 100 sampel dengan jeda 10 ms. Data sudut dikirim sekitar 50 Hz melalui format X:nilai,Y:nilai.

Aplikasi Android menggunakan ekstensi UrsAI2MQTT. Fitur utamanya mencakup koneksi *broker*, monitoring posisi dan sudut, status *gripper*, berat beban, mode operasi, status PID, input koordinat, perintah *execute* dan *home*, empat koordinat *preset*, serta pengaturan parameter PID. Alamat bawaan *broker* adalah 192.168.100.9 dengan port 1883.

e) Desain Prototype

Rangka *prototype* memiliki panjang 1.000 mm, lebar 700 mm, dan tinggi 700 mm. Titik pengambilan serta peletakan berada sekitar 200,18 mm dan 157,99 mm dari sisi rangka. *Gripper* memiliki tinggi 138,49 mm dan lebar maksimum sekitar 156 mm.

3.4 Variabel dan Analisis Data

Variabel penelitian meliputi akurasi posisi dalam sentimeter, sudut maksimum ayunan dalam derajat, dan waktu perpindahan dalam detik. Analisis dilakukan menggunakan galat, persentase galat, dan *Mean Absolute Error* (MAE). Nilai galat yang semakin kecil menunjukkan bahwa posisi aktual semakin mendekati nilai referensi.

Perangkat lunak mengatur proses inisialisasi, pembacaan sensor, pemilihan mode operasi, serta pemindahan beban. Sistem menyediakan dua mode operasi, yaitu mode supervisi dan mode otomatis. Pada mode supervisi, operator menentukan koordinat *pickup* dan *drop* melalui aplikasi Android. Pada mode otomatis, proses pemindahan dipicu oleh *limit switch* dan menggunakan koordinat yang telah ditetapkan dalam program.

HASIL DAN PEMBAHASAN

4.1 Hasil Perancangan Perangkat Keras

Perangkat keras *prototype overhead crane* terdiri atas rangka utama, *trolley*, *gripper*, beban uji, dan panel kendali. Rangka dibuat dari profil besi siku berbentuk persegi panjang sebagai lintasan gerak horizontal dan vertikal. Panjang lintasan sumbu X adalah 36 cm, sedangkan lintasan sumbu Y mencapai 87 cm.

Gripper menggunakan rangka kayu dan digerakkan motor servo untuk menjepit beban saat proses *pickup*. Beban uji berupa kotak kayu dengan panjang bagian atas 25,5 cm, lebar 9 cm, dan tinggi 7 cm. *Trolley* bergerak pada rel besi profil siku menggunakan roda karet. Dua motor DC JGA25-370 digunakan sebagai penggerak sumbu X dan sumbu Z, sedangkan motor DC *gearbox* menggerakkan sumbu Y. Arduino Pro Micro, *load cell*, dan HX711 dipasang pada *trolley* agar terintegrasi dengan mekanisme pengangkatan.

Panel kendali memuat *power supply*, tiga modul L298N, serta terminal blok distribusi kabel. Penandaan kabel berdasarkan fungsi digunakan untuk memudahkan pemeriksaan dan perawatan rangkaian.

4.2 Hasil Perancangan Perangkat Lunak

Sistem bekerja melalui integrasi aplikasi Android, *broker* Eclipse Mosquitto, ESP32, Arduino Mega 2560, dan Arduino Pro Micro. Operator menghubungkan aplikasi ke *broker* MQTT, memasukkan koordinat *pickup* dan *drop*, lalu menekan tombol EXECUTE.

Perintah diteruskan melalui MQTT menuju ESP32, kemudian dikirim melalui komunikasi serial ke Arduino Mega. Arduino Mega menjalankan kendali PID berdasarkan umpan balik *encoder*. Pada saat yang sama, Arduino Pro Micro membaca sudut beban melalui MPU6050 dan mengirimkan data tersebut untuk mendukung kendali *anti-swing*.

Setelah mencapai titik *pickup*, *gripper* menjepit beban. Crane kemudian bergerak menuju koordinat *drop*, melepaskan beban, dan kembali ke posisi *home*. Posisi crane, sudut beban, status *gripper*, dan berat beban dikirim secara berkala ke aplikasi Android.

4.3 Pengujian Sensor dan Komponen

Pengujian dilakukan terhadap *encoder*, *limit switch*, MPU6050, *gripper*, dan *load cell*. Kalibrasi *encoder* dilakukan lima kali pada setiap sumbu dengan membandingkan jumlah pulsa dan jarak fisik yang diukur menggunakan penggaris.

Nilai konversi yang diperoleh adalah 0,0085 cm/pulsa untuk sumbu X dan 0,0098 cm/pulsa untuk sumbu Y. Nilai tersebut digunakan pada seluruh pengujian posisi berikutnya. Perbedaan antara target dan hasil aktual dipengaruhi oleh slip mekanis, toleransi komponen, serta ketelitian pengukuran manual.

Pengujian *encoder* sumbu X menggunakan lima target, yaitu 500–2.500 pulsa, dengan total 25 data. MAE yang diperoleh sebesar 0,61 cm dan rata-rata persentase galat 6,01%, sehingga akurasi rata-rata mencapai 93,99%. Galat terbesar sebesar 17,65% terjadi pada target 1.000 pulsa. Kondisi tersebut dikaitkan dengan *undershoot*, pengaruh *dead zone*, dan slip mekanis.

Pengujian sumbu Y menggunakan enam target, yaitu 1.000–6.000 pulsa, dengan total 30 data. MAE yang diperoleh sebesar 0,40 cm dan rata-rata galat 1,24%, sehingga akurasi rata-rata mencapai 98,76%. Galat terbesar sebesar 3,06% terjadi pada target 5.000 pulsa. Hasil ini menunjukkan bahwa pembacaan posisi pada sumbu Y lebih konsisten daripada sumbu X.

Enam *limit switch* berhasil terbaca ketika ditekan. Pengujian *homing* sebanyak sepuluh kali juga menunjukkan bahwa motor selalu berhenti ketika *limit switch Home X*, *Home Y*, dan Z atas aktif. Tingkat keberhasilan pengujian mencapai 100%.

MPU6050 diuji pada sudut 0°, 5°, 10°, 20°, 25°, dan 30°, masing-masing sebanyak lima kali. Aplikasi pengukur sudut pada smartphone digunakan sebagai acuan. Pada sumbu X diperoleh MAE 1,45° dan rata-rata galat 6,78%, sehingga akurasi rata-rata mencapai 93,22%. Pada sumbu Y, MAE mencapai 1,92° dengan rata-rata galat 9,93% dan akurasi 90,07%. Penyimpangan dikaitkan dengan keterbatasan alat acuan, *drift* sensor, orientasi pemasangan, getaran mekanis, dan pengaturan *complementary filter*.

Tabel 3. Hasil Pengujian *Gripper*

No	*Gripper* Menutup?	Beban Jatuh?	Hasil
1	MENUTUP	TIDAK	BERHASIL
2	MENUTUP	TIDAK	BERHASIL
3	MENUTUP	TIDAK	BERHASIL
4	MENUTUP	TIDAK	BERHASIL
5	MENUTUP	TIDAK	BERHASIL
6	MENUTUP	TIDAK	BERHASIL
7	MENUTUP	TIDAK	BERHASIL
8	MENUTUP	TIDAK	BERHASIL
9	MENUTUP	TIDAK	BERHASIL
10	MENUTUP	TIDAK	BERHASIL

Berdasarkan hasil pengujian pada Tabel 3, *gripper* berhasil menutup dan menjepit beban pada seluruh percobaan tanpa menyebabkan beban terjatuh. Tingkat keberhasilan pengujian mencapai 100% dari sepuluh percobaan. Hasil tersebut menunjukkan bahwa mekanisme *gripper* yang digerakkan oleh dua motor servo SG90 dengan sudut buka 130° dan sudut tutup 85° dapat bekerja secara konsisten selama proses pengambilan, pemindahan, dan peletakan beban.

Load cell diuji sebanyak 30 kali dalam kondisi diam dengan target awal 0 gram dan beban 47 gram. Pembacaan awal rata-rata sebesar −2,48 gram dengan MAE 2,54 gram, yang menunjukkan adanya *drift* dari titik nol. Pada kondisi beban, rata-rata persentase galat sebesar 5,77% dengan akurasi 94,23%. Galat terbesar mencapai 23,83% pada pembacaan 58,2 gram. Pembacaan menjadi lebih tidak stabil ketika crane bergerak karena dipengaruhi getaran dan ayunan. Karena itu, fungsi monitoring berat masih memerlukan pengembangan untuk digunakan pada kondisi dinamis.

4.4 Penentuan Parameter Kendali

Parameter PID ditentukan melalui *tuning* manual berdasarkan pengaruh K_p, K_i, dan $K_d$terhadap respons sistem sebagaimana dijelaskan oleh Ogata. Penyetelan dilakukan secara terpisah pada sumbu X dan Y.

Proses dimulai dengan mengaktifkan komponen proporsional. Nilai $K_p$dinaikkan sampai crane mampu mendekati target dengan respons yang memadai. Komponen $K_d$kemudian ditambahkan untuk mengurangi *overshoot*, sedangkan $K_i$digunakan dalam nilai kecil untuk mengurangi *steady-state error* tanpa meningkatkan osilasi.

Parameter akhir sumbu X adalah $K_p = 0{,}200$, $K_i = 0{,}0008$, dan $K_d = 0{,}0020$. Pada sumbu Y digunakan $K_p = 0{,}500$, $K_i = 0{,}0009$, dan $K_d = 0{,}0150$. Nilai pada sumbu Y lebih besar karena lintasannya lebih panjang dan memiliki karakteristik mekanis yang berbeda.

Kendali *anti-swing* hanya menggunakan komponen derivatif dengan $K_d = 0{,}8$pada sumbu X dan $K_d = 1{,}0$pada sumbu Y. Keluarannya dibatasi hingga ±40 PWM pada sumbu X dan ±25 PWM pada sumbu Y. *Deadband* sebesar 0,8° diterapkan agar sistem tidak merespons *noise* ketika beban relatif diam.

4.5 Pengujian Mode Operasi

Sistem menyediakan mode supervisi dan mode otomatis. Pada mode supervisi, operator memasukkan koordinat *pickup* dan *drop* melalui aplikasi. Setelah perintah diterima, seluruh proses pemindahan dijalankan oleh *state machine* tanpa intervensi tambahan.

Pada mode otomatis, proses dimulai ketika beban menekan *limit switch pickup*. Sistem terlebih dahulu memeriksa *limit switch drop*. Jika area *drop* masih terisi, proses ditunda sampai area tersebut kosong. Crane kemudian mengambil beban, memindahkannya ke koordinat tetap, meletakkannya, dan kembali ke posisi *home*.

4.6 Analisis Pengujian Mode Supervisi

Pengujian mode supervisi dilakukan sebanyak 15 kali menggunakan tiga variasi titik *pickup* dan satu titik *drop* pada koordinat (2500,6000), yang setara dengan 21,25 cm pada sumbu X dan 58,80 cm pada sumbu Y.

Ketika PID diaktifkan, MAE sumbu X sebesar 2,60 cm dan sumbu Y sebesar 1,94 cm. Rata-rata galat masing-masing mencapai 12,25% dan 3,30%, sedangkan waktu perpindahan rata-rata adalah 31,1 detik. Sudut maksimum rata-rata tercatat 11,34° pada sumbu X dan 10,84° pada sumbu Y.

Ketika PID dimatikan, MAE meningkat menjadi 7,17 cm pada sumbu X dan 27,86 cm pada sumbu Y. Rata-rata galat mencapai 33,74% dan 47,38%, dengan waktu perpindahan 33,85 detik. Hasil tersebut menunjukkan bahwa kendali PID meningkatkan akurasi posisi, terutama pada sumbu Y.

Tabel 4. Perbandingan Kendali *PID* dihidupkan dan Kendali *PID* dimatikan

Parameter	*PID* dihidupkan	*PID* dimatikan
MAE X (cm)	2,60	7,17
MAE Y (cm)	1,94	27,86
Rata-rata %Galat X	12,25%	33,74%
Rata-rata %Galat Y	3,30%	47,38%
Rata-rata Sudut Max X (°)	11,34	12,84
Rata-rata Sudut Max Y (°)	10,84	9,19
Rata-rata Waktu (dtk)	31,1	33,85

Namun, Tabel 4 menunjukkan sudut maksimum rata-rata sumbu Y sebesar 10,84° saat PID aktif dan 9,19° saat PID tidak aktif. Dari sisi ayunan beban, pengaktifan kendali PID memberikan hasil yang berbeda pada kedua sumbu. Rata-rata sudut maksimum sumbu X

menurun dari 13,05° ketika PID dimatikan menjadi 11,34° ketika PID dihidupkan. Sebaliknya, rata-rata sudut maksimum sumbu Y meningkat dari 8,99° menjadi 10,84°. Hasil ini menunjukkan bahwa kendali *anti-swing* pada mode supervisi telah mengurangi ayunan pada sumbu X, tetapi belum memberikan penurunan yang konsisten pada sumbu Y. Penyetelan parameter kendali *anti-swing* sumbu Y masih perlu dioptimalkan agar kinerjanya lebih stabil pada berbagai koordinat *pickup*

4.7 Analisis Pengujian Mode Otomatis

Pengujian mode otomatis dilakukan sepuluh kali pada koordinat *drop* (3000,6000), yang setara dengan 25,5 cm pada sumbu X dan 58,8 cm pada sumbu Y. Ketika PID diaktifkan, MAE sumbu X sebesar 1,16 cm dan sumbu Y sebesar 3,31 cm. Rata-rata galat mencapai 4,55% dan 5,63%. Sudut maksimum rata-rata adalah 8,28° pada sumbu X dan 5,07° pada sumbu Y, dengan waktu operasi 27,3 detik.

Ketika PID dimatikan, MAE meningkat menjadi 5,54 cm pada sumbu X dan 27,23 cm pada sumbu Y. Rata-rata galat mencapai 21,73% dan 46,31%. Sudut maksimum rata-rata meningkat menjadi 11,0° dan 7,99°, sedangkan waktu operasi bertambah menjadi 33,29 detik.

Perbandingan tersebut menunjukkan bahwa penggunaan kendali PID posisi dan *anti-swing* pada mode otomatis menghasilkan posisi akhir yang lebih akurat, sudut ayunan lebih rendah, serta waktu operasi lebih singkat. Peningkatan MAE sumbu Y dari 3,31 cm menjadi 27,23 cm ketika PID dimatikan memperlihatkan pengaruh kendali terhadap kestabilan perpindahan pada lintasan yang lebih panjang.

Validasi yang lebih jelas terhadap kendali *anti-swing* terdapat pada pengujian mode otomatis. Ketika PID dihidupkan, sudut maksimum rata-rata adalah 8,28° pada sumbu X dan 5,07° pada sumbu Y. Ketika PID dimatikan, nilainya meningkat menjadi 11,0° dan 7,99°. Data tersebut menunjukkan penurunan ayunan pada kedua sumbu.

KESIMPULAN

Berdasarkan proses perancangan, implementasi, pengujian, dan analisis yang telah dilakukan, dapat disimpulkan bahwa:

1. Sistem *prototype overhead crane* berbasis IoT berhasil dirancang dengan mengintegrasikan Arduino Mega 2560 sebagai pengendali utama, Arduino Pro Micro sebagai pemroses data MPU6050, dan ESP32 sebagai penghubung komunikasi MQTT. Eclipse Mosquitto digunakan sebagai *broker*, sedangkan aplikasi Android berbasis MIT App Inventor berfungsi sebagai antarmuka monitoring dan kendali supervisi. Sistem dapat beroperasi dalam mode supervisi dan mode otomatis.
2. Sistem kendali posisi menggunakan PID pada sumbu X dan Y, sedangkan peredaman ayunan menggunakan kendali derivatif atau *D only*. Parameter akhir kendali posisi adalah $K_p = 0{,}200$, $K_i = 0{,}0008$, dan $K_d = 0{,}0020$pada sumbu X serta $K_p = 0{,}500$, $K_i = 0{,}0009$, dan $K_d = 0{,}0150$pada sumbu Y. Kendali *anti-swing* menggunakan $K_d = 0{,}8$pada sumbu X dan $K_d = 1{,}0$pada sumbu Y.
3. Hasil kalibrasi menghasilkan faktor konversi sebesar 0,0085 cm/pulsa pada sumbu X dan 0,0098 cm/pulsa pada sumbu Y. Pengujian *encoder* memperoleh MAE 0,61 cm dengan rata-rata galat 6,01% pada sumbu X serta MAE 0,40 cm dengan rata-rata galat 1,24% pada sumbu Y. Hasil tersebut menunjukkan bahwa *encoder* dapat digunakan sebagai umpan balik posisi, meskipun pembacaan pada sumbu X masih dipengaruhi oleh *dead zone*, slip mekanis, dan karakteristik penggeraknya.
4. Sensor MPU6050 memperoleh MAE 1,45° dan rata-rata galat 6,78% pada sumbu X, sedangkan pada sumbu Y diperoleh MAE 1,92° dan rata-rata galat 9,93%. Seluruh *limit switch* berhasil beroperasi pada setiap pengujian, sementara *gripper* mampu menjepit beban tanpa terjatuh dalam sepuluh percobaan. Pengujian *load cell* pada kondisi diam

menghasilkan akurasi rata-rata 94,23%, tetapi pembacaannya belum stabil ketika crane bergerak karena dipengaruhi getaran dan ayunan beban.

5. Pada mode supervisi, pengaktifan PID menurunkan MAE sumbu X dari 7,17 cm menjadi 2,60 cm dan MAE sumbu Y dari 27,86 cm menjadi 1,94 cm. Rata-rata waktu operasi juga berkurang dari 33,85 detik menjadi 31,1 detik. Kendali *anti-swing* menurunkan rata-rata sudut maksimum sumbu X dari 13,05° menjadi 11,34°, tetapi pada sumbu Y nilainya meningkat dari 8,99° menjadi 10,84°. Temuan ini menunjukkan bahwa kendali posisi bekerja lebih konsisten dibandingkan kendali *anti-swing* sumbu Y pada mode supervisi.
6. Pada mode otomatis, kinerja kendali lebih konsisten. Pengaktifan PID menghasilkan MAE 1,16 cm pada sumbu X dan 3,31 cm pada sumbu Y, sedangkan saat PID dimatikan nilainya meningkat menjadi 5,54 cm dan 27,23 cm. Rata-rata sudut maksimum juga menurun dari 11,0° menjadi 8,28° pada sumbu X serta dari 7,99° menjadi 5,07° pada sumbu Y. Waktu operasi berkurang dari 33,29 detik menjadi 27,3 detik.

Secara keseluruhan, sistem yang dikembangkan mampu melakukan pemindahan beban berdasarkan koordinat, memantau kondisi crane melalui aplikasi Android, dan meningkatkan ketepatan posisi melalui penerapan kendali PID. Kendali *anti-swing* menunjukkan hasil yang baik pada mode otomatis, tetapi masih memerlukan penyetelan lebih lanjut pada sumbu Y untuk memperoleh kinerja yang konsisten pada mode supervisi. Pengukuran berat beban juga masih terbatas pada kondisi diam dan memerlukan pengembangan agar stabil saat crane beroperasi.

DAFTAR PUSTAKA

[1]. N. S. Bintang, H. S. Bintang, and G. A. Bintang, “Penerapan kesehatan dan keselamatan kerja (K3) dalam sektor manufaktur: Analisis kasus pada industri di Indonesia,” Jurnal Keilmuan dan Aplikasi Bidang Tekstil dan Manajemen Industri, vol. 8, no. 2, pp. 96–103, 2025.
[2]. T. L. Rusli and Isdayani, “Sistem kendali posisi menggunakan feedback control pada overhead crane,” Politeknik Negeri Ujung Pandang, 2024.
[3]. W. R. Kurnaiawan, G. Anindita, and R. M. Dhani, “Analisis kecelakaan pekerjaan lifting dengan overhead crane menggunakan metode ECFA, Fishbone, dan Pareto Analysis,” Journal of Safety, Health, and Environmental Engineering, vol. 1, no. 1, pp. 1–6, 2023.
[4]. K. S. Hong and U. H. Shah, Dynamic and Control of Industrial crane. Singapore: Springer, 2018.
[5]. R. A. Kulweic, Material Handling Handbook. New York, NY, USA: John Wiley & Sons, 1985.
[6]. K. Ogata, Modern Control Engineering, 3rd ed. New Jersey, NJ, USA: Prentice Hall, 1997.
[7]. A. Bahga and V. Madisetti, Internet of Things: A Hands-on Approach. Hyderabad, India: Universities Press Private Limited, 2015.
[8]. V. Lampkin, W. T. Leong, L. Olivera, S. Rawat, N. Subrahmanyam, and R. Xiang, Building Smarter Planet Solutions with MQTT and IBM WebSphere MQ Telemetry. New York, NY, USA: IBM Corporation, 2012.
[9]. Arduino, “Arduino Mega 2560 Rev3 (SKU: A000067),” Arduino Store, Monza, Italy, 2026. [Online]. Available: https://store.arduino.cc/products/arduino-mega-2560-rev3
[10]. Espressif Systems, “ESP32 series datasheet,” Espressif Systems, 2024. [Online]. Available: https://docs.espressif.com/projects/esp-dev-kits/en/latest/esp32/esp32-devkitc/user_guide.html#functional-description.
[11]. Arduino, "Micro," Arduino Documentation, 2026. [Online]. Tersedia: https://docs.arduino.cc/hardware/micro. [Diakses: 11 Juli 2026].

[12]. B. D. Hadi dan R. Kurnianingtyas, "Perancangan Sistem Kendali Motor DC dengan Fuzzy Menggunakan Simulink Matlab," Elektriese: Jurnal Sains danTeknologi Elektro, vol. 15, no. 01, Apr. 2025, doi: 10.47709/elektriese.v15i01.5778. [Online]. Tersedia: https://jurnal.itscience.org/index.php/elektriese/article/download/5778/4386/27246. [Diakses: 11 Juli 2026].
[13]. A. Hughes and B. Drury, Electric Motors and Drives: Fundamentals, Types and Applications, 4th ed. Oxford, U.K.: Newnes, 2013.
[14]. U. Latifa and J. S. Saputro, "Perancangan robot arm gripper berbasis Arduino Uno menggunakan antarmuka LabView," Barometer, vol. 3, no. 2, pp. 138–141, 2018.
[15]. STMicroelectronics, "L298 dual full-bridge driver datasheet," STMicroelectronics, 2024. [Online]. Available: https://www.st.com/resource/en/datasheet/l298.pdf
[16]. Avia Semiconductor, "HX711 24-bit analog-to-digital converter (ADC) for weigh scales datasheet," Avia Semiconductor, 2024. [Online]. Available: https://cdn.sparkfun.com/datasheets/Sensors/ForceFlex/hx711_english.pdf
[17]. R. Gazali and Ismuharram, "Prototype modul latih pneumatic dan motor DC berbasis PLC," Jurnal Elektro & Informatika Swadharma, vol. 1, no. 2, pp. 26–32, 2021.
[18]. TDK InvenSense, "MPU-6000 and MPU-6050 product specification," TDK InvenSense, 2024. [Online]. Available: https://invensense.tdk.com/wp-content/uploads/2015/02/MPU-6000-Datasheet1.pdf
[19]. Phidgets Inc., "3133 - Micro load cell (0-5kg) - CZL635 datasheet," Phidgets Inc., May 13, 2011. [Online]. Available: https://www.phidgets.com/docs/Load_Cell_Primer
[20]. Endang and R. Hidayat, "Rancangan bangun sistem otomatis pengalih sumber daya cadangan DC berbasis baterai pack lithium," Aisyah Journal of Informatics and Electrical Engineering, vol. 6, no. 1, pp. 1–10, 2025.
[21]. W. R. Pratama, B. Yulianti, and A. Sugiharto, "Prototype smart parking modular berbasis Internet of Things," Jurnal Teknologi Industri, vol. 11, no. 2, pp. 52–59, 2022.
[22]. M. Banzi and M. Shiloh, Getting Started with Arduino, 3rd ed. New York, NY, USA: Maker Media, 2014.
[23]. M. B. Ulum and F. Badri, "Sistem monitoring cuaca dan peringatan banjir berbasis IoT dengan menggunakan aplikasi MIT App Inventor," Jurnal Informatika dan Teknik Elektro Terapan, vol. 11, no. 3, pp. 319–328.
[24]. Eclipse Foundation, "Eclipse Mosquitto," Mosquitto.org, 2026. [Online]. Available: https://mosquitto.org/.