

LANDASAN TEORI

2.1 Tinjauan Pustaka

Peneliti mengumpulkan penelitian sebelumnya yang akan dijadikan sumber referensi sekaligus penunjang pada penelitian ini. Berikut tinjauan pustaka yang digunakan dapat dilihat pada Tabel 2.1.

Tabel 2. 1 Tinjauan Pustaka

Nomor Literatur	Penulis	Tahun	Judul
Literatur 1	Agung Pangestu	2021	Pengembangan Firmware Pada Sub Controller Robot Sepak Bola Humanoid Krakatau Fe Menggunakan Protokol Dynamixel 2.0
Literatur 2	Robin Mirrow	2020	Embedded debug Interface for Debug
Literatur 3	Klara Branstatter,BSc	2020	Konstruktion einer Sandboxfür die Analyse von Kontrollalgorithmen unddas Training von Robotern
Literatur 4	Marc Bestmann, Jesper Guldenstein,and Jianwei Zhang	2019	High Frequency Multi Bus Servo and Sensor Communication Using Dynamixel Protocol
Literatur 5	Xinkui FENG	2019	Firmware development for Roombots
Literatur 6	Sergi Martnez Snchez	2018	A Programable Six-Axis Compilant Device Based On A GoughStewart Parallel Platform

2.1.1 Tinjauan Literatur 1

(Pangestu, 2021), Pada penelitian ini, berhasil mengirimkan paket data dan dapat mengendalikan servo dynamixel menggunakan board KSC 2.0 menggunakan protokol dynamixel 2.0. Mikrokontroler yang digunakan adalah Teensy 4.0 dan menggunakan FT232RL sebagai konversi dari komunikasi serial dari USB ke UART dengan maximum bandwidth 3Mbps. Pengujian pergerakan servo dynamixel yang dilakukan menggunakan servo dynamixel jenis MX 64T.

2.1.2 Tinjauan Literatur 2

(Mirow, 2020), Pada penelitian ini, berhasil mengendalikan servo dynamixel menggunakan jenis board STM32F7508-DK dengan menerapkan protokol dynamixel 2.0 untuk komunikasinya. Untuk mengkonversikan komunikasi serial dari USB ke UART menggunakan FT232R. Untuk mengirimkan data sebesar 8 KiB digunakan kontroler DMA (Direct Memory Access). Tujuan penelitian ini adalah menerapkan pada robot Wolfgang menggunakan RS485 yang kompatibel. Dalam firmware pada robot Wolfgang dapat menampung data sebesar 2 MBd.

2.1.2 Tinjauan Literatur 3

Menurut(Klara Brandstätter, 2020) membuat firmware baru yang dapat dikendalikan dengan jarak jauh dengan data. File firmware menggunakan embedded menggunakan Bahasa pemrograman C. File Firmware baru dapat dikirim melalui terminal menuju ke pengontrol. Program baru dapat langsung diterapkan pada robot. Selanjutnya dilakukan uji gerakan. Setelah melakukan uji gerakan yaitu membuat program ulang yang di diperbarui berisi data dynamixel. Dengan proses ini dapat di otomatisasi di masa depan. Penelitian ini menggunakan servo dynamixel jenis AX 12 sebanyak 15 servo sebagai penggerak dan menggunakan

protokol dynamixel 1.0. Kontroler yang digunakan adalah CM 530 yang didalamnya terdapat microprocessor berbasis ARM Cortex STM32F103RE. Untuk komunikasi wireless menggunakan modul ZigBee.

2.1.3 Tinjauan Literatur 4

(Bestmann, Guldenstein and Zhang, 2019), Pada penelitian ini berfokus pada keterbatasan kecepatan ketika berkomunikasi saat melakukan baca data sensor dan memerintahkan actuator menggunakan multi-bus pada komunikasi dynamixel. Adapun cara yang digunakan adalah meningkatkan frekuensi loop lebih dari 1 kHz. Pada robot humanoid melakukan pembacaan data servo dan memberikan perintah bersifat secara terus menerus dilakukan. Frekuensi yang meningkat akan menghasilkan waktu reaksi yang lebih cepat juga. Pada penelitian ini menggunakan jenis servo dynamixel tipe MX-64 sebanyak 20. Satu servo dynamixel memiliki kecepatan untuk membaca posisi yaitu sebesar 10 byte. Tetapi pada setiap servo dynamixel dibutuhkan 4 byte untuk memberikan posisi tujuan. Pada satu bus memiliki performa terbaik yaitu menggunakan baud rate sebesar 1 Mbaud sampai 2 Mbaud dan maksimal baud rate 4 Mbaud yang dapat digunakan. Pada penelitian ini berhasil meningkatkan jumlah bus dan kecepatan baud dengan hasil pencapaian terbaik yaitu sebesar 1373 Hz.

2.1.4 Tinjauan Literatur 5

(Feng, 2019), Menurut pada penelitian ini berfokus pada pengembangan firmware embedded pada mikrokontroler di dalam Roobots. Protokol komunikasi yang digunakan adalah protokol ASCII dengan pengujian komunikasi yaitu menggunakan RS232 dan RS485. Penelitian ini menggunakan mikrokontroler dsPIC33FJ128MC802 dari perusahaan Microchip. Modul yang digunakan untuk

komunikasi PC ke Roombots modul UART(Universal Asynchronous Receiver Transmitter). Hasil dari penelitian ini peneliti merancang library libp33 untuk mikrokontroler dsPIC33FJ128MC802. Komunikasi ASCII menggunakan standar RS485 sehingga mudah untuk di debug. Merancang firmware tiga board elektronik yaitu board komunikasi, board ACM dan board sensor. Menerapkan plot kecil pada MATLAB yang dapat digunakan untuk mengukur parameter mikrokontroler.

2.1.5 Tinjauan Literatur 6

(Ciències and Sánchez, 2018), menurut melakukan penelitian berfokus pada pengembangan sistem yang dapat mengontrol gerakan paralel dengan gaya eksternal. Dan menjelaskan bagaimana komunikasi antara board actuator dan komputer. Pada penelitian ini menggunakan mikrokontroler Open CM 904 sebagai kendali . Aktuator yang digunakan adalah servo dynamixel sebanyak 6 dengan tipe AX 12+. Untuk mengkonversikan komunikasi serial dari USB atau RS232 (serial port) ke RS485 atau TTL (*Half-Duplex*) menggunakan USB2Dynamixel. Komunikasi yang digunakan adalah protokol dynamixel 1.0. Firmware berfungsi untuk menghubungkan antara perangkat keras dan perangkat lunak berfungsi dengan baik. Firmware yang dibuat untuk mempermudah komunikasi dengan baik.

2.2 Firmware

Firmware merupakan lapisan antara perangkat keras (*hardware*) dan perangkat lunak (*software*) untuk mengendalikan proses komponen dalam bentuk fisik menggunakan program yang ditentukan oleh pengguna dan informasi proses lainnya(Wang *et al.*, 2016). Firmware pada dasarnya adalah menggabungkan beberapa instruksi dan data yang diperlukan untuk membuat perangkat berfungsi. Firmware berfungsi untuk mengendalikan perangkat keras dapat dioperasikan. Pada

Krakatau Sub Controller (KSC) firmware berfungsi untuk mengendalikan komponen yang ada pada Krakatau Sub Controller (KSC) seperti servo dynamixel dan sensor. Pada penelitian ini, peneliti mengembangkan firmware pada KSC versi 2.1 menggunakan protokol dynamixel 1.0.

2.3 Servo Dynamixel

Dynamixel merupakan jenis motor servo yang dikembangkan oleh robotis, yang berasal dari perusahaan asal Korea Selatan. Jenis ini sering digunakan dalam pembuatan robot termasuk robot sepak bola humanoid. Dynamixel dapat disebut sebagai servo yang cerdas (*smart servo*) karena adanya mikroprosesor memiliki kemampuan seperti posisi servo, beban kerja, suhu pada servo serta komunikasi dengan kontroler.(Choiri and Widiawan, 2015). Servo dynamixel masing-masing memiliki pin. Pin tersebut berfungsi sebagai konektor sekaligus arus dan transmisi data yang terhubung dari pin ke pin lainnya. Pada penelitian ini, peneliti menggunakan servo dynamixel jenis MX-28 sebagai pengujian. Gambar dari servo dynamixel dapat dilihat pada gambar 2.1



Gambar 2. 1 Aktuator Dynamixel Tipe MX-28T
Sumber : <https://emanual.robotis.com/>

2.4 Protokol Dynamixel

Menurut (leonardo catur K.E.P., 2012), protokol merupakan sekumpulan aturan yang mendefinisikan beberapa fungsi membuat relasi, mengirim pesan dan memecahkan berbagai masalah komunikasi data agar komunikasi data dapat berjalan dengan baik. Dynamixel sering digunakan pada pembuatan robot dan dapat dikendalikan menggunakan pengontrol tertentu. Untuk mengendalikan dynamixel komunikasi dapat menggunakan protokol yang sesuai. Dynamixel menggunakan protokol komunikasi serial *half duplex Universal Asynchronous Receiver Transmitter* (UART) dengan 8 bit, 1 stop bit, dan tidak bit paritas. Paket pada dynamixel memiliki dua jenis paket yaitu paket instruksi dan paket status. Perintah yang dikirimkan main controller menuju ke dynamixel disebut paket instruksi. Sedangkan respon balik dari paket yang dikirimkan dari dynamixel menuju ke main controller disebut paket status.

Pada penelitian ini, peneliti menggunakan protocol dynamixel versi 1.0. Berikut struktur instruksi paket dan status paket dapat dilihat pada Tabel 2.2 dan Tabel 2.3

Tabel 2. 2 Instruksi paket pada protokol dynamixel 1.0

Header1	Header2	Packet ID	Length	Instruction	Param 1	...	ParamN	Checksum
0Xff	0xFF	Packet ID	Length	Instruction	Param 1	...	Param N	Checksum

Tabel 2. 3 Paket status pada protokol dynamixel 1.0

Header1	Header2	Packet ID	Length	Error	Param 1	...	Param N	Checksum
0Xff	0xFF	Packet ID	Length	Error	Param 1	...	Param N	Checksum

2.4.1 Header

Header digunakan sebagai awalan paket yang akan dikirim. Jika header sesuai dengan aturan yang telah ditetapkan maka selanjutnya adalah menuju ke Packet ID. Jika tidak maka akan diperiksa nilai header tersebut.

2.4.2 Packet ID

ID Packet merupakan nomor yang unik yang terdapat pada servo dynamixel. ID paket pada setiap dynamixel memiliki ID yang berbeda beda. Pemberian ID pada paket instruksi dan paket status berfungsi untuk mempermudah kontroler mengirimkan paket sesuai dengan alamat tujuan. Pemberian ID yang unik dapat membedakan alamat pada masing masing dynamixel. Pada dynamixel ID memiliki batasan Id yang dapat digunakan.

1. *Range*, 0-253 (0x00 – 0xFD) jumlah ID yang dapat digunakan sebanyak 254 ID.
2. *Broadcast ID*, ID 254 berfungsi untuk menghubungkan semua perangkat untuk mengeksekusi Paket Instruksi.

2.4.3 Length

Length merupakan panjang data dari paket yang dikirim. Length akan memberitahukan ukuran byte dari instruction, parameter, dan Checksum. Pada protocol dynamixel 1.0 mempunyai rumus sebagai berikut.

$$\text{Length} = \text{Parameter (N)} + 2$$

2.4.4 Instruction

Pada protocol dynamixel 1.0 instruction berfungsi untuk memberitahukan perintah apa yang akan dilakukan. Berikut instruction yang terdapat pada protokol dynamixel 1.0 dapat dilihat pada Tabel 2.4

Tabel 2. 4 Instruction

Value	Instruction	Description
0x01	Ping	Berfungsi untuk melakukan pemeriksaan terhadap paket yang dikirim telah sesuai dengan tujuan paket dengan ID yang sama.
0x02	Read	Berfungsi untuk membaca paket data.
0x03	Write	Berfungsi untuk menulis paket data
0x04	Reg Write	Berfungsi untuk mendaftarkan paket ke status siaga, selanjutnya paket akan diproses melalui perintah action
0x05	Action	Berfungsi untuk melakukan proses dari paket yang telah terdaftar
0x06	Factory Reset	Berfungsi untuk melakukan mengembalikan ke dalam bentuk pengaturan awal (default) table kontrol
0x07	Reboot	Berfungsi untuk melakukan proses seperti awal (restart)
0x083	Sync Write	Berfungsi untuk menulis data dari alamat dan panjang yang sama pada beberapa perangkat
0x092	Bulk Read	Berfungsi untuk membaca data dari alamat dan panjang yang berbeda pada beberapa perangkat

2.4.5 Parameters

Parameter merupakan data untuk membantu instruksi. Dalam penggunaannya parameter berbeda beda tergantung jenis instruksi apa yang akan digunakan misalnya seperti ping, write, read dll.

2.4.6 Checksum

Checksum berfungsi untuk melakukan pemeriksaan terhadap paket data yang dikirim apakah mengalami kendala seperti kerusakan paket data selama komunikasi sedang berlangsung. Pada protokol dynamixel 1.0 checksum dapat dihitung dengan menggunakan rumus sebagai berikut.

$$\text{Checksum} = \sim(\text{ID} + \text{Length} + \text{Instruction} + \text{Parameter 1} + \dots + \text{Parameter (N)})$$

2.4.7 Status Packet

Paket status merupakan paket kembalian dari paket yang dikirim dari perangkat menuju ke main control. Paket status memiliki struktur tabel yang sama dengan paket instruksi. Yang membedakan adalah adanya instruction di instruction paket dan adanya error pada paket status. Berikut adalah tabel status kesalahan (error) ketika sedang melakukan pengoprasian pada dynamixel. Adapun status kesalahan dapat dilihat pada Tabel 2.5.

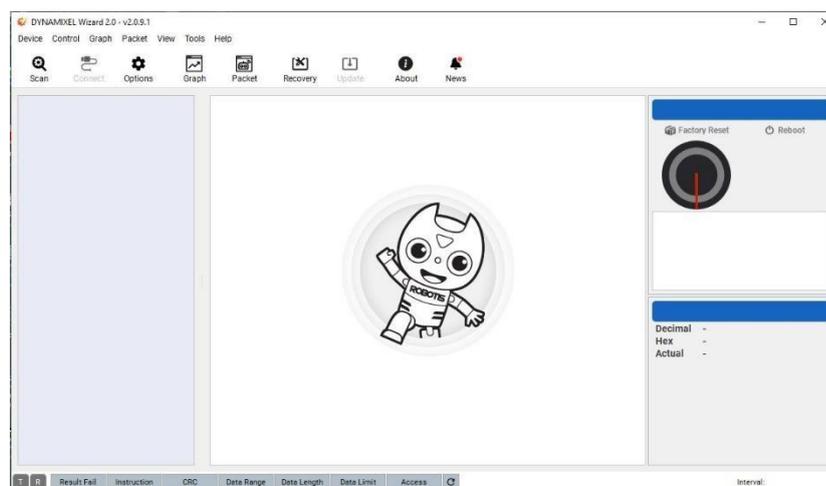
2. 5 Tabel status paket

Bit	Error	Description
Bit 7	0	
Bit 6	Instruction Error	Kesalahan ketika mengirim instruksi yang tidak ditentukan dan memberikan instruksi tanpa instruksi Reg Write
Bit 5	Overload Error	Kesalahan ketika beban saat ini tidak dapat dikontrol dengan pengaturan torsi
Bit 4	Checksum Error	Kesalahan ketika paket yang dikirim salah
Bit 3	Range Error	Kesalahan ketika instruksi tidak dalam jangkauan yang digunakan
Bit 2	Overheating Error	Kesalahan ketika suhu pada dynamixel diluar table kontrol

Bit 1	Angle Limit Error	Kesalahan ketika posisi yang tertulis di luar Batasan sudut CW dan CCW
Bit 0	Input Voltage Error	Kesalahan ketika pemberian tegangan diluar tablekontrol

2.5 Dynamixel Wizard

Dynamixel wizard merupakan software pengembangan Robotis yang digunakan untuk memecahkan masalah dan memperbarui servo dynamixel. Dynamixel wizard sendiri dapat dioperasikan dari berbagai sistem operasi seperti windows, mac, maupun linux. Dynamixel wizard berfungsi sebagai mengelola firmware (memperbarui dan memulihkan), memeriksa status pengontrol dan perangkat peripheral (melakukan tes), dan mengatur mode yang dibutuhkan (pengaturan). Dynamixel wizard dapat digunakan pada protocol dynamixel versi 1.0 dan versi 2.0. pada penelitian ini, peneliti menggunakan software dynamixel wizard 2.0. Peneliti melakukan pengiriman paket data berupa hexadecimal yang selanjutnya dapat dilihat dalam bentuk table sehingga mudah untuk dipahami. Berikut tampilan software dynamixel wizard 2.0 dapat dilihat pada gambar



Gambar 2. 2 Tampilan Tools Dynamixel Wizard 2.0

Sumber:<https://emanual.robotis.com/>

2.6 Teensy

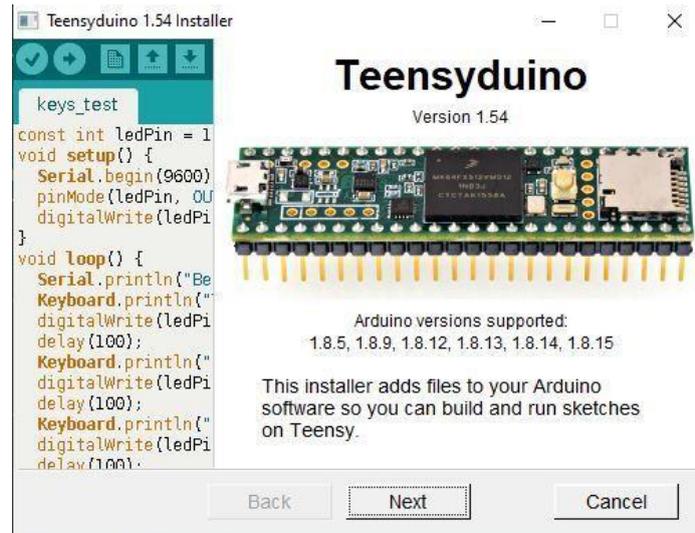
Teensy merupakan salah satu jenis *board* sistem pengembangan mikrokontroler berbasis usb dengan ukuran yang kecil tetap dapat mengimplementasikan banyak jenis proyek(PJRC, 2021a). Teensy dibangun dengan mikrokontroler ARM Cortex-M7 dengan *clock speed* yaitu 600 MHz. Board Teensy menyediakan akses 32 pin GPIO(*General Purpose Input-Output*) dan memiliki converter analog ke digital dan juga realtime(Lambert, *et al.*, 2021). Pada penelitian ini. Peneliti menggunakan *board* teensy versi 4.0 yang akan dipasang pada Krakatau Sub Controller versi 2.1. Berikut gambar teensy versi 4.0.



Gambar 2. 3 Teensy versi 4.0
Sumber: <https://www.pjrc.com/>

2.6 Teensyduino

Teensyduino merupakan software tambahan Arduino IDE, software ini digunakan untuk menjalankan sketch pada Teensy dan Teensy++(PJRC, 2021). Pada *Krakatau Sub Controller* (KSC) versi 2.1 menggunakan microcontroller teensy 4.0 sehingga memerlukan software tersebut. Berikut tampilan dari Teensyduino dapat dilihat pada gambar



Gambar 2. 4 Tampilan Teensyduino

Sumber: <https://www.pjrc.com/>

2.7 Bahasa Pemrograman C++

Bahasa pemrograman C++ merupakan bahasa pemrograman tingkat tinggi (high-level). Bahasa C++ sendiri sering digunakan untuk penulisan program seperti embedded maupun robotic. Pada tahap ini, peneliti menggunakan bahasa pemrograman C++ untuk menulis kode program yang akan di compile oleh software teensyduino atau Arduino IDE. Berikut logo dari C++ dapat dilihat pada gambar.

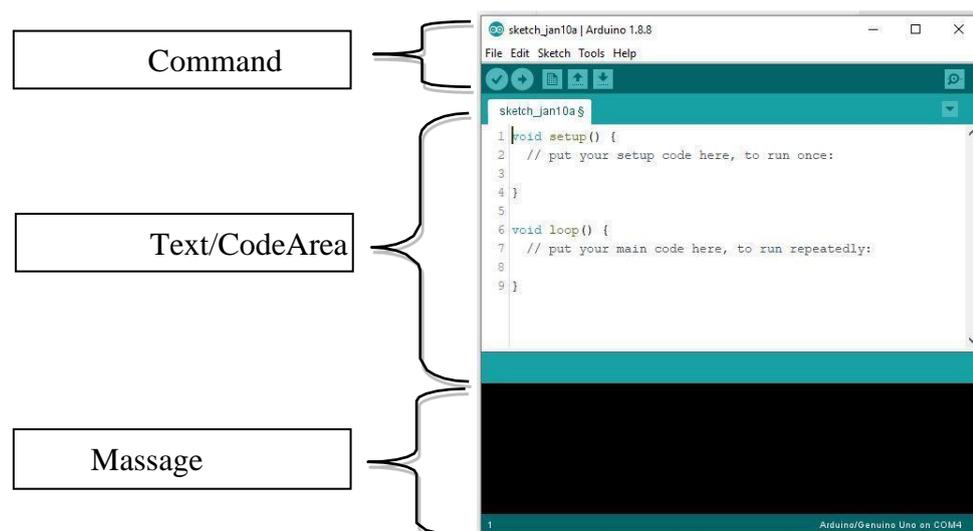


Gambar 2. 5 Logo C++

Sumber: <https://www.cplusplus.com/>

2.8 Arduino Ide

Pada penelitian ini, peneliti menggunakan software Arduino IDE yang merupakan software text editor yang dapat digunakan seperti membuat baru dan mengedit kode program yang akan digunakan sebagai compile program bahasa C++. Arduino IDE adalah software serangkaian instruksi yang memberikan informasi perangkat keras apa yang akan dilakukan dan bagaimana cara melakukan (Badamasi, 2014). Arduino IDE terbagi menjadi 3 bagian utama yaitu : *command area*, *text/Code area*, dan *message window area*. Berikut tampilan software Arduino IDE dapat dilihat pada gambar



Gambar 2. 6 Logo Arduino

Sumber : <https://www.arduino.cc/>

2.9 Hal Duplex

Half duplex merupakan komunikasi serial yang disebut UART (Universal Asynchronous Receiver Transmitter). Half duplex adalah sebuah metode komunikasi serial pengiriman (RxD) dan penerima (TxD) agar dapat saling bertukar informasi tetapi dalam waktu tidak bersamaan(Prayudha, Pranata and Prastyo, 2020). Komunikasi data half duplex dipresentasikan terbagi pola bit dan urutan bit, selanjutnya dikirim secara satu bit demi satu bit sesuai dengan alamat tujuan. Supaya data yang dikirimkan sesuai dengan alamat tujuan maka harus memiliki Slave ID dan dilanjutkan data yang akan dikirim, semua alamat akan menerima data tersebut jika data yang dikirimkan memiliki ID yang yang berbeda maka data diabaikan tetapi jika memiliki ID yang sama maka data tersebut akan diproses(Vitria, 2008).