



BAB I

PENDAHULUAN

1.1 Latar Belakang

Logistik merujuk pada penyimpanan barang dalam jumlah besar dan memiliki keterkaitan erat dengan sektor industri, termasuk industri jasa pengiriman logistik. Penyedia jasa logistik sering menghadapi tantangan dalam mempertahankan performa manajemen logistik di gudangnya. Masalah ini sering muncul ketika persediaan di gudang meningkat, ditambah dengan kurangnya manajemen dan komunikasi saat proses pembongkaran logistik ketika angkutan tiba, serta di beberapa perusahaan, angkutan logistik hanya sampai di gerbang gudang. Hal ini membuat pekerja kesulitan menemukan barang di dalam gudang. Yunardi dkk. (2021) menyatakan bahwa di era saat ini, industri jasa pengiriman atau logistik telah tumbuh dengan pesat, yang dipicu oleh meningkatnya permintaan konsumen terhadap barang dan jasa, baik dari individu maupun perusahaan.

Kemajuan terkini dalam pengembangan mobile robot telah membuka banyak kemungkinan penerapannya di lingkungan industri. Mobile robot dirancang untuk bergerak melalui suatu lingkungan dan dapat menentukan jalur geraknya sendiri. Mobile robot dibagi menjadi dua kelompok yaitu robot beroda dan robot berkaki, tergantung pada sistem geraknya (Latif dkk., 2019). Mobile robot berpotensi dimanfaatkan untuk industri pergudangan otomatis, seperti memuat, mengangkut, dan membongkar barang-barang rapuh dari satu tempat ke tempat lain secara terus menerus dan mandiri. Umumnya metode untuk pergerakan pada robot industri dan juga robot pelayan yakni dengan memanfaatkan roda yang mana membutuhkan lebih sedikit energi, waktu pengembangan dan biaya daripada pergerakan berjalan kaki atau terbang (Tajti et.al., 2014). Salah satu keberhasilan implementasi aplikasi robot telah didemonstrasikan oleh robot Amazon Kiva, yang digunakan sebagai mesin transportasi otonom untuk menyortir dan mengantarkan barang yang dikirim.



Robot tersebut dapat melakukan perjalanan ke sebuah rute navigasi yang telah ditentukan sebelumnya dalam koordinat planar dan mampu memindahkan bahan dari rak penyimpanan ke pemenuhan zona.

Salah satu sistem yang paling umum digunakan pada mobile robot beroda ialah sistem penggerak diferensial. Sistem ini merupakan non-holonomic, yakni sistem yang hanya mampu bergerak maju, mundur, dan berputar. Sistem penggerak tersebut memiliki batasan secara kinematika sehingga robot perlu melakukan manuver dengan cara mengubah arah hadap apabila robot ingin bergerak ke satu tujuan yang tidak hanya memerlukan gerakan maju dan mundur saja. Hal ini juga kurang efisien secara mobilitas karena robot tidak dapat menjangkau segala arah tanpa melakukan manufer atau gerak memutar untuk meraih tujuannya. Selain sistem penggerak differensial terdapat juga sistem penggerak robot yang disebut holonomic. Holonomic robot merupakan salah satu jenis robot yang mempunyai tiga atau lebih roda omnidirectional yang memiliki kemampuan untuk bergerak bebas di segala arah tanpa perlu melakukan manuver atau mengubah orientasi.

Namun masalah utama pada holonomic robot yakni bagaimana membuat robot dapat bergerak menuju titik-titik tertentu secara otonom. Sistem kendali posisi pada robot dapat diterapkan karena dalam aplikasinya robot dapat mengarahkan dirinya sendiri untuk bergerak ke titik-titik tertentu sesuai dengan masukan yang diberikan. Di sisi lain, sistem kendali posisi merupakan aspek kunci dalam navigasi robot. Sistem kendali posisi yang akurat memungkinkan robot untuk mengetahui posisinya dalam lingkungan sekitarnya, yang menjadi prasyarat penting untuk navigasi yang efisien dan aman. Salah satu metode yang umum digunakan untuk sistem kendali posisi adalah teknik odometry, yang memanfaatkan informasi pergerakan relatif dari roda atau sensor lainnya untuk mengestimasi posisi robot (Yunardi dkk., 2021).

Berdasarkan uraian permasalahan tersebut, penulis mencoba mengajukan ide melalui sebuah judul penelitian “**RANCANG BANGUN KENDALI POSISI BERBASIS WHEEL ODOMETRY PADA HOLONOMIC MOBILE ROBOT BERODA TIGA**” untuk menyelesaikan permasalahan tersebut. Dengan



memadukan kemampuan holonomic robot dalam navigasi yang fleksibel dengan sistem kendali posisi yang andal berbasis wheel odometry, diharapkan dapat diciptakan robot yang mampu beroperasi secara efisien sesuai dengan kebutuhan operator dibandingkan dengan robot non-holonomic seperti penggerak differensial yang tidak leluasa untuk bergerak di lingkungannya.

1.2 Rumusan Masalah

Mengacu pada uraian latar belakang penelitian, maka dapat dirumuskan masalah sebagai berikut :

1. Bagaimana merancang desain holonomic mobile robot beroda tiga dengan sistem kendali posisi berbasis wheel odometry?
2. Bagaimana menerapkan sistem kendali posisi pada holonomic mobile robot beroda tiga untuk meraih posisi tujuannya?
3. Bagaimana kinerja sistem kendali posisi pada holonomic mobile robot beroda tiga berdasarkan wheel odometry?

1.3 Tujuan Penelitian

Tujuan yang ingin dicapai dalam penelitian ini :

1. Menghasilkan prototipe *holonomic mobile robot* beroda tiga dengan sistem kendali posisi berbasis *wheel odometry*.
2. Menerapkan *inverse kinematic* pada sistem kendali posisi berbasis *wheel odometry* pada *holonomic mobile robot* beroda tiga.
3. Menguji kinerja sistem yang telah dirancang dengan membandingkan simulasi model matematis odometry robot.

1.4 Manfaat Penelitian

Penelitian ini memiliki beberapa manfaat sebagai berikut :

1. Bagi penulis, penelitian ini menjadi bentuk pengabdian, dan sebagai evaluasi untuk mengukur kemampuan potensi diri sendiri, serta dapat



menerapkan disiplin ilmu teori dan aplikasi yang telah didapatkan selama masa perkuliahan.

2. Bagi mahasiswa, penelitian ini dapat memberikan kontribusi dan dapat digunakan sebagai dasar untuk pengembangan penelitian di bidang robotika dan otomasi.
3. Bagi Departemen Teknik Elektro, terkhusus bidang teknik kendali, penelitian ini diharapkan dapat memberikan sumbangsi positif terhadap ilmu pengetahuan mata kuliah sistem instrumentasi elektronik, sistem kendali digital, dan sistem berbasis mikroprosesor.
4. Bagi kemajuan ilmu pengetahuan dan teknologi, penelitian ini diharapkan bisa menjadi pemicu kreativitas bagi para generasi muda untuk terus mengembangkan ilmu pengetahuan dan teknologi kedepannya.

1.5 Ruang Lingkup

Adapun ruang lingkup pada penelitian ini, yaitu :

1. Sistem yang dibangun bersifat *prototype*
2. Parameter yang dikendalikan berupa posisi x_{robot} , y_{robot} dan θ_{robot} berdasarkan *wheel odometry*, serta kecepatan rotasi masing-masing motor berdasarkan *inverse kinematic*.
3. Parameter yang dapat diubah secara variatif merupakan target posisi x_{robot} dan y_{robot} .
4. Parameter yang tidak dapat diubah merupakan target posisi θ_{robot} yang akan ditetapkan bernilai 0.
5. Parameter waktu t untuk menentukan kecepatan translasi robot tidak akan ditentukan.
6. Target posisi x_{robot} dan y_{robot} diberikan secara manual melalui *user interface* robot.



BAB II

TINJAUAN PUSTAKA

2.1 Sistem Gerak Robot Beroda

Sistem gerak robot beroda secara umum terbagi dua yaitu non-holonomic dan holonomic. Pada tipe non-holonomic memiliki hanya ada dua derajat kebebasan yang dapat dikontrol yaitu akselerasi (atau pengereman) dan sudut belok roda kemudi. Contoh robot beroda tipe non-holonomic yaitu pada robot dengan differential drive yang mana memiliki kecepatan linear 1 axis dan kecepatan angular sehingga untuk membuat robot bergerak mencapai tujuan koordinat (x, y) harus melakukan manuver atau perubahan orientasi robot.

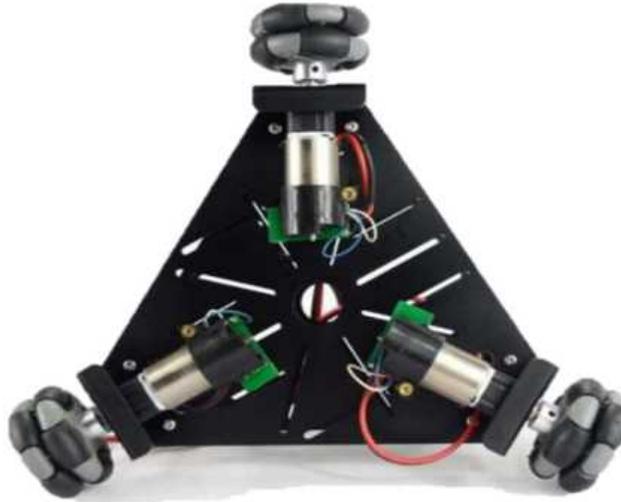


Gambar 1 Contoh model robot non-holonomic

Holonomic Robot adalah robot yang memiliki 3 derajat kebebasan yang dapat dikontrol yaitu akselerasi pada x-axis, akselerasi pada y-axis dan sudut orientasi robot sehingga dapat bergerak ke segala arah tanpa melakukan rotasi sehingga robot memiliki ruang gerak bebas dalam melakukan pergerakan. Robot memungkinkan untuk melakukan pergerakan dengan banyak kombinasi seperti bergerak maju/mundur, geser kanan/kiri dan rotasi secara bersamaan. Hal ini dikarenakan dengan adanya dukungan roda yang digunakan pada robot ini, roda khusus ini mempunyai roda kecil tambahan yang berporos tegak lurus pada roda inti, sehingga roda dapat bergerak segala arah. Sehingga robot dengan metode



holonomic ini dapat melakukan pergerakan/mobilisasi robot lebih cepat, karena tanpa bermanuver saat belok (Abseno, 2019).



Gambar 2 Contoh model robot holonomic (Thingsbits, 2018)

2.2 Kinematika Holonomic Robot Beroda Tiga

Kinematika robot adalah studi analisis pergerakan kaki atau roda robot terhadap sistem kerangka koordinat acuan yang diam atau bergerak tanpa memperhatikan gaya yang menyebabkan pergerakan tersebut. Model kinematika robot dalam kinematika dikenal istilah forward kinematics dan inverse kinematics. Forward Kinematics adalah metode untuk menentukan orientasi dan posisi end effector dari besarnya nilai sudut pada aktuator. Sedangkan inverse kinematics merupakan kebalikan dari forward kinematics yaitu metode untuk mengetahui nilai sudut pada aktuator agar end effector dapat mencapai posisi yang dikehendaki (Setiawan dkk, 2015).

Pada holonomic robot beroda tiga diatur penempatan roda yang berjarak sama dengan perbedaan sudut 120 derajat satu sama lain seperti yang ditunjukkan pada Gambar 2. Hubungan antara kecepatan translasi dan kecepatan rotasi pada roda omni-directional adalah sebagai berikut.

$$v = r\dot{\alpha} \quad (1)$$

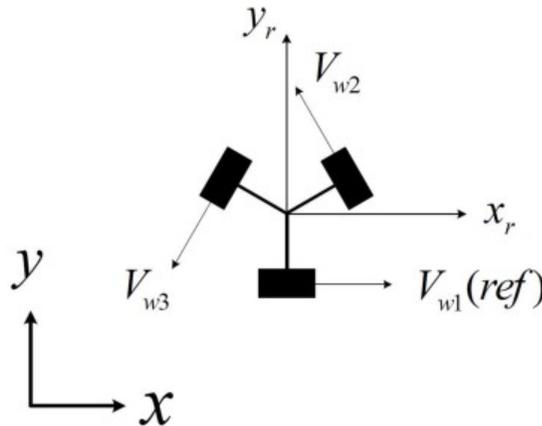


dimana,

r = jari-jari roda

$\dot{\alpha}$ = kecepatan angular motor

v = kecepatan translasi roda



Gambar 3 Representasi Kinematik dari Sistem Penggerak robot holonomic tiga roda omni-directional (Inthiam dan Deelertpaiboon, 2014)

Oleh karena itu, vektor kecepatan sudut roda odometri dapat ditulis sebagai fungsi kecepatan linier dan sudut robot bergerak dalam bentuk matriks seperti yang ditunjukkan pada persamaan berikut.

$$\begin{bmatrix} \dot{\alpha}_1 \\ \dot{\alpha}_2 \\ \dot{\alpha}_3 \end{bmatrix} = \frac{1}{r} \begin{bmatrix} \cos 0 & \sin 0 & L \\ \cos 120 & \sin 120 & L \\ \cos 240 & \sin 240 & L \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \omega \end{bmatrix} \quad (2)$$

dimana,

L = jarak antara roda dan titik pusat robot

\dot{x} = kecepatan linear robot pada x-axis

\dot{y} = kecepatan linear robot pada y-axis

ω = kecepatan rotasi robot

$\dot{\alpha}_{(1,2,3)}$ = kecepatan angular masing-masing roda

Untuk melihat kecepatan linear dan rotasi robot bergerak dapat diperoleh dengan menyelesaikan matrix invers dalam persamaan (2). Hasilnya ditunjukkan pada persamaan (3).



$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \omega \end{bmatrix} = r \begin{bmatrix} \frac{2}{3} & -\frac{1}{3} & -\frac{1}{3} \\ 0 & \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} \\ \frac{1}{3L} & \frac{1}{3L} & \frac{1}{3L} \end{bmatrix} \begin{bmatrix} \dot{\alpha}_1 \\ \dot{\alpha}_2 \\ \dot{\alpha}_3 \end{bmatrix} \quad (3)$$

2.3 Three Omni-Wheel Odometry

Odometry adalah penggunaan data dari pergerakan aktuator untuk memperkirakan perubahan posisi secara *real time*. *Odometry* digunakan untuk memperkirakan posisi relatif terhadap posisi awal (Wibowo dkk., 2018). Sesuai dengan model kinematika holonomic robot beroda tiga pada Gambar 3, posisi dan arah hadap robot dapat diperoleh dengan melakukan integral pada kecepatan linear robot dan kecepatan angular roda sesuai dengan persamaan (3).

$$x = r \left(\frac{2}{3} \alpha_1 - \frac{1}{3} \alpha_2 - \frac{1}{3} \alpha_3 \right) \quad (4)$$

$$y = r \left(\frac{1}{\sqrt{3}} \alpha_2 - \frac{1}{\sqrt{3}} \alpha_3 \right) \quad (5)$$

$$\theta = \frac{r}{3L} (\alpha_1 - \alpha_2 - \alpha_3) \quad (6)$$

dimana,

x, y = posisi robot

θ = sudut heading robot

$\alpha_{(1,2,3)}$ = posisi sudut masing-masing roda

Untuk mendapatkan posisi sudut α_1 , α_2 dan α_3 dari rotary encoder, perlu dilakukan konversi nilai karena output encoder memiliki satuan pulsa per putaran. Konversi nilai posisi sudut ke pulsa per putaran ditunjukkan pada persamaan (7) dan (8).

$$C_{position} = \frac{2\pi}{Encoder Resolution} \quad (7)$$

$$C_{heading} = \frac{360}{Encoder Resolution} \quad (8)$$

dimana,

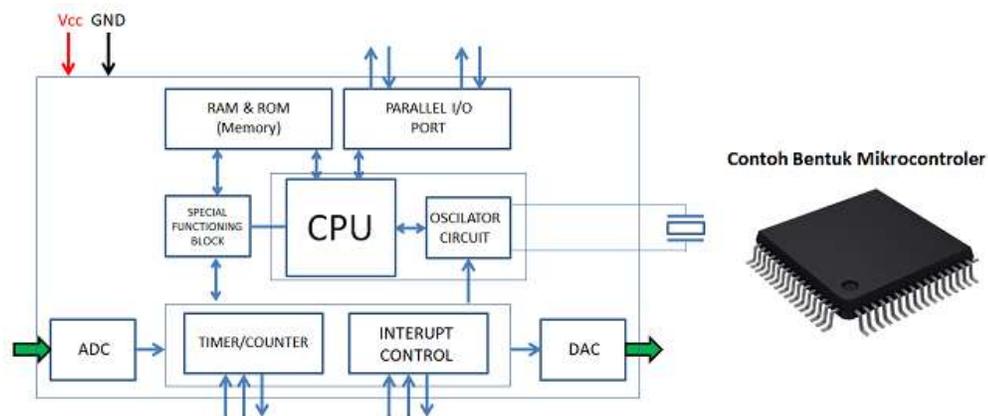


$C_{position}$ = nilai hitungan posisi relatif dari encoder

C_{headi} = nilai hitungan sudut relatif dari encoder

2.4 STM32

Mikrokontroler adalah suatu rangkaian terintegrasi (IC) yang bekerja untuk aplikasi pengendalian. Meskipun mempunyai bentuk lebih kecil dari computer pribadi dan mainframe, mikrokontroler dibangun dengan elemen-elemen yang sama. Mikrokontroler adalah alat yang mengerjakan instruksi-instruksi yang diberikan, artinya bagian utama dari suatu sistem otomatis/terkomputerisasi adalah program didalamnya dibuat oleh programmer. Sadewa dkk. (2015) menyatakan bahwa program menginstruksikan mikrokontroler untuk melakukan jalinan yang panjang dari aksi-aksi sederhana untuk melakukan tugas yang lebih kompleks sesuai keinginan programmer.



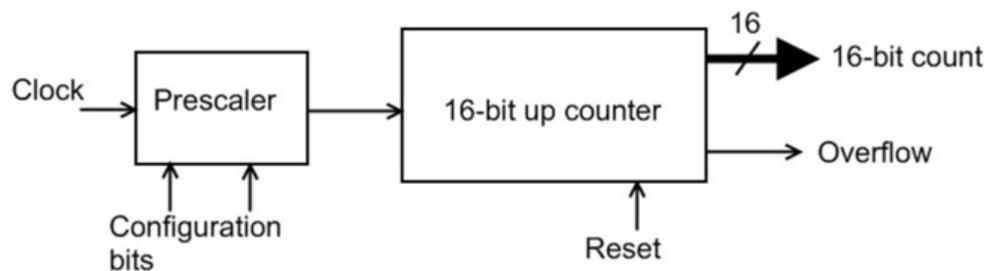
Gambar 4 Struktur Pembangun Mikrokontroler (Rachmat, 2022)

Mikrokontroler STM32 merupakan salah satu *embedded system* yang terdiri dari beberapa komponen yang tertanam dalam satu keping *chip* yang dapat diprogram dengan fungsi tertentu microcontroller terdiri dari beberapa komponen seperti *microprocessor*, *memory*, *port I/O* dan beberapa *peripheral* seperti ADC, *Serial port*, *Timer*, DAC, dan *peripheral* lainnya yang mendukung fungsi pengontrolan (Rachmat, 2022).



2.4.1 Timer

Timer pada mikrokontroler adalah sebuah fitur yang memungkinkan mikrokontroler atau mikroprosesor untuk mengukur waktu atau memicu suatu kejadian secara periodik. Timer ini dapat menjadi bagian dari perangkat keras (*hardware*) yang terintegrasi langsung ke dalam mikrokontroler, atau bisa juga diimplementasikan melalui perangkat lunak (*software*) menggunakan fitur-fitur yang tersedia dalam mikrokontroler. Timer digunakan untuk mengatur waktu dalam sistem tertentu, melakukan tugas-tugas yang berkaitan dengan waktu secara periodik, atau memicu kejadian tertentu setelah interval waktu tertentu.



Gambar 5 Struktur Timer pada STM32

Timer hardware terdiri dari komponen fisik yang terintegrasi langsung ke dalam mikrokontroler. Timer ini memiliki prescaler, counter, dan register kontrol yang memungkinkan pengaturan waktu dengan presisi tinggi. Timer hardware biasanya lebih akurat dan responsif dibandingkan dengan timer software. Timer software dibuat menggunakan perangkat lunak dalam mikrokontroler. Meskipun tidak sepresisi timer hardware, timer software masih dapat digunakan untuk keperluan yang kurang kritis atau aplikasi yang memerlukan waktu yang lebih kasar. Timer software biasanya menggunakan pengaturan counter atau delay loop untuk mengukur waktu.

Timer pada embedded system digunakan dalam berbagai aplikasi, termasuk pengaturan waktu, pembacaan sensor secara berkala, pengendalian

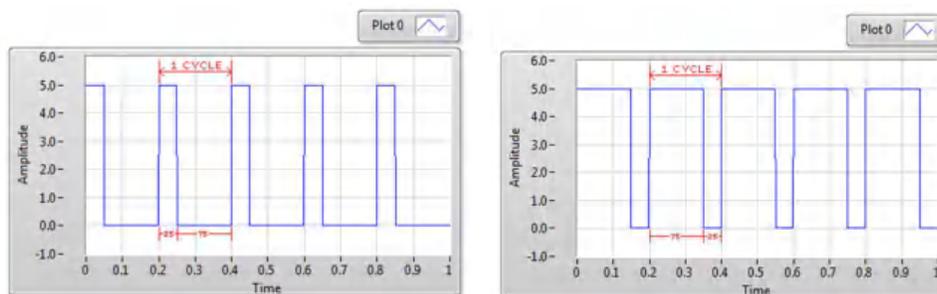


perangkat secara periodik, dan menghasilkan sinyal PWM untuk kontrol motor atau kecerahan lampu LED.

2.4.2 Pulse Width Modulation (PWM)

PWM merupakan teknik untuk mengendalikan sirkuit analog dengan keluaran digital dari mikrokontroler. PWM digunakan untuk menghasilkan sinyal analog menggunakan sumber digital. PWM terdiri dari dua komponen utama yakni *duty cycle* dan frekuensi.

Dengan menyalakan dan mematikan sinyal digital secara berulang-ulang dengan kecepatan yang cukup cepat dan pada siklus kerja tertentu, output akan tampak seperti sinyal analog bertegangan konstan. Misalnya: Untuk membuat sinyal 2V dari sumber digital yang bertegangan HIGH (5V) atau LOW (0). Kita dapat menggunakan PWM dengan *duty cycle* 40% di sini. *Duty cycle* 40% ini akan menghasilkan tegangan rata-rata $5 \times 0,4 = 2V$.



(a)

(b)

Gambar 6 Sinyal PWM: (a) Duty Cycle 25%, (b) Duty Cycle 75%

Untuk menghasilkan sinyal PWM pada mikrokontroler STM32 dibutuhkan Timer yang dikonfigurasi sebagai PWM Generator. Timer diatur untuk menghasilkan frekuensi Timer Clock sebesar 10 kHz berdasarkan persamaan berikut.

$$Timer\ Clock\ (Hz) = \frac{APB\ Timer\ Clock}{Prescaler} \quad (9)$$

$$Frekuensi\ PWM\ (Hz) = \frac{Timer\ Clock}{Counter\ Periode} \quad (10)$$



$$Duty Cycle (\%) = \frac{CCR}{Counter\ Periode} \times 100\% \quad (11)$$

dimana,

APB Timer Clock = frekuensi clock pada peripheral bus timer

Timer Clock = frekuensi timer clock

CCR = nilai counter clock Register

Counter Periode = resolusi counter timer

Diketahui besar frekuensi pada APB Timer yaitu 170 MHz, sehingga Prescaler diberikan nilai 170-1. Alasan dikurang dengan angka 1 dikarenakan Ini merupakan pengaturan sesuai dengan register di STM32. Register Prescaler dan Register ARR diatur sedemikian rupa sehingga menambahkan angka 1 ke nilai seperti yang ditunjukkan pada Gambar 19.

14.4.11 TIM1 and TIM8 prescaler (TIMx_PSC)

Address offset: 0x28

Reset value: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSC[15:0]															
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

Bits 15:0 **PSC[15:0]**: Prescaler value

The counter clock frequency (CK_CNT) is equal to $f_{CK_PSC} / (PSC[15:0] + 1)$.
PSC contains the value to be loaded in the active prescaler register at each update event (including when the counter is cleared through UG bit of TIMx_EGR register or through trigger controller when configured in "reset mode").

Gambar 7 Prescaler Timer STM32

2.4.3 Interrupt Triggered (ITR)

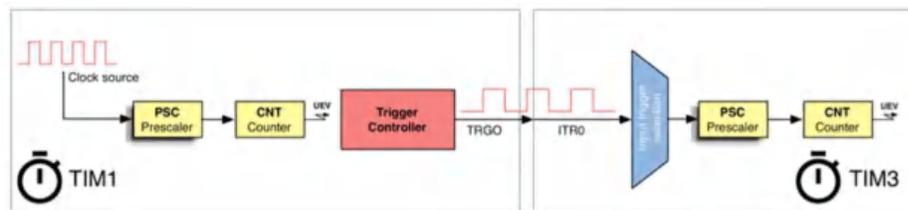
ITR (Interrupt Triggered) pada STM32 Merujuk pada mekanisme dalam mikrokontroler STM32 yang memungkinkan peristiwa pemicu (trigger events) untuk menghasilkan interupsi (interrupt) pada mikrokontroler. Dalam konteks STM32, ITR biasanya mengacu pada bagaimana interupsi diaktifkan dan dipicu oleh berbagai kondisi dalam mikrokontroler, seperti waktu, input eksternal, atau kondisi perangkat keras lain yang terhubung dengan mikrokontroler.



Pada STM32, hampir semua periferal (misalnya timer, ADC, USART, dan lainnya) dapat menghasilkan interupsi. Masing-masing perangkat memiliki saluran interupsi yang dapat diprogram untuk dipicu oleh peristiwa-peristiwa tertentu, seperti perubahan status atau penyelesaian operasi.

Contoh peripheral dengan ITR antara lain:

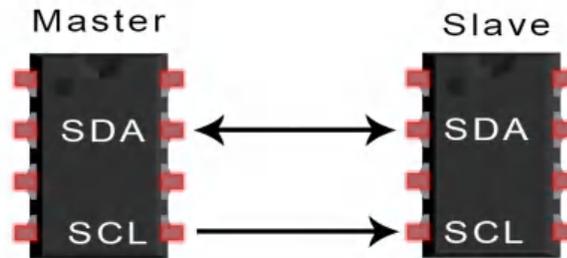
- Timer: Ketika sebuah timer mencapai nilai tertentu atau terjadi peristiwa tertentu (misalnya overflow), ia dapat memicu interupsi.
- ADC: Setelah konversi ADC selesai, interupsi dapat dipicu untuk memberi tahu CPU bahwa hasil ADC sudah siap untuk diproses.
- GPIO: Perubahan status pada pin GPIO dapat memicu interupsi, seperti perubahan status tombol atau sinyal eksternal lainnya.



Gambar 8 Prinsip Kerja Timer Master dan Timer Slave dengan ITR pada STM32

2.4.4 I2C

I2C (Inter-Integrated Circuit) adalah sebuah protokol komunikasi serial yang digunakan untuk menghubungkan perangkat-perangkat elektronika dengan kebutuhan pin yang minimal. I2C diimplementasikan dalam perangkat keras dan perangkat lunak untuk memungkinkan komunikasi antar perangkat dengan mudah.



Gambar 9 Ilustrasi protokol komunikasi I2C

Fitur-fitur yang ada pada peripheral I2C adalah sebagai berikut:

- Multi-master dan Multi-slave: STM32 mendukung konfigurasi multi-master dan multi-slave, memungkinkan beberapa master dan slave berada pada satu bus.
- Kecepatan: I2C pada STM32 mendukung beberapa mode kecepatan, termasuk mode Standar (hingga 100 kHz), Mode Cepat (hingga 400 kHz), Mode Cepat plus (hingga 1 MHz), dan Mode kecepatan tinggi (hingga 3,4 MHz).
- Pengalamatan: STM32 mendukung pengalamatan 7-bit dan 10-bit.
- DMA dan Interrupt: STM32 I2C dapat berinteraksi dengan DMA untuk mentransfer data yang efisien dan juga mendukung interupsi untuk pengelolaan event.

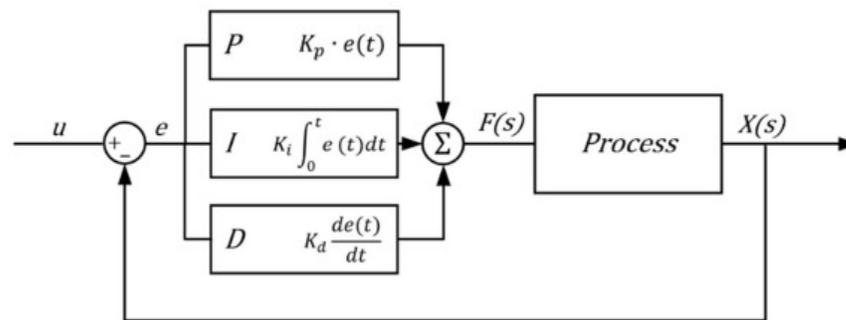
2.5 Kendali Proportional Integral Derivative (PID)

Pengendali PID adalah pengendali yang sering digunakan dibidang industri, robotika maupun *home automation*. Kendali PID merupakan suatu kendali yang sederhana dan dikategorikan sebagai kendali *Single Input single Output (SISO)*. Kendali merupakan kendali daur tertutup dan bekerja dengan cara mengoreksi keluaran suatu sistem dengan melakukan perhitungan selisih antara *set point* atau sinyal masukan dan keadaan sistem saat ini yang diumpan balik sehingga memperoleh nilai *error* kemudian nilai *error* dikalkulasi pada pengendali PID, sehingga mengeluarkan sinyal berupa usaha untuk mengurangi nilai *error* pada sistem tersebut. Kendali proportional memiliki output yang



besarnya sebanding dengan besarnya sinyal error. Output kontroler merupakan perkalian antara penguatan proporsional dengan sinyal error. Kendali integral memiliki karakteristik seperti sebuah operasi integral, output kontroler dipengaruhi oleh perubahan yang sebanding dengan perubahan nilai sinyal error. Output kontroler merupakan penjumlahan terus menerus dari perubahan sinyal error. Kendali derivatif memiliki sifat seperti suatu operasi differensial. Perubahan yang mendadak pada masukan kontroler mengakibatkan perubahan yang sangat besar dan cepat. Kontroler ini tidak akan menghasilkan output saat sinyal error konstan sehingga tidak akan mempengaruhi keadaan mantap (Saepulah, 2019).

Gabungan aksi kontrol proportional, integral, dan derivative yang terlihat dalam Gambar 9 mempunyai keunggulan dapat saling menutupi kekurangan dan kelebihan dari masing-masing kontroler.



Gambar 10 Blok Diagram Sistem Kendali PID

Persamaan kontroler PID ini dapat dinyatakan sebagai berikut:

$$x(t) = K_p \cdot e(t) + K_i \int e(t) dt + K_d \cdot de(t)/dt \quad (12)$$

dimana,

K_p = konstanta proportional

K_i = konstanta integral

K_d = konstanta derivative

$e(t)$ = sinyal error

$x(t)$ = output kendali

Untuk mendapatkan kendali PID yang memiliki performa baik dibutuhkan pengaturan gain atau konstanta yang tepat pada komponen *proportional* (K_p), *integral* (K_i), dan *derivative* (K_d) (Rachmat, 2022). Dalam pengaturan konstanta



PID ada beberapa hal yang perlu diperhatikan dari *respon* keluaran sistem yang dikendalikan, berikut beberapa parameter yang perlu diperhatikan.

1. *Rise time*, merupakan waktu yang dibutuhkan sistem untuk mencapai dari kondisi awal menuju nilai *setpoint*.
2. *Overshoot*, merupakan nilai maksimum dari *respon* sistem ketika menuju stabil pada *setpoint*.
3. *Settling time*, didefinisikan sebagai waktu yang dibutuhkan sistem hingga mencapai kondisi *steady state*.
4. *Steady-state error*; merupakan selisih antara keluaran sistem saat kondisi *steady state* dengan *setpoint*.
5. *Stability*, merupakan *respon* suatu sistem untuk tetap bertahan pada posisi *steady-state* ketika mencapai *setpoint*.
6. *Steady state*

2.7 Penelitian Terkait

Tabel 1 Penelitian Terkait

Deskripsi penelitian	Pembahasan
----------------------	------------



<p>Judul : Ackerman Unmanned Mobile Vehicle Based on Heterogeneous Sensor in Navigation Control Application</p> <p>Tahun : 2023</p> <p>Penulis : Chi-Huang Shih Cheng-Jian Lin Jyun-Yu Jhang</p>	<p>Pada paper ini menyajikan tentang implementasi teknik odometry untuk kontrol navigasi robot. Robot yang dirancang berupa tipe Ackerman yang modelnya berupa tricycle steering yakni terdapat dua roda seperti penggerak differensial dan satu roda depan yang difungsikan untuk steering atau manuver robot.</p>
<p>Judul : Localization of the Closed-Loop Differential Drive Mobile Robot Using Wheel Odometry</p> <p>Tahun : 2023</p> <p>Penulis : Gurpeet Singh Vijay Kumar</p>	<p>Paper ini membahas tentang robot dengan penggerak diferensial beroda dua dirancang dan dikembangkan untuk lingkungan bebas hambatan. Kedua motor dikendalikan secara individual untuk mencapai lokasi yang diinginkan menggunakan inverse kinematic. Keadaan posisi relative robot diperkirakan dengan menerapkan forward kinematic setelah mendapat umpan balik dari encoder magnetik.</p>
Deskripsi penelitian	Pembahasan
<p>Judul :</p>	<p>Pada paper ini membahas tentang omni-</p>



PID Control Schematic Design for directional mobile robot (OMR) yang Omni-directional Wheel Mobile merupakan holonomic robot dapat Robot Cilacap State of Polytechnic bergerak ke segala arah dengan menggunakan roda omni. Penelitian ini

Tahun :

2022

Penulis :

Hendi Purnata

Syahrul Ramadan

Muhammad Arif Hidayat

Irvan Maulana

menyajikan metode pengontrol PID pada plant OMR dengan tujuan tertentu untuk mencapai titik yang ditentukan. OMR menggunakan tiga roda dengan perbedaan sudut 120 derajat. Pada tahapan penelitian ini dilakukan pemodelan melalui bantuan MATLAB penentuan forward kinematic dan inverse kinematic, penentuan pengontrol PID untuk penentuan posisi serta perancangan dan pengujian robot dengan trajectory planning yang sudah ditentukan oleh peneliti.

www.pajestis.com
Optimal
Print

