REPUBLIK INDONESIA
KEMENTERIAN HUKUM DAN HAK ASASI MANUSIA

SURAT PENCATATAN
CIPTAAN

Dalam rangka pelindungan ciptaan di bidang ilmu pengetahuan, seni dan sastra berdasarkan Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta, dengan ini menerangkan:

Nomor dan tanggal permohonan	:	EC002022112351, 24 Desember 2022
Pencipta		
Nama	:	**Afianto dan Mada Jimmy Fonda Arifianto**
Alamat	:	Jl. Yaktapena Raya K8/A21 (Komplek Pertamina), Kel. Pondok Ranji, Kec. Ciputat Timur, Tangerang Selatan, BANTEN, 15412
Kewarganegaraan	:	Indonesia
Pemegang Hak Cipta		
Nama	:	**Politeknik Astra**
Alamat	:	Jl. Gaya Motor Raya No 8 Sunter II Tanjung Priok, Jakarta Utara, JAKARTA Utara, DKI JAKARTA, 14330
Kewarganegaraan	:	Indonesia
Jenis Ciptaan	:	**Diktat**
Judul Ciptaan	:	**TRACKLESS AUTOMATED GUIDED VEHICLE (AGV) DENGAN NAVIGASI RTLP-UWB DAN KENDALI MOTOR SWERVE DRIVE**
Tanggal dan tempat diumumkan untuk pertama kali di wilayah Indonesia atau di luar wilayah Indonesia	:	24 Desember 2022, di Jakarta
Jangka waktu pelindungan	:	Berlaku selama 50 (lima puluh) tahun sejak Ciptaan tersebut pertama kali dilakukan Pengumuman.
Nomor pencatatan	:	000428095

adalah benar berdasarkan keterangan yang diberikan oleh Pemohon.
Surat Pencatatan Hak Cipta atau produk Hak terkait ini sesuai dengan Pasal 72 Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta.

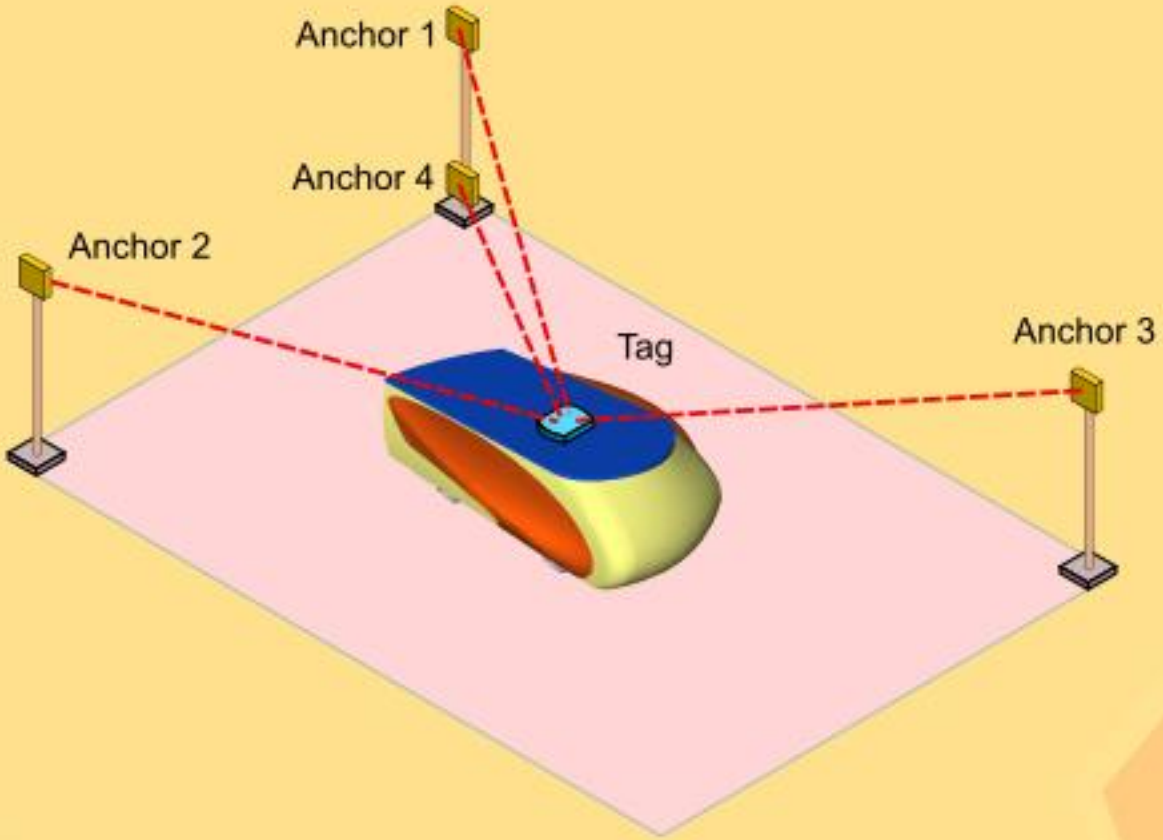
a.n Menteri Hukum dan Hak Asasi Manusia
Direktur Jenderal Kekayaan Intelektual
u.b.
Direktur Hak Cipta dan Desain Industri

Anggoro Dasananto
NIP.196412081991031002

Disclaimer:
Dalam hal pemohon memberikan keterangan tidak sesuai dengan surat pernyataan, Menteri berwenang untuk mencabut surat pencatatan permohonan.

BUKU AJAR

TRACKLESS AUTOMATED GUIDED VEHICLE (AGV) dengan Navigasi RTLS-UWB dan Kendali Motor *Swerve Drive*



Afianto, S.T., M.T., M.Sc.
Mada Jimmy Fonda Arifianto, S.T., M.Sc.

Jakarta, Desember 2022

POLITEKNIK ASTRA

Buku Ajar

TRACKLESS AUTOMATED GUIDED VEHICLE (AGV) DENGAN NAVIGASI RTLP-UWB DAN KENDALI MOTOR *SWERVE DRIVE*

oleh

Afianto, S.T., M.T., M.Sc.
Mada Jimmy Fonda Arifianto, S.T., M.Sc.

Jakarta, Desember 2022

POLITEKNIK ASTRA

Buku Ajar

"Trackless Automated Guided Vehicle (AGV) dengan Navigasi RTLS-UWB dan Kendali Motor Swerve Drive"

Penulis:
Afianto, S.T., M.T., M.Sc.
Mada Jimmy Fonda Arifianto, S.T., M.Sc.

ISBN:

Editor:
Dr. Syahril Ardi, S.T., M.T.

Penerbit:
LP2M Politeknik Astra
Jl. Gaya Motor Raya no.8 Sunter II Jakarta Utara
Jakarta 14330 Indonesia
Telp/Fax: +62-21 651 9555/+62-21 651 9821
Email: *lppm@polman.astra.ac.id*
Website: *https://lppm.polman.astra.ac.id*

PRAKATA

Dengan rahmat Allah SWT yang telah diberikan kepada kami sehingga penulis dapat menyelesaikan tulisan Buku Ajar yang berjudul "***Kendali Trackless Automated Guided Vehicle (AGV)***". Maksud dan tujuan dari penulisan Buku Ajar ini adalah untuk menambah wawasan dan pengetahuan tentang pengendalian AGV jenis *Trackless* bagi para pembaca dan juga penulis, yang selanjutnya bisa diaplikasikan untuk kepentingan masyarakan umum.

Proses *material handling* di industri, perlu melakukan pemilihan yang tepat guna mendapatkan peningkatan produktivitas. Dengan adanya berbagai pilihan baik model manual maupun otomatis, maka dengan tujuan untuk meningkatkan produktivitas dipilihlah secara otomatisasi guna mengurangi biaya produksi. Proses pendistribusian otomatis tersebut salah satunya adalah dengan menggunakan mobile robot model *Trackless Automatic Guided Vehicle* (AGV) yaitu pergerakan AGV yang melewati rute tanpa adanya jalur fisik untuk menuju dari satu point ke point lainnya. Mobile Robot *Trackless* AGV mempunyai navigasi dengan sistem Local Positioning System – Ultra Wideband (*Indoor* GPS) adapun untuk pergerakan dan kemudi dibuat dalam satu unit modul pada tiap rodanya yaitu menggunakan jenis *Swerve Drive Motor*.

Buku ajar ini akan memberikan wawasan dan pengetahuan tentang model penggerak AGV, model navigasi tanpa jalur fisik dan sistim pengendali motor DC jenis *Swerve Drive*. Dan yang menjadi fokus penulis adalah bagaimana model navigasi dan pengendali motor DC menggunakan sistim Proposional Integral Derivatif (PID) pada sebuah AGV tanpa menggunakan jalur fisik atau biasa disebut *Trackless* AGV.

Tujuan akhir dari penulisan buku ajar ini tentunya adalah *untuk* memberikan dampak positif yang banyak bagi penggunannya dalam proses pendistribusian material/barang menggunakan model *Trackless* AGV. Beberapa manfaat dari AGV jenis Trackless ini adalah sebagai berikut : waktu proses *material handling* akan lebih efektif, tepat dan efisien, memiliki nilai estetika pada area kerja yang lebih baik, kemudahan dalam pengaturan rute menjadi flesibel, pengupahan karyawan berkurang, menghilangkan kelalaian akibat *human error* dan angka kecelakaan kerja dapat diminimalisasi, serta mengurangi biaya perawatan khususnya pada jalur lintasan.

Buku Ajar ini tersusun tentunya hasil dari dukungan beberapa pihak. Oleh karena itu, penulis mengucapkan terima kasih kepada pihak-pihak yang telah berkontribusi baik langsung

maupun tidak langsun. Penulis sangat menyadari bahwa Buku Ajar yang disusun ini masih jauh dari kesempurnaan, untuk itu kritik dan saran yang membangun sangat kami harapkan. Semoga Buku Ajar ini dapat memberikan bermanfaat bagi kita semua. Terimakasih.

Jakarta, 20 Desember 2022

Penulis
Afianto, S.T., M.T., M.Sc.
Mada Jimmy Fonda Arifianto, S.T., M.Sc.

DAFTAR ISI

Daftar Gambar

Bab 1. PENDAHULUAN

1.1. Latar Belakang

Era industri 4.0 memicu pengaplikasikan teknologi yang mengarah pada otomatisasi dalam segala hal, industri-industri saat ini mulai memanfaatkan teknologi robotika untuk membantu tugas-tugas manusia, hal ini ditujukan untuk meningkatkan produktifitas serta kualitas produk. Tidak terkecuali salah satu proses yang sangat penting dalam rantai produksi yaitu distribusi barang produksi yaitu pada proses pemindahan barang dari suatu tempat ke tempat lain. Model transportasi menggunakan kemudi yang dilakukan oleh manusia akan segera ditinggalkan, hal ini didorong oleh beberapa penemuan teknologi maju yang dapat mengurangi bahkan sampai menghilangkan beberapa resiko pada manusia maupun material. Hal – hal yang bisa menjadi masalah dengan model kemudi manusia adalah kelalaian manusia sehingga terjadi kesalahan lokasi baik asal maupun tujuan pengiriman atau sampai terjadinya kecelakaan, belum lagi tentang biaya tenaga kerja yang semakin tinggi. Untuk itu, diperlukan otomatisasi pada sarana-sarana pendistribusian tersebut, salah satu penerapan otomatisasi pada sarana pendistribusian barang yaitu *Automatic Guided Vehicle* (AGV). *AGV* merupakan suatu alat transportasi untuk memindahkan suatu barang/material, dikendalikan secara otomatis menggunakan navigasi yang nantinya dapat bergerak mengikuti jalur yang telah ditentukan hingga sampai tujuan. Model transportasi otomatis untuk distribusi barang/material tersebut bisa dilakukan secara cepat dan efisien. Dengan melihat manfaat yang begitu banyak seperti efisiensinya tinggi, handal, ketahanan yang kuat, fleksibel, dan akurasi gerak yang tinggi, inilah yang membuat *AGV* banyak digunakan di industri dalam proses manufaktur, perakitan dan logistic, rumah makan hingga rumah sakit. Ada hal lain yang sebenarnya harus dihindari adalah pemanfaatan *AGV line follower* yang menggunakan panduan garis atau sejenisnya, sebab akan mengganggu keindahan, kebersihan dan kerapian lingkungan. Hal tersebut tidak disarankan untuk area perkantoran, pusat perbelanjaan, rumah sakit dan lingkungan kampus.

1.2. Tujuan dan Manfaat

Alat transportasi otomatis AGV ini memiliki tujuan utama adalah untuk meningkatkan produktifitas dengan memperhatikan faktor lingkungan global, yaitu untuk mengurangi pencemaran lingkungan dengan mengurangi emisi gas buang. Adapun manfaat dari pengaplikasian AGV tentu dapat menurunkan biaya operator dan faktor kecelakaan kerja.

Pengurangan biaya transportasi secara langsung akan berdampak pada penurunan biaya produksi.

1.3. Tinjauan Pustaka

1.3.1. Automated Guided Vehicle (AGV)

Alat transportasi AGV berjalan dengan panduan/navigasi untuk pemindahan atau pendistribusian barang dari satu tempat ke tempat yang lainnya. Alat pemandu AGV bisa berupa jalur fisik (misalnya ditandai di lantai dengan pita magnetik atau label), AGV jenis ini disebut AGV *Line Folower*. Ada juga jenis pemandu virtual, jadi pemandu berada pada program di dalam AGV, jenis ini bisa dengan teknologi laser (*laser guided*) bahkan yang sekarang sedang dikembangkan menggunakan LPS (*local positioning system*) semacam GPS indoor yang hanya untuk daerah ruangan tertentu saja. Adapun penggunaan AGV dalam menjalankan usahannya dapat memberikan beberapa dampak keuntungan:

a) **Pengendalian biaya,** biaya sistem AGV sangat dapat diprediksi, sementara biaya tenaga kerja cenderung meningkat dan dapat berubah dengan cepat tergantung pada kondisi ekonomi lokal atau pada saat permintaan produk meningkat.

b) Handal, keandalan dan pengiriman tepat waktu yang dilakukan oleh AGV meningkatkan efisiensi operasi. Kemampuan untuk beroperasi dengan waktu yang non-stop namun tetap menjaga konsistensi dalam operasi.

c) **Keamanan**, AGV selalu mengikuti jalur panduan dan akan berhenti jika mengalami adanya halangan, hal ini dapat meningkatkan keselamatan pekerja di sekitarnya. Pada indutri manufaktur sering kali alat transportasi difungsikan untuk membawa barang yang besar dan berat, hal ini ada kalanya potensi *human error* sehingga muncul bahaya cidera atau kecelakaan kerja terhadap pekerja. Dengan pengendalian gerakan pada AGV yang diatur seaman mungkin maka dapat mengurangi kecelakaan kerja serta kerusakan produk yang dibawa. Serta kehilangan barang atau kesalahan penempatan barang dapat dilacak dengan cepat oleh sistem.

d) **Kemudahan,** bebas jelajah. Adanya roller atau belt conveyor cukup membantu produktifitas tapi tidak mampu mengatasi permasalahan saat kebutuhan yang komplek dan tuntutan fleksibilitas operasional meningkat. AGV menghilangkan masalah akses yang dibuat oleh konveyor dan membutuhkan lebih sedikit ruang daripada forklift konvensional, serta memungkinkan untuk akses ke lorong yang lebih sempit

e) **Fleksibel,** AGV pada logistik gudang juga harus fleksibel terhadap variable yang ada seperti : Jalur navigasi dapat diubah karena kebutuhan produksi dan layout yang

berkembang bahkan teknologi sekarang AGV mampu melakukan pergerakan dengan memberikan program secara *teaching*, mampu dikombinasikan dengan konveyor, tersedia sistem yang dapat dikolaborasi dengan perangkat lain seperti robot lengan.

f) **Efisien,** AGV harus mampu bekerja efisien terhadap waktu operasionalnya terutama dalam proses alur transfer barang. Kemampuan yang diperlukan seperti pendeteksian rute terpendek secara akuran, perencanaan rute alternatif saat terdapat hambatan, pemetaan posisi barang-barang agar mudah ditelusuri hingga penambahan informasi layout gudang untuk infomasi titik penjemputan dan pengiriman.

g) **Pengurangan biaya operasi** - Pengisian dan penanganan baterai bisa otomatis dengan sistem AGV dan akselerasi / perlambatan pergerakan yang dikontrol ini meminimalkan keausan pada komponen.

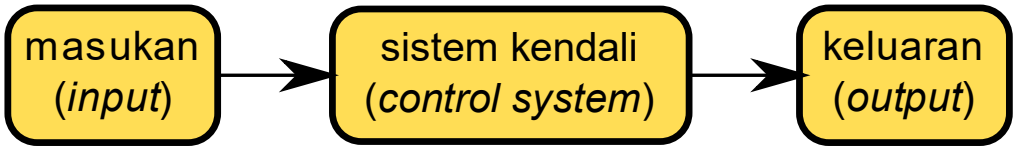
h) **Repeatability,** AGV dapat diprediksi dan andal dalam melakukan tugas gerakan yang berulang – ulang.

i) **Terorganisir,** Pada skala gudang yang besar, tentu akan diperlukan lebih dari 1 AGV dan banyak fasilitas yang harus dilayani. Maka robot harus dirancang untuk dapat bekerja sama dalam 1 armada terpusat. Dengan kendali armada terpusat maka dapat diintegerasikan dengan perangkat lunak pengelolaan pabrik atau gudang seperti MES atau WMS, sehingga kontrol lalu lintas dan rute dapat direncanakan secara optimal.

j) **Skalabilitas,** Lebih banyak AGV dapat ditambahkan untuk memperluas kapasitas dan *throughput*.

1.3.2. Sistim Kendali

Sistem kendali/kontrol didefinisikan suatu alat atau susunan alat yang terkait sedemikian rupa sehinga dapat memerintah, mengarahkan, atau mengatur keadaan dari suatu sistem diri sendiri atau sistem lain. Sistem kendali tersusun atas sub-sistem dan proses (*plants*) yang diberikan *input* sebagai kinerja yang diinginkan untuk mendapatkan keluaran (*output*). Gambar di bawah ini menununjukkan blok diagram untuk sistem kendali paling sederhana.



Gambar 1 Diagram blok sistem kendali

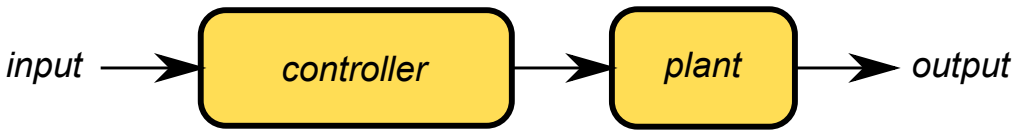
Pada Sistem Kendali, dapat diklasifikasikan pada beberapa hal:

1. Sistem Kendali Manual dan Otomatik,
2. Sistem Lingkar Terbuka (*Open Loop*) dan Lingkar Tertutup (*Closed Loop*),
3. Sistem Kendali Kontinu dan Diskrit.

Adapun untuk sumber penggerak dapat berupa sistem aktuator Elektrik, Mekanik, Pneumatik, dan Hidraulik.

Sistem Kendali Manual dilakukan oleh manusia yang bertindak sebagai operator, sedangkan Sistem Kendali Otomatik dilakukan oleh peralatan yang bekerja secara otomatis dan operasinya dibawah pengawasan manusia. Ada dua jenis system kendali yaitu Sistem Kendali Lingkar Terbuka (*Open Loop*) dan Sistem Kendali Lingkar Tertutup (*Closed Loop*), dan bisa dijelaskan secara singkat sebagai berikut :

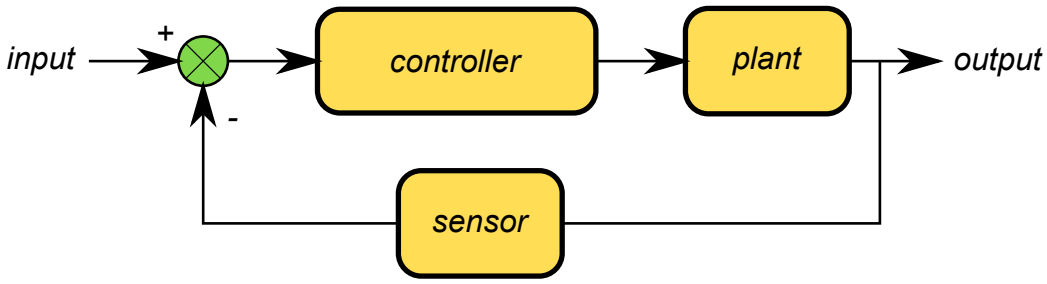
a. Sistem Kendali Lingkar Terbuka (*Open Loop*), mengendalikan suatu proses dimana besaran keluaran tidak digunakan sebagai umpan balik terhadap besaran masukan, sehingga variable yang dikendalikan pada keluaran tidak dapat dibandingkan terhadap nilai/variable yang diinginkan pada masukan.



Gambar 2 Sistem kontrol loop terbuka

Pada Sistem Kendali Loop Terbuka memiliki beberapa keuntungan antara lain, sederhana, harganya murah, dapat dipercaya. Namun sistem kendali ini memiliki kekurangan yaitu kurang akurat karena tidak terdapat koreksi terhadap kesalahan pada keluarannya. Dengan adanya gangguan, sistem control terbuka tidak dapat melaksanakan tugas yang sesuai diharapkan. Sistem kontrol terbuka dapat digunakan hanya jika hubungan antara masukan dan keluaran diketahui dan tidak terdapat gangguan internal maupun eksternal.

b. Sistem Kendali Lingkar Tertutup (*Closed Loop*) mengendalikan suatu proses dengan memanfaatkan besaran keluaran untuk digunakan sebagai umpan balik pada besaran masukan, sehingga variable yang dikendalikan dapat dibandingkan terhadap nilai/variable yang diinginkan. Perbedaan nilai yang terjadi antara besaran keluaran dengan besaran masukan ini digunakan sebagai koreksi yang merupakan sasaran pengendalian suatu proses.



Gambar 3 Sistem kontrol loop tertutup

Sinyal kesalahan penggerak, yang merupakan selisih antara sinyal masukan dan sinyal umpan balik, diumpankan ke kontroler untuk memperkecil kesalahan dan membuat agar keluaran sistem mendekati harga yang diinginkan.

Berikut ini adalah bagian-bagian pada sistem kendali tertutup:

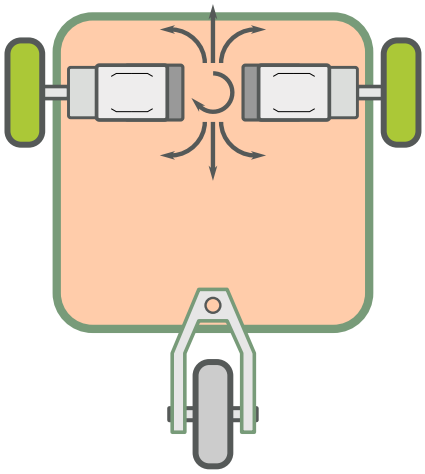
1. Masukan (*Input*), merupakan stimulus yang diberikan pada sistem kontrol, merupakan nilai yang diinginkan bagi variabel yang dikontrol selama pengontrolan. Harga ini tidak tergantung pada keluaran sistem
2. Keluaran (*Output*), merupakan tanggapan pada sistem kontrol, merupakan harga yang akan dipertahankan bagi variabel yang dikontrol, dan merupakan harga yang ditunjukan oleh alat pencatat
3. Beban (*Plant*) merupakan sistem fisis yang akan dikontrol (misalnya mekanis, elektris, hidraulik ataupun pneumatic).
4. Alat Kontrol/Kendali (*Controller*), merupakan peralatan/ rangkaian untuk mengontrol beban (sistem). Alat ini bisa digabung dengan penguat
5. Elemen Umpan Balik (*feedback*), menunjukan/mengembalikan hasil pencatan ke detector sehingga bisa dibandingkan terhadap harga yang diinginkan (di stel)
6. alat deteksi kesalahan (*Error Detector*), merupakan alat pendeteksi kesalahan yang menunjukan selisih antara masukan (*input*) dan respons melalui umpan balik (*feedback path*).
7. Gangguan merupakan sinyal-sinyal tambahan yang tidak diinginkan. Gangguan ini cenderung mengakibatkan nilai keluaran berbeda dengan harga masukanya, gangguan ini biasanya disebabkan oleh perubahan beban sistem, bisa juga karena adanya perubahan kondisi lingkungan, getaran ataupun yang lain.

Bab 2. Model Penggerak AGV

Sistem pergerakan pada AGV dilakukan oleh roda penggerak yang mempunya susunan bisa 2, 3 atau 4 roda penggerak, demikian juga sebagai sistem kemudi bisa dilakukan oleh 2, 3 atau bahkan ke 4 rodanya. Berikut ini akan dijelaskan beberapa jenis sistem penggerak dan kemudi pada AGV :

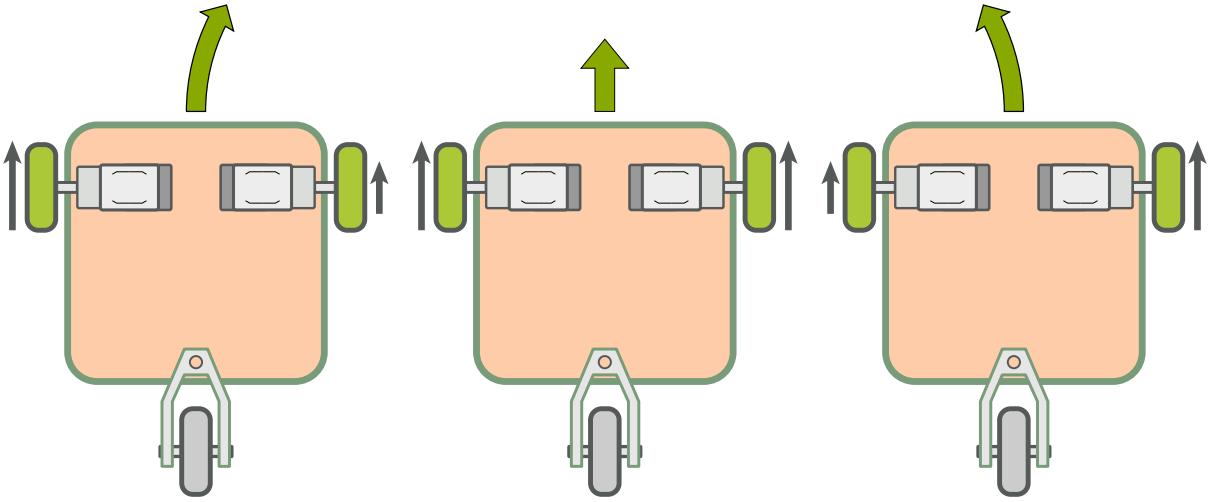
2.1. Model Two Differential Drives as Steering

Model dengan 2 buah motor penggerak yang secara perbedaan kecepatan mengatur arah gerak dan belok AGV sehingga fungsi roda penggerak juga menjadi steering arah pergerakan AGV. AGV model ini paling banyak diaplikasikan sebab disamping lebih mudah dalam pengendaliannya juga lebih murah dalam membuatnya. Berikut adalah contoh desain AGV model *Two differential Driver as Steering*.



Gambar 4 Model AGV: dua penggerak diferensial sebagai kemudi

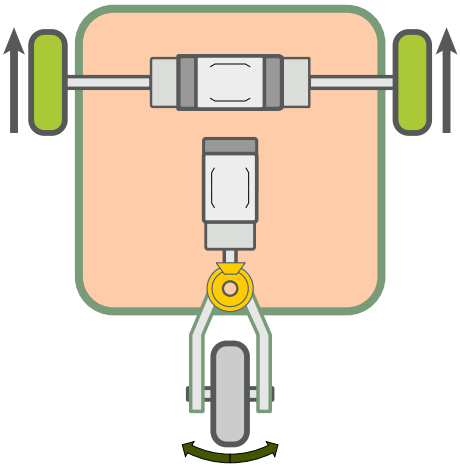
Untuk berbelok ke kiri, putaran roda kanan dibuat lebih cepat. Sebaliknya untuk berbelok ke kanan, putaran roda kiri dibuat lebih cepat. Jenis AGV ini membutuhkan 2 buah roda Casters sebagai penggerak yang terpasang pada Motor DC, serta 1 buah roda Caster bantuan sebagai roda penyangga untuk membantu agar AGV dapat berdiri sejajar dan dapat berbelok. AGV jenis ini dibuat untuk dapat bergerak 2 arah (maju dan mundur). Berikut digambarkan bagaimana kondisi arah putaran dan kecepatan dari motor agar AGV bisa melakukan gerak maju, belok kanan, dan belok kiri. Adapun untuk pergerakan mundur maka 2 motor penggerak diputar dengan arah terbalik dari arah maju dengan kecepatan yang sama.



Gambar 5 Kendali AGV dua penggerak diferensial sebagai kemudi

2.2. Model Trycycle Wheels Drive As Steering (One Drive And One Steering)

AGV jenis ***dapat bergerak secara*** Gesit dengan radius putar kecil. AGV ini menggunakan metode memisahkan antara roda yang digunakan sebagai penggerak (drive), dan roda yang digunakan sebagai pengemudi (steering). AGV jenis ini biasanya dibuat untuk bergerak 1 arah saja (maju). Gambar dibawah ini memberikan gambaran bagaimana kondisi arah putaran dan kecepatan dari motor agar AGV bisa melakukan gerak maju, belok kanan, dan belok kiri. Adapun untuk
pergerakan mundur untuk AGV model ini tidak disarankan sebab akan mengendalikan AGV dengan cukup rumit.

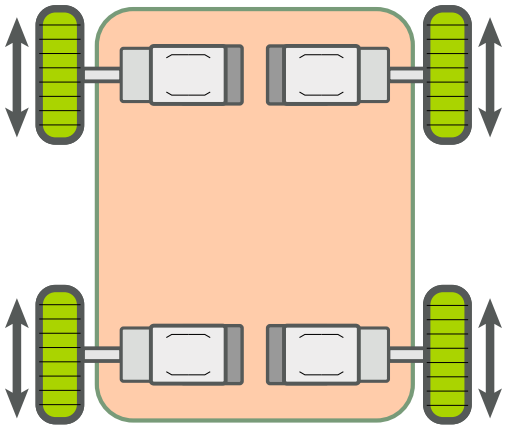


Gambar 6 Model AGV: satu penggerak dan satu kemudi

2.3. Model Four Differential Wheels Drive As Steering

Pada prinsipnya AGV jenis ini menggunakan 4 buah roda dengan motor penggerak, yang secara diferensial dapat mengatur arah belok AGV. AGV jenis ini dibuat untuk dapat bergerak bergerak 2 arah (maju dan mundur), dan berbelok ke kanan dan ke kiri dengan mengatur

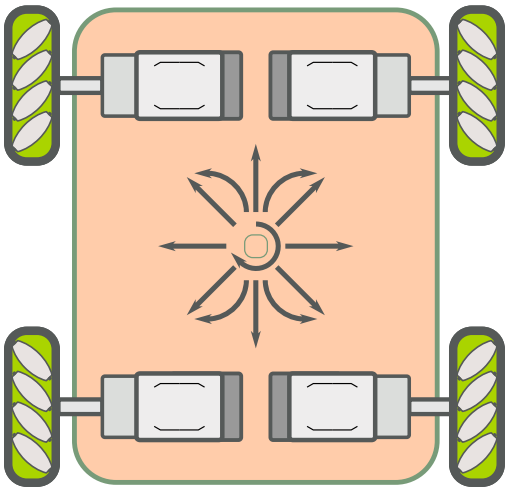
kecepatan dari 2 roda sisi kanan dan 2 roda sisi kiri yang berbeda. AGV model ini biasanya digunakan untuk dibebani, mendorong atau menarik beban yang besar sehingga membutuhkan torsi yang besar pula.



Gambar 7 Model AGV: Empat roda penggerak diferensial sebagai kemudi

2.4. Model Four Mecanum Wheels

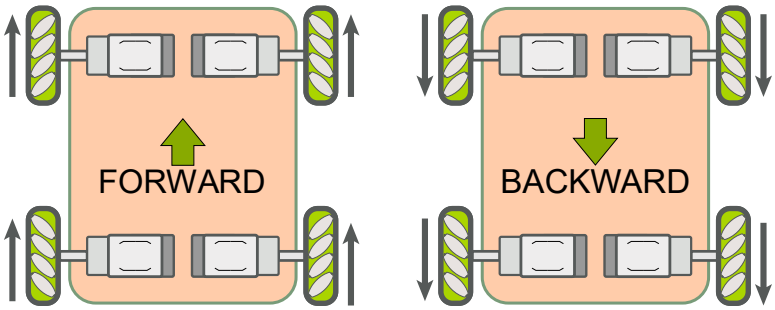
AGV ini adalah pengembangan dari AGV tipe ke-3 yang sudah dijelaskan diatas. Namun menggunakan *mecanum wheels* sebagai roda penggeraknya, sehingga AGV dapat bergerak bebas dan memungkinkan gerakan berputar sampai 360 derajat. AGV tipe ini adalah tipe yang paling modern saat ini, dan tergolong omnidirection, yang artinya dapat bergerak kesegala arah (maju, mundur, kanan, kiri, serong kanan, serong kiri dan berputar), dan kelebihannya adalah jenis AGV yang sesuai untuk lingkungan yang jalurnya sempit.



Gambar 8 Model AGV: Empat roda mechanum

Berikut ini akan dijelaskan bagaimana AGV Model *Four Mecanum Wheel* melakukan 16 arah Gerakan.

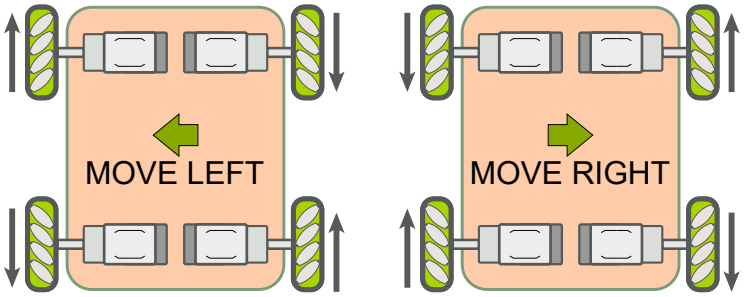
1. Gerak Maju dan Mundur



Gambar 9 Gerak maju dan mundur AGV mechanum

AGV akan bergerak maju atau mundur, dengan cara ke 4 roda mecanum berputar searah dan berkecepatan yang sama (arah dan kecepatan tiap roda, ditunjukan oleh gambar panah di sisi roda).

2. Gerak ke Kanan dan ke Kiri

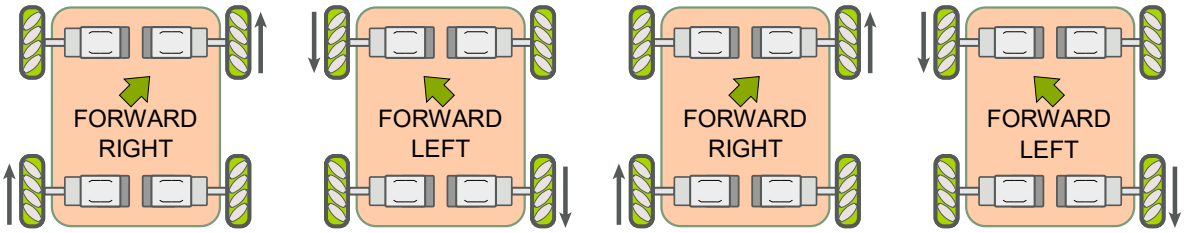


Gambar 10 Gerak kanan dan kiri AGV mechanum

AGV akan bergerak ke kanan dan ke kiri, dengan cara ke 2 roda mecanum yang berlisangan berputar searah jarum jam (CW) dan 2 roda mecanum yang bersilangan lainnya berputar berlawanan arah jarum jam (CCW) dengan kecepatan ke 4 roda yang sama.

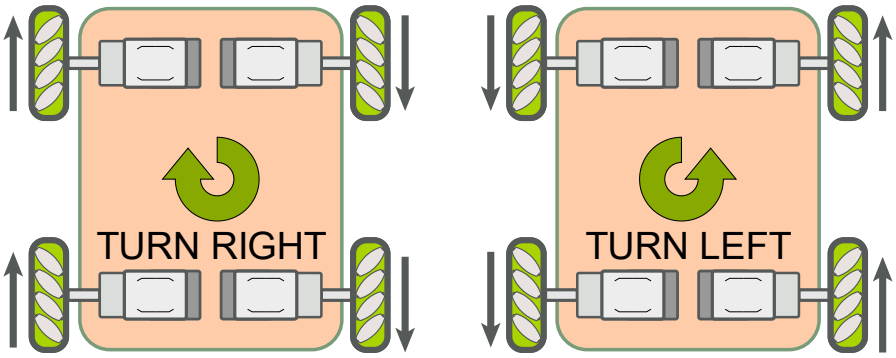
3. Gerak Serong Kanan dan Serong Kiri

AGV akan melakukan gerakan serong kanan dan kiri apabila ke 2 roda mecanum yang berlisangan berputar searah jarum jam (CW) dan 2 roda mecanum yang bersilangan lainnya tidak diputar (diam) dengan kecepatan ke 2 roda yang berputar adalah sama.



Gambar 11 Gerak serong kanan dan kiri AGV mechanum

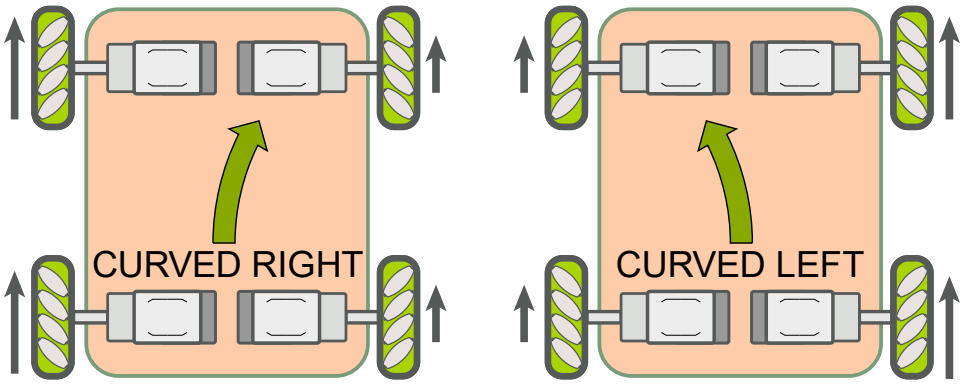
4. Gerak Putar Kanan dan Putar Kiri



Gambar 12 Gerak putar kanan dan putar kiri AGV mechanum

AGV akan melakukan gerakan putar kanan dan putar kiri apabila ke 2 roda mecanum yang disisi sama berputar searah jarum jam (CW) dan 2 roda mecanum di sisi sama lainnya berputar berlawanan arah jarum jam (CCW) dengan kecepatan ke 4 roda yang adalah sama.

5. Gerak Belok Kanan dan Belok Kiri

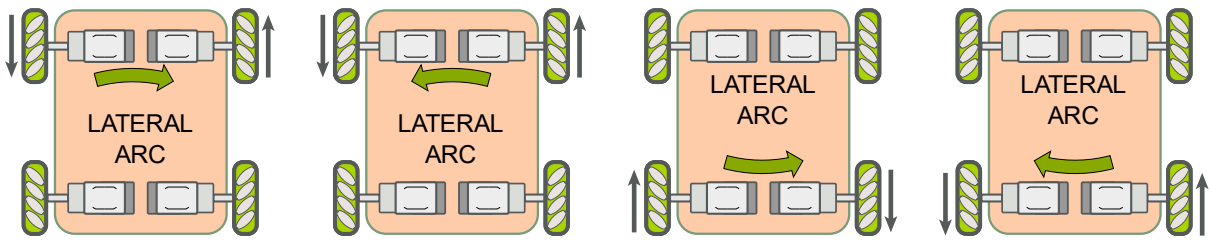


Gambar 13 Gerak belok kanan dan belok kiri AGV mechanum

AGV akan melakukan gerakan belok kanan dan belok kiri apabila ke 2 roda mecanum yang disisi sama berputar searah jarum jam (CW) dengan kecepatan sama besar dan 2 roda mecanum

disisi sama lainnya berputar searah jarum jam (CW) juga, namun dengan kecepatan yang lebih rendah dari 2 roda sisi lainnya.

6. Gerak Lateral Kanan dan Lateral Kiri

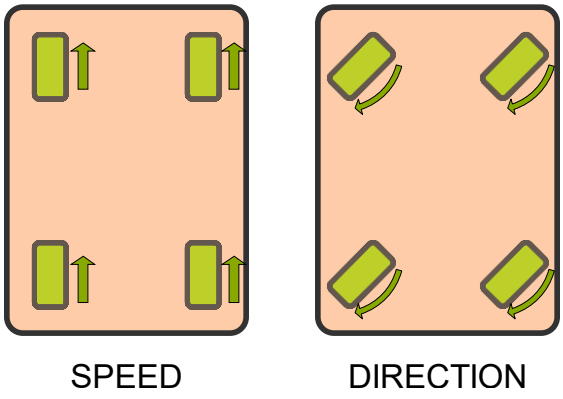


Gambar 14 Gerak lateral kanan dan kiri AGV mechanum

AGV akan melakukan Gerakan Lateral kanan atau kiri, bila 2 roda depan atau 2 roda belakan saja yang digerakan dengan arah yang saling berlawanan.

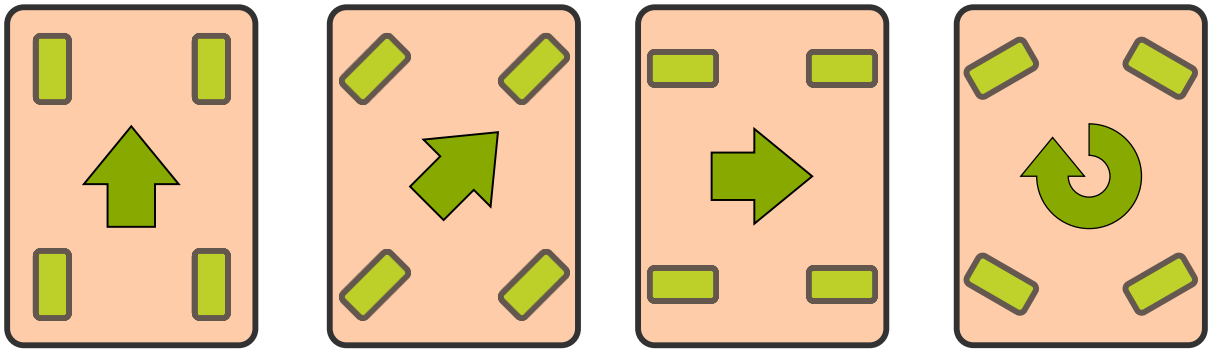
2.5. Model Swerve Drive Wheel

AGV yang digerakkan dan dikemudikan menggunakan Swerve Drive pada ke empat rodannya ini memungkinkan Gerakan AGV bisa lebih bebas lagi dalam pergerakannya, sebab Drive jenis ini kemudinya bisa bergerak 360 derajat dengan posisi di sudut 0 – 360 derajat. Tiap unit Drive ini memiliki 2 buah motor, dimana 1 motor digunakan untuk pergerakan putaran roda dan satu motor lainnya digunakan sebagai kemudi untuk arah pergerakan AGV. Swerve drive jauh lebih kompleks dan jauh lebih mahal daripada drive Mecanum. Namun, drive ini memiliki satu keunggulan utama dibandingkan penggerak Mecanum yaitu memiliki traksi yang lebih besar sebab kemudi dilakukan oleh motor yang berbeda, sehingga memiliki daya gerak yang lebih besar dan sesuai untuk mengangkut beban besar.



Gambar 15 Pengaturan motor penggerak dan motor kemudi dengan Swerve Drive

Sebuah AGV dengan swerve drive membutuhkan 4 motor untuk menggerakkan roda penggerak (satu untuk setiap roda), 4 encoder untuk melacak jarak yang ditempuh pada setiap roda, 4 motor untuk mengontrol sudut roda (satu untuk setiap roda), dan 4 encoder yang melacak sudut yang dihadapi setiap roda, hal inilah yang membuat AGV drive model ini mahal.

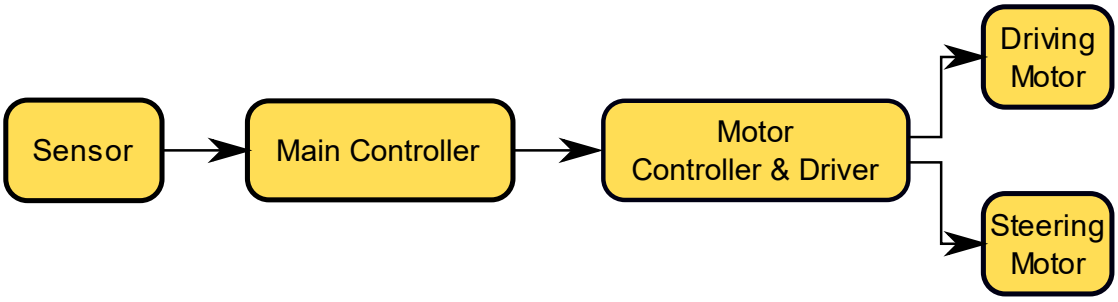


Gambar 16 Jenis pergerakan AGV dengan Swerve Drive

Bab 3. Metode Kendali PID

AGV merupakan suatu kendaraan yang dikendalikan secara manual dan otomatis menggunakan sistem navigasi dengan pengendalian pola gerakan menuju tempat yang dituju. Sistem navigasi menjadi bagian terpenting agar *AGV* dapat bergerak secara mandiri. Berikut ini adalah tipe berdasarkan navigasinya :

1. Tipe *Rail*, yaitu tipe navigasi dengan memberikan rel disepanjang jalur yang akan dilewati AGV.
2. Tipe *Wired*, yaitu tipe navigasi yang menggunakan kabel yang di tanam di lantai kemudian AGV dibekali sensor yang mendeteksi medan elektromagnet dari kabel.
3. Tipe *Line Tape*, yaitu tipe navigasi dengan menggunakan garis yang dibuat dengan membedakan warna terhadap lantai (optical) atau *magnetic tape*.
4. Tipe *Laser*, yaitu tipe navigasi dengan menggunakan laser transmiter pada AGV dan sejumlah laser reflector yang ditempatkan di titik – titik tertentu. Pantulan dari titik – titik tersebut memberikan gambaran bagi AGV untuk memetakan lingkungan sekitarnya.
4. Tipe *Trackless LPS*, yaitu tipe navigasi dengan menggunakan teknologi Local Positioning System pada AGV.

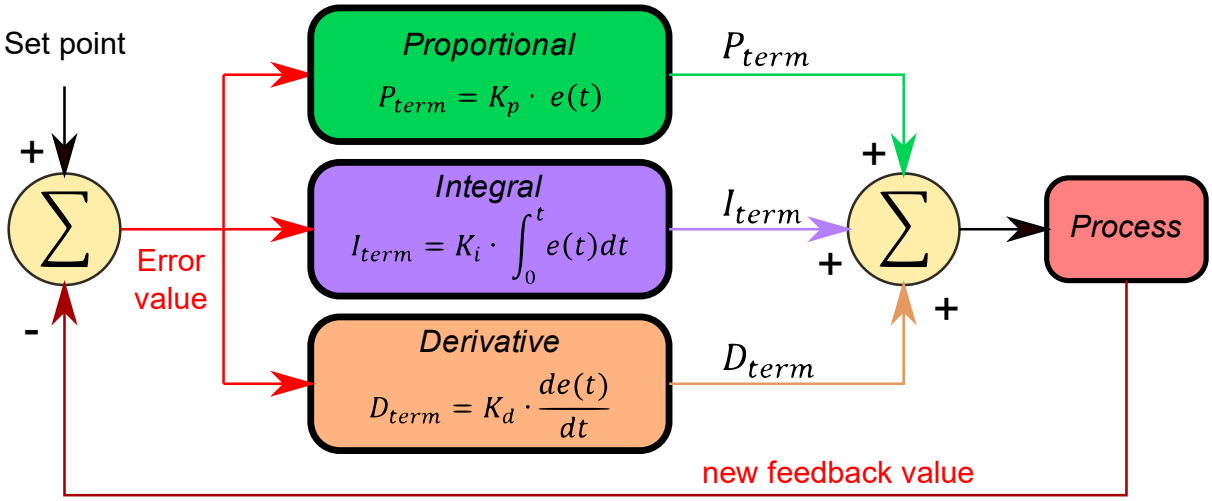


Gambar 17 Diagram blok kendali AGV

3.1. Konsep Kendali PID (Proportional-Integral–Derivative)

Pada AGV Trackless memiliki beberapa tingkatan level, yaitu level basic dan level professional atau advance, untuk yang basic yaitu AGV hanya mengandalkan sensor dan motor driver seperti yang sudah dijelaskan diatas, namun jika level advance atau professional tidak hanya mengandalkan pembacaan sensor dan motor drive namun juga membutuhkan rumus agar AGV dapat berjalan stabil dan mulus saat di lintasan, agar lebih jelas berikut adalah penjelasan prinsip kerja masing-masing level dalam AGV Trackless. AGV Trackless yang advance level

yaitu AGV Trackless yang menggunakan PID kontrol sebagai pengontrol untuk jalannya AGV. PID adalah Proportional, Integral dan Derivative, secara sederhana maksud dari Proportional yaitu untuk memperbaiki respon transien, Integral digunakan untuk menghilangkan error steady state, dan Derivative digunakan untuk memberikan efek redaman, ketiga faktor tersebut sangatlah penting untuk terciptanya suatu sistem yang sempurna, baik untuk jalannya AGV saat lurus, mundur atau berbelok. Kombinasi yang bisa digunakan untuk sebagai sistem kontrol yaitu PI, PD dan PID, selain dari itu, tidak diperkenankan untuk digunakan karena akan menimbulkan efek yang tidak diinginkan.



Gambar 18 Blok Diagram Kendali PID

PID merupakan kontroler mekanisme umpan balik yang biasanya dipakai pada sistem kontrol industri. Sebuah kontroler PID secara kontinyu menghitung nilai kesalahan sebagai beda antara setpoint yang diinginkan dan variabel proses terukur pada *feedback value*. Kontroler mencoba untuk meminimalkan nilai kesalahan setiap waktu dengan penyetelan variabel kontrol ke nilai baru yang ditentukan oleh formula PID.

Definisi-definisi PID

Dalam Kontrol PID ada beberapa istilah-istilah yang digunakan. Berikut adalah istilah-istilah yang digunakan pada kendali Trackless AGV:

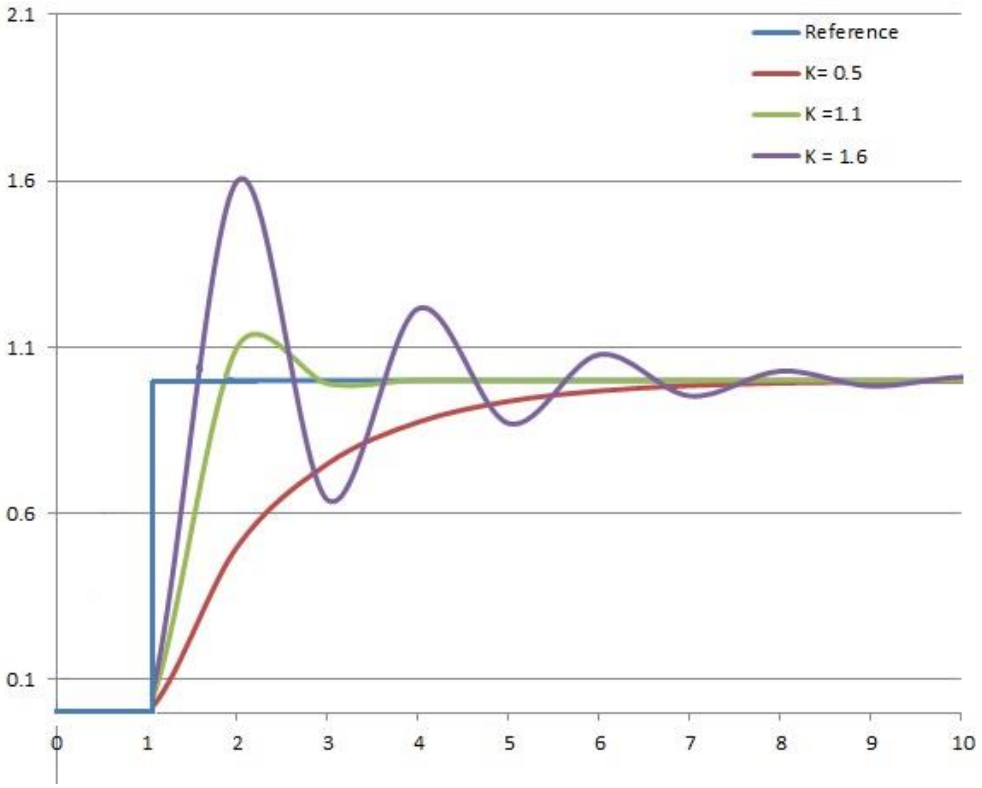
a) Target Position (*set-point*) adalah untuk mengikuti garis, posisi ini adalah garis tengah. Di mana akan merepresentasikannya dengan nilai nol.

b) Measured Position (*feedback value*) adalah seberapa jauh ke kiri atau ke kanan terhadap garis. Nilai ini dapat negatif atau positif untuk merepresentasikan posisi relatif terhadap garis.
c) Error adalah perbedaan antara target position dan measured position.
d) Proportional adalah mengukur berapa jauh AGV keluar dari garis. ProPortional merupakan dasar untuk membaca posisi AGV dengan menggunakan sensor. Semakin banyak sensor yang terpasang semakin banyak data, maka akan semakin akurat AGV dapat mengukur posisi AGV di atas garis.
e) Integral adalah mengukur akumulasi error terhadap waktu. Nilai integral naik ketika AGV tidak berada di tengah garis. Semakin lama AGV tidak berada di tengah garis, semakin tinggi nilai integral.
f) Derivative adalah mengukur seberapa sering AGV bergerak dari kiri ke kanan atau dari kanan ke kiri.
g) Faktor P = Kp, adalah konstanta yang digunakan untuk memperbesar dan memperkecil pengaruh dari Proportional.
h) Faktor I = Ki, adalah konstanta yang digunakan untuk memperbesar dan memperkecil pengaruh dari Integral.
i) Faktor D = Kd, adalah konstanta yang digunakan untuk memperbesar dan memperkecil pengaruh dari Derivative.

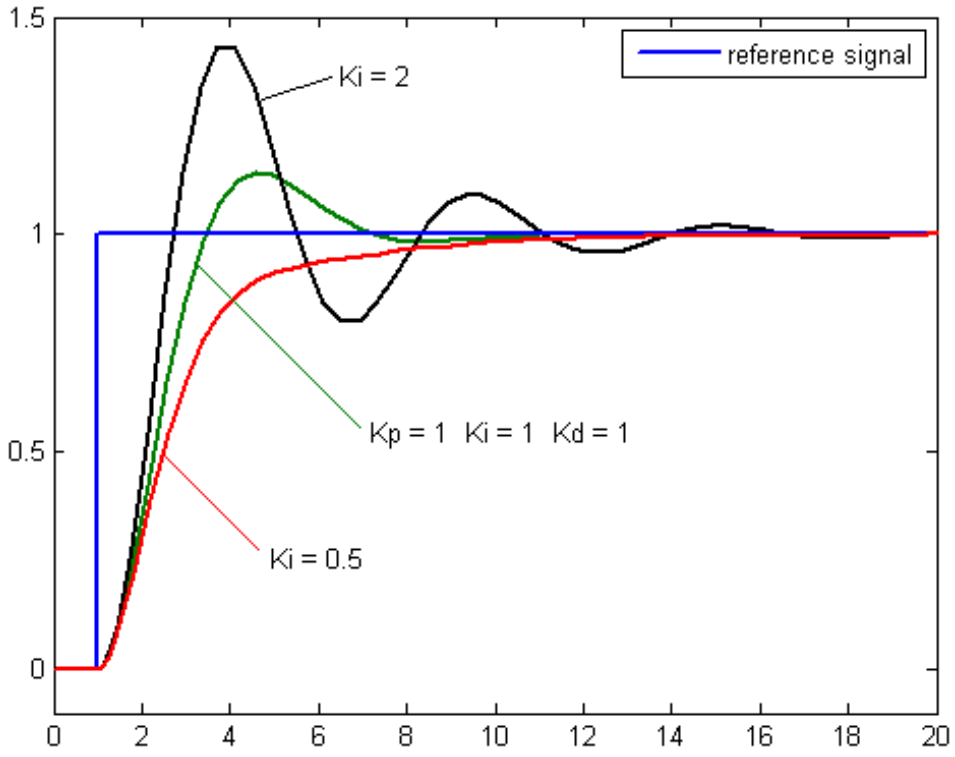
Keluaran kendali Proposional (P) memiliki hubungan yang proporsional (seimbang) dengan *error (deviasi*). Jika Kp di-set tinggi, tanggapannya cepat, tetapi jika terlalu tinggi sistem menjadi tidak stabil.

Kendali Integral (I) untuk mengoreksi keluaran dengan mengintegralkan error. Dalam kasus penyetelan (*adjustment*) kendali P, error yang besar akan menghasilkan penyetelan keluaran besar, jika error kecil penyetelan keluaran akan kecil juga. Namun error tidak dapat dibuat nol, kinerja integral mengkonpensasi masalah ini. Koreksi integral dilakukan dengan mengakumulasi error disetiap pembacaan PV, sehingga akhirnya deviasi nol. Tidak seperti kendali P, kendali I jarang digunakan sendirian melainkan dikombinasikan dengan dengan kendali Proporsional atau dengan kendali *Proposional Derivative*.

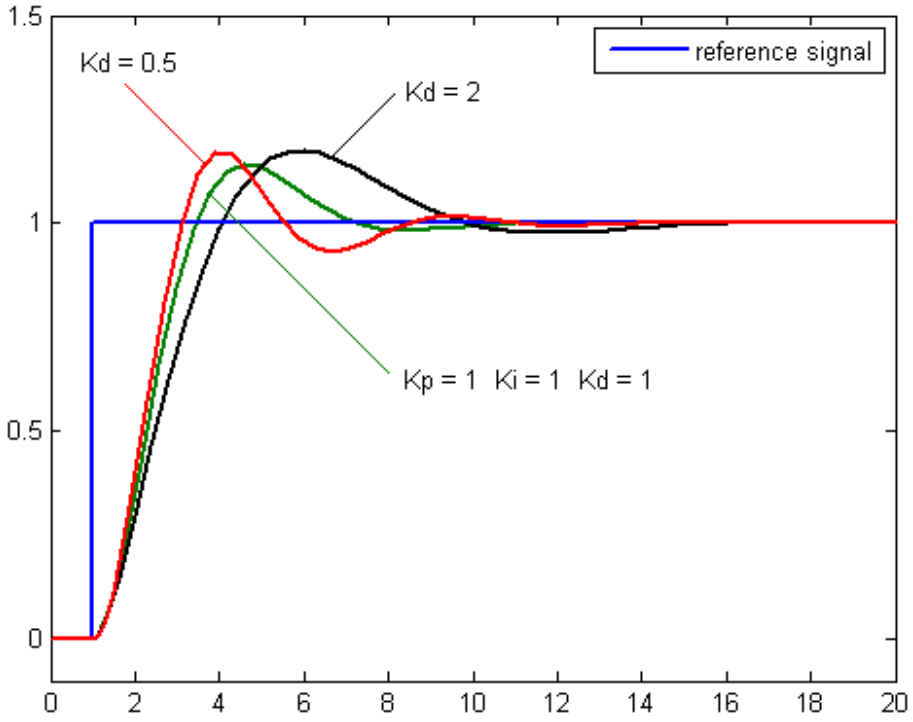
Kendali Derivative (D) akan bekerja pada saat peralihan, jika tidak ada perubahan error maka keluaran kendali nol. Kendali D memiliki aksi meredam sehingga memperbaiki lonjakan, pada keluaran kendali Derivatif (D) akan proporsional terhadap laju (rate) perubahan error. Seperti kendali I, kendali D akan dikombinasikan dengan kendali Proporsional atau dengan kendali Proposional Integral.



Gambar 19 Sinyal Tanggapan dengan memvariasikan P, Ki, Kd



Gambar 20 Sinyal Tanggapan dengan memvariasikan Ki



Gambar 21 Sinyal Tanggapan dengan memvariasikan Kd

Keluaran sinyal kendali PID dirumuskan:

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

Dalam persamaan waktu diskrit maka keluaran sinyal kendali dirumuskan:

$$PID = K_p e(k) + K_i T_s \sum_{j=0}^{k} e(j) + K_d \frac{e(k) - e(k-1)}{T_s}$$

dimana;

PID = Sinyal keluaran kendali

Kp = Gain proporsional, parameter tuning

Ki = Gain Integral, parameter tuning

Kd = Gain Derivatif, parameter tuning

Ts = waktu cuplik (sampling time)

e(k) = Error (Sp – PV)
e(k-1) = Last Error
Sp = Setpoint
PV = Variabel proses (bobot sensor pembacaan)

Dengan Kp, Ki, and Kd, semuanya positif, menandakan koefisien untuk Proporsional, Integral, dan Derivatif, secara berurutan, pada model ini:

- Proporsional (P) bertanggung jawab untuk nilai kesalahan saat ini. Contohnya, jika nilai kesalahan besar dan positif, maka keluaran kontrol juga besar dan positif.
- Integral (I) bertanggung jawab untuk nilai kesalahan sebelumnya. Contoh, jika keluaran saat ini kurang besar, maka kesalahan akan terakumulasi terus menerus, dan kontroler akan merespon dengan keluaran lebih tinggi.
- Derivatif (D) bertanggung jawab untuk kemungkinan nilai kesalahan mendatang, berdasarkan pada rate perubahan tiap waktu.

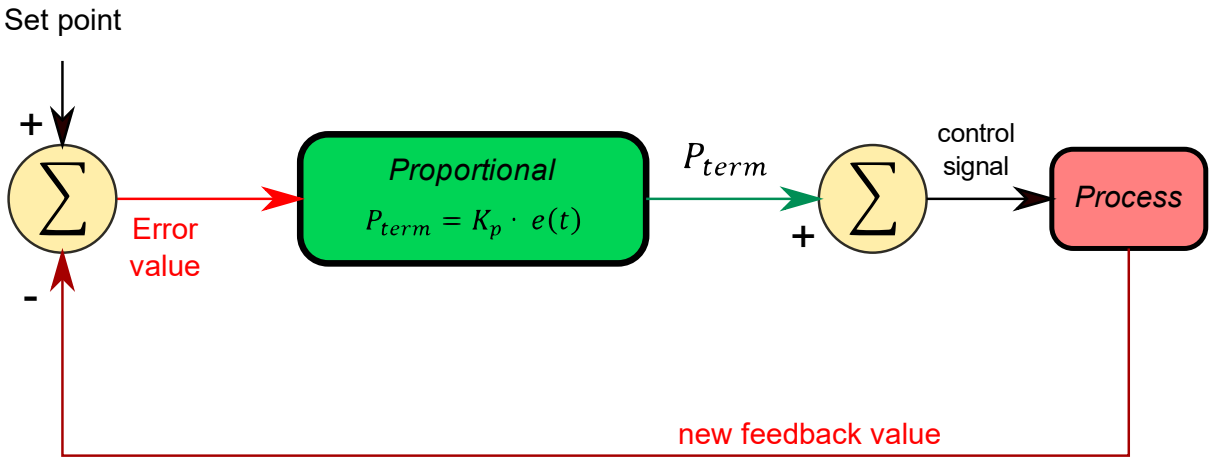
Karena kontroler PID hanya mengandalkan variabel proses terukur, bukan pengetahuan mengenai prosesnya, maka dapat secara luas digunakan. Dengan penyesuaian (tuning) ketiga parameter model, kontroler PID dapat memenuhi kebutuhan proses. Respon kontroler dapat dijelaskan dengan bagaimana responnya terhadap kesalahan, besarnya overshoot dari setpoint, dan derajat osilasi sistem. penggunaan algoritme PID tidak menjamin kontrol sistem akan optimum atau stabil. Beberapa aplikasi mungkin hanya menggunakan satu atau dua term untuk memberikan kontrol sistem yang sesuai. Hal ini dapat dicapai dengan mengontrol parameter yang lain menjadi nol. Kontroler PID dapat menjadi kontroler PI, PD, P atau I tergantung aksi apa yang digunakan. Kontroler PI biasanya adalah kontroler paling umum.

Berikut ada beberapa kondisi keluaran dengan menggunakan simulasi system kendali :

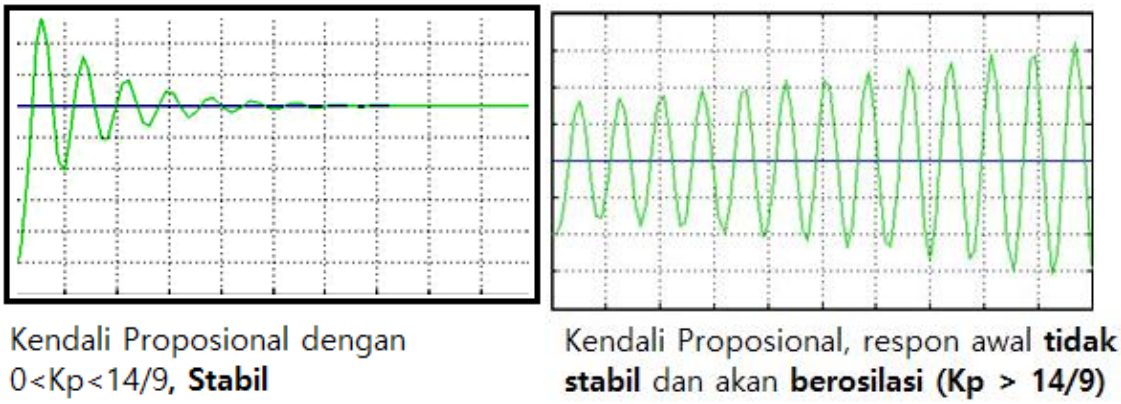
a. Kontroler Proportional

Persamaan matematis:

ó ò سي ںِ ٰΩ ò



Gambar 22 Diagram blok kendali proposional

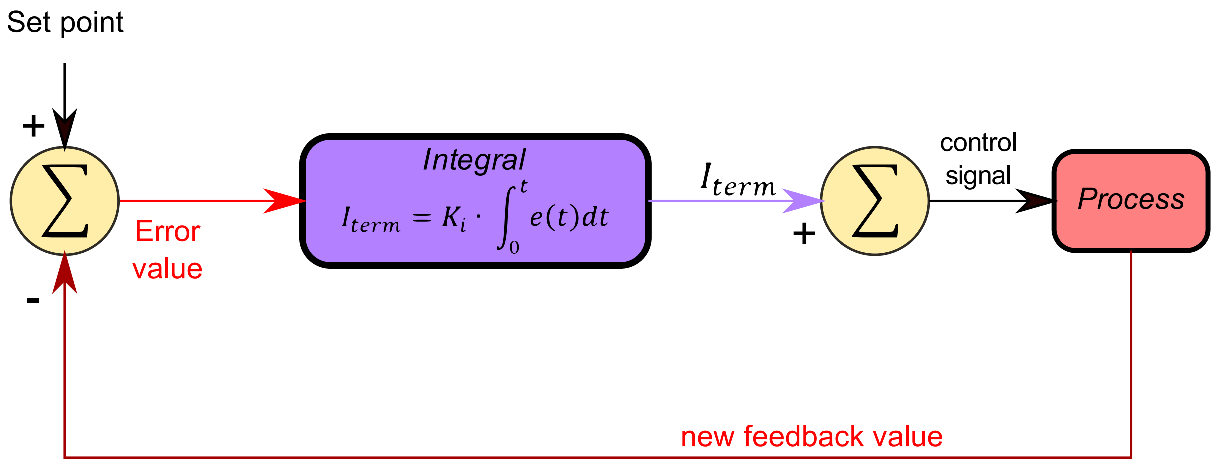


Gambar 23 Sinyal Tanggapan Kendali Proporsional

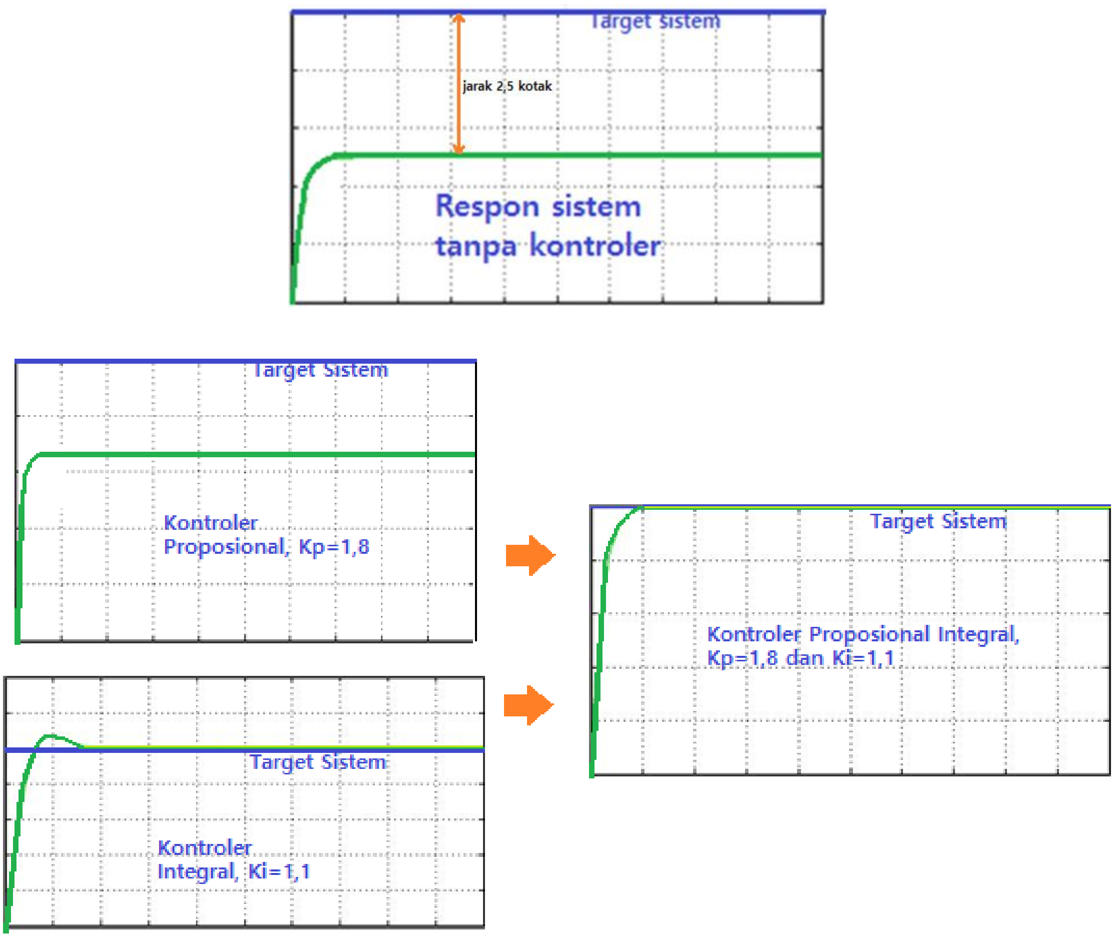
b. Kontroler Integral

Persamaan matematis :

$$u(t) = K_i \int_0^t e(t)\,dt$$



Gambar 24 Diagram blok kendali integral



Gambar 25 Sinyal Tanggapan Kendali Integral

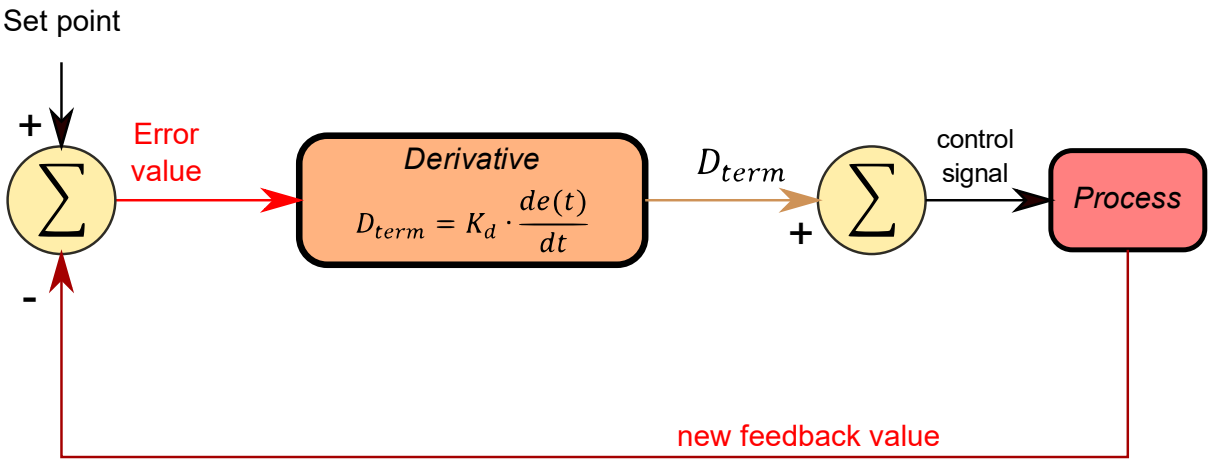
c. Kontroler Derivative

Fungsi dari kontroler derivative yaitu;

1. Memberikan efek redaman pada sistem yang berosilasi
2. Memperbaiki respon transien, karena memberikan aksi saat ada perubahan error
3. D hanya berubah saat ada perubahan error, sehingga saat ada error statis D tidak beraksi.

   Persamaan matematis :

$$\text{Output} = K_d \frac{de(t)}{dt}$$



Gambar 26 Diagram blok kendali derivative

Gambar 28. Sinyal Tanggapan Kendali Derivative

Prinsip Kendali PID pada AGV Trackless dengan berbagai persamaan matematis diatas bisa dijelaskan lebih lanjut sebagai berikut, nilai Proporsional (P) kira-kira akan sebanding dengan

posisi AGV terhadap garis. Artinya, jika AGV tepat di tengah garis maka nilai proporsional tepat 0. Nilai integral mencatat sejarah gerakan AGV, hal ini merupakan jumlah dari semua nilai proporsional yang dicatat sejak AGV mulai berjalan. Derivatif adalah tingkat perubahan nilai proporsional. Nilai P.I.D adalah ukuran kesalahan yang ditemui ketika AGV mengikuti garis. Adapun nilai Kp, Ki dan Kd adalah konstanta proporsional, integral, dan derivatif sebagai parameter yang kemudian dikalikan dengan kesalahan untuk menyesuaikan posisi AGV.

d. Menetukan Parameter Nilai Konstanta Kontroler P I D

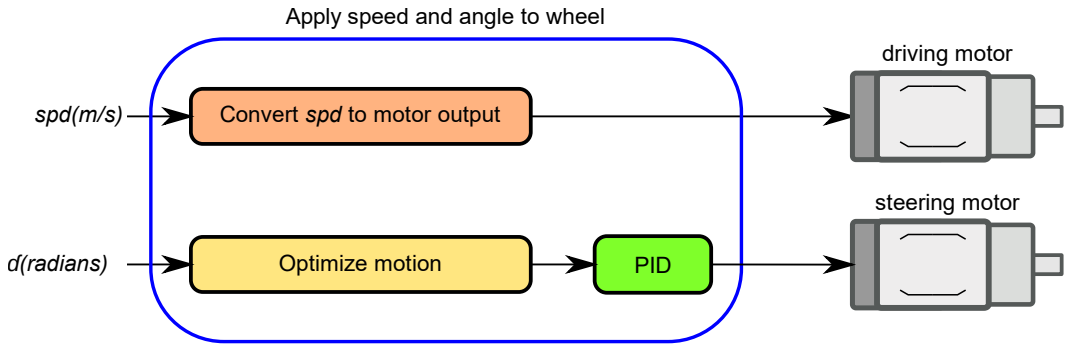
Untuk menemukan nilai parameter PID: nilai Kp, Ki dan Kd, harus melakukan ujicoba dengan mengvariasikan nilai ke-3 parameter tersebut, sehingga didapat pergerakan yang baik/stabil pada segala kondisi. Karena nilai konstanta tersebut akan berbeda-beda pada unit AGV dan ukuran line yang berbeda. Nilai konstanta PID di masukan secara *trial* and *error*, sehingga proses ini dilakukan dengan metode eksperimental nilai Kp, Ki dan Kd pada formula PID hingga ditemukan hasil sistem yang stabil. Disarankan mulai dari pemberian nilai dasar pada Driver Aktuator berupa nilai PWM yang rendah untuk mendapatkan kecepatan motor dasar yang rendah. Kemudian masukkan nilai Kp dan Kd dengan nilai kecil. Kemudian beranjak kenilai yang lebih besar dan untuk menghasilkan koreksi yang baik, perlu dikalikan dengan konstanta yang lebih besar dan dengan demikian, nilai Kp biasannya akan lebih kecil dibandingkan dengan nilai Kd. Bisa dilakukan dengan mulai hanya dengan Nilai Kp saja, dan kemudian ketika mendapatkan gerakan yang baik, mulai menambahkan nilai Kd dan lakukan ujicoba tersebut berulangkali untuk mendapatkan gerakan AGV yang halus dan baik pada segala kondisi lintasan. Demikian juga dalam menentukan nilai Ki seperti yang dilakukan pada proses mencari nilai Kd. Secara singkat bisa dilakukan langkah-langkah cara untuk masukan Kp, Ki, Kd kontrol PID pada AGV Trackless adalah sebagai berikut:

1. Menggunakan kontrol proporsional terlebih dahulu, dengan mengabaikan konstanta integratif dan derivatifnya, dengan memberikan nilai nol pada integratif dan derivatif.
2. Menambahkan terus konstanta proporsional maksimum hingga keadaan stabil namun AGV masih akan berosilasi.
3. Untuk meredam osilasi, maka tambahkan konstanta derivatif dengan membagi dua nilai proporsional, amati keadaan pergerakan AGV hingga stabil dan lebih responsif.
4. Jika sistem AGV telah stabil, kontrol integratif dapat menjadi opsional, dalam artian jika ingin mencoba-coba tambahkan kontrol integratif tersebut, namun pemberian nilai integratif yang tidak tepat dapat membuat pergerakan AGV menjadi tidak stabil.
5. Nilai set point kecepatan dan nilai batas bawah/atas memberikan acuan kecepatan AGV.

6. Nilai time sampling (waktu cuplik) juga mempengaruhi perhitungan PID, saat penggunaan kontrol integratif dan derivatif.

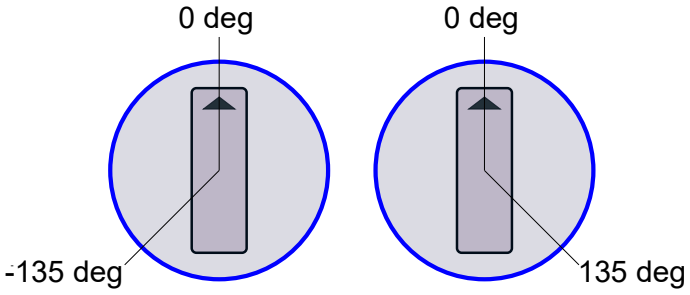
3.2. Pengendalian pada motor-motor pada *swerve drive*

Motor penggerak dan motor kemudi dikendalikan berdasarkan masukan kecepatan dan nilai sudut yang diinginkan. Motor penggerak menggunakan mode kecepatan (*speed mode*), sedangkan motor kemudi menggunakan mode posisi (*position mode*). Motor kemudi dikendalikan dengan PID agar posisi dapat dipertahankan di sudut yang diprogram oleh kontroler utama.

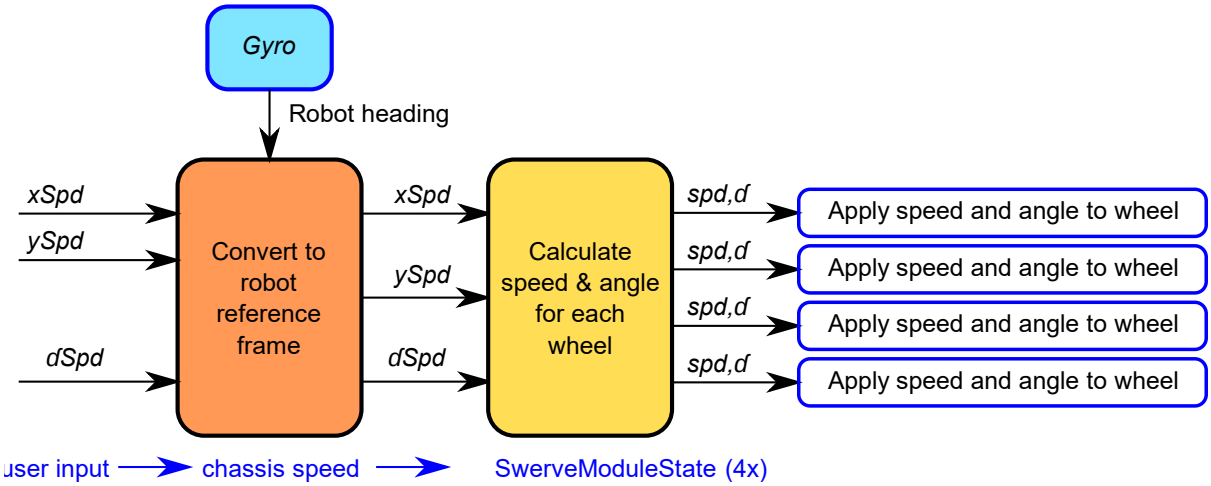


Gambar 27 Pengendalian motor penggerak dan kemudi berdasarkan kecepatan dan sudut

Pergerakan kemudi memiliki bataan putar mulai dari -135 derajat sampai 135 derajat seperti pada gambar di bawah. Untuk membatasi ini, diperlukan limit switch agar komponen mekanik dan perkabelan dapat terlindungi. Home position sensor juga terdapat pada mekanik kemudi untuk mengetahui di mana lokasi titik 0 derajat.

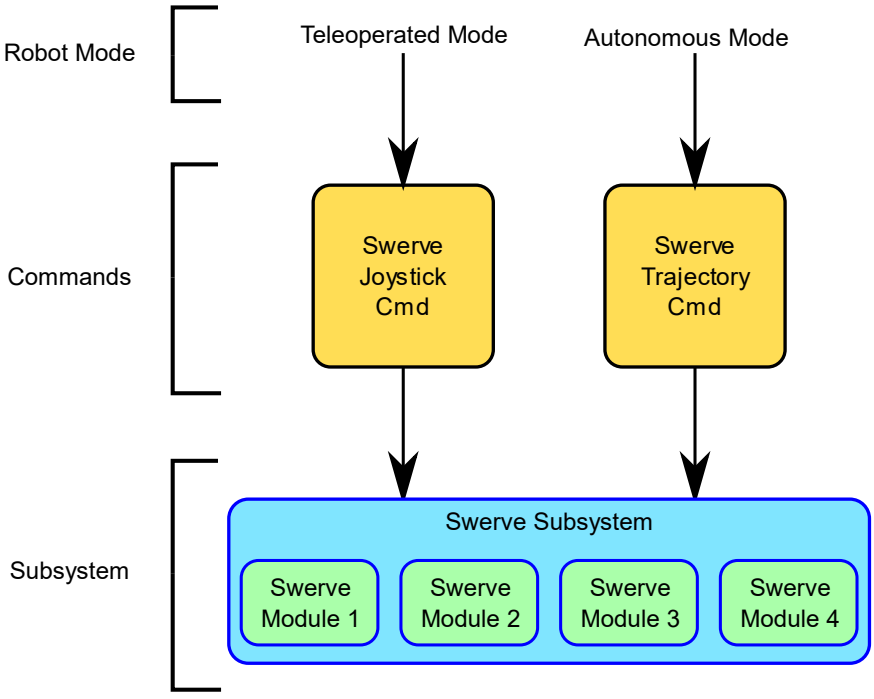


Gambar 28 Batasan sudut kemudi



Gambar 29 Pengendalian masing-masing swerve berdasar masukan pengguna dan sensor

Pengendalian motor-motor swerve akan lebih mudah dijalankan oleh pengguna jika terdapat konversi nilai-nilai kecepatan yang vektornya memiliki elemen ke arah sumbu X dan sumbu Y. Gambar di atas adalah diagram blok bagaimana mengkalkulasi nilai-nilai yang diinput oleh pengguna menjadi nilai yang diumpankan untuk menggerakkan roda dengan kecepatan dan sudut tertentu.

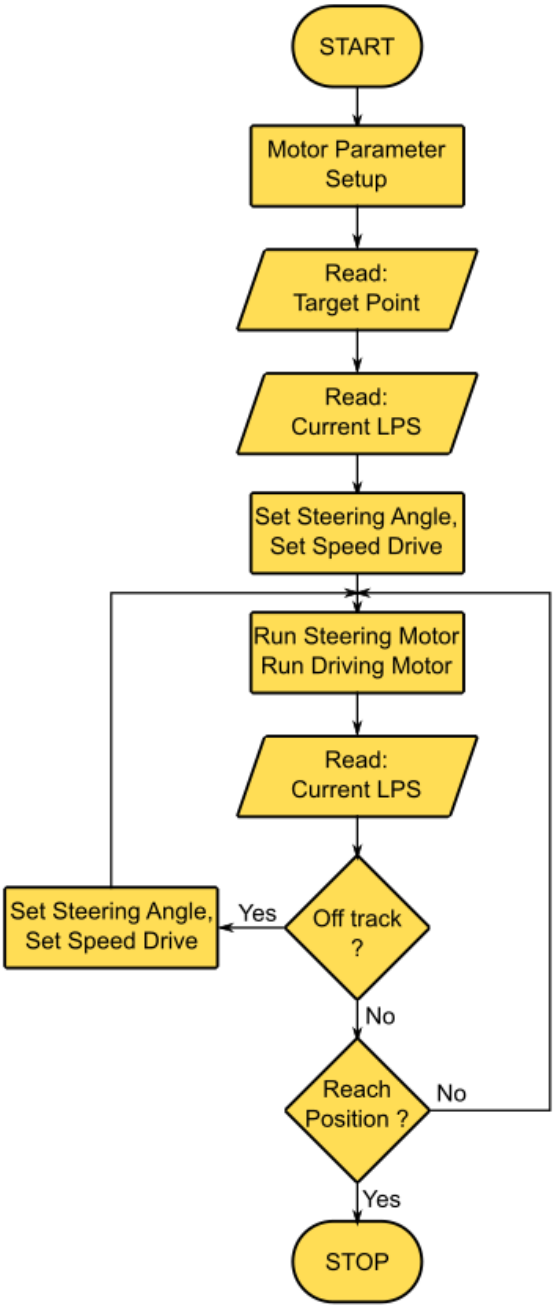


Gambar 30 Pilihan mode robot untuk kendali Swerve Drive

Gambar di atas adalah diagram blok yang menunjukkan layer aplikasi lebih tinggi, di mana pengguna dapat menentukan pergerakan AGV dengan mode manual secara jarak jauh (*Teleoperated Mode*) ataupun dengan mode otomatis menggunakan aplikasi navigasi (*Autonomous Mode*). Perintah-perintah diberikan dalam program komputer untuk menerima

masukan (misalnya joystick) atau berdasarkan rincian tujuan yang sudah disimpan pada variable array yang berisi posisi-posisi tujuan.

3.3. Perancangan program AGV



Gambar 31 Flowchart perancangan program AGV

Program ini dijalankan oleh komputer pengendali yang terdapat di AGV. Program dimulai dengan pengaturan parameter-parameter motor, antara lain percepatan, perlambatan, dan kecepatan maksimal. Selanjutnya program menunggu perintah ke mana titik yang harus dituju (*target point*). Perintah ini berasal dari aplikasi lain melalui komunikasi WiFi.

Setelah mendapatkan *target point*, posisi terkini (*actual point*) dibaca dari mikrokontroler yang terhubung dengan modul LPS.

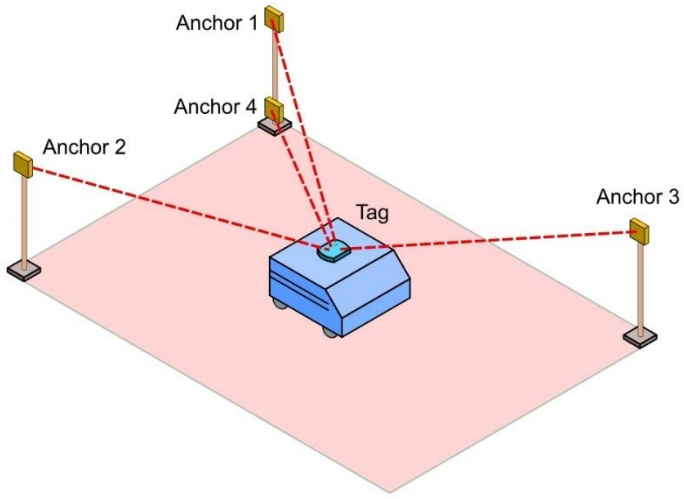
Berdasarkan target point dan actual point, maka dihitunglah sudut kemudi dan kecepatan gerak. Paramater ini digunakan untuk operasi berikutnya yaitu menjalankan motor kemudi dilanjutkan motor penggerak. Sambil berjalan, posisi actual dibaca secara periodik. Apabila keluar jalur (off track), maka perlu dilakukan pengaturan ulang terhadap sudut kemudi dan kecepatan penggerak agar motor kembali menuju posisi tujuan.

Apabila tidak keluar jalur, maka dilakukan pemeriksaan apakah AGV sudah sampai atau belum. Jika belum, maka motor penggerak melanjutkan perjalanan, jika sudah sampai maka AGV berhenti.

Saat mengaktifkan tombol ON pada AGV, maka AGV akan langsung mendeteksi jalur yang terbaca pada sensor yang digunakan sebagai pemandu AGV dalam melakukan pergerakan, sensor akan mengirimkan logika 0 pada pin alamat masing-masing sensor dan pergerakan driver AGV akan mengikuti jalur yang ada. AGV dapat menentukan suatu pilihan jalur yaitu akan berbelok kekanan atau kekiri serta pilihan lurus bila menemui jalur perempatan. Untuk menghentikannya cukup dengan menekan tombol stop.

Bab 4. Local Positioning System

Trackless AGV yaitu AGV yang memiliki pemanju jalur tanpa ada jalur fisik, sehingga pemandu berada pada program di dalam kendali AGV dengan data pengolahan dari perangkat navigasi. Beberapa kelebihan dan keuntungan dari Trackless AGV ini adalah : lingkungan kerja yang dilewati AGV tersebut memiliki estetika yang baik sebab tidak terlihat jalur-jalur pemandu AGV di lantai kerja, tidak memerlukan biaya perawatan jalur fisik yang seringkali harus diganti/dirawat akibat dilewati kendaraan/orang atau karena secara usia jalur tersebut, kemudian akan menjadi mudah dalam merubah rute yang biasa disebut fleksibilas rute.



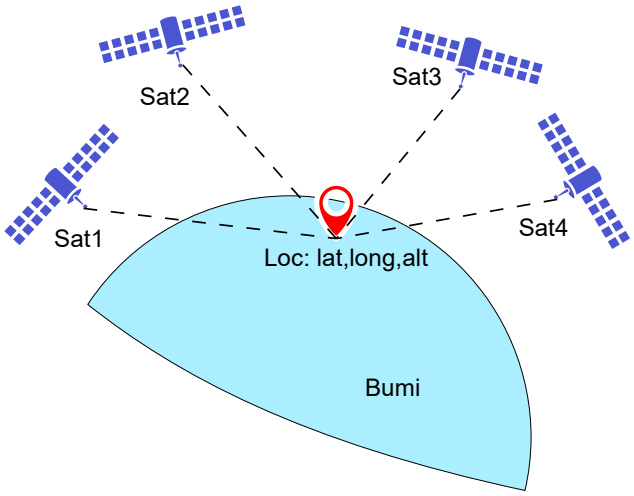
Gambar 15. Sistem *Trackless AGV*

4.1. GPS (Global Positioning System)

GPS (Global Positioning System) adalah sistem navigasi dan penentuan posisi berbasis satelit yang dikembangkan oleh Amerika Serikat dan beroperasi pada tahun 1995. Sistem navigasi ini memberikan informasi posisi global dengan prinsip one-way ranging, yaitu satelit-satelit yang mengorbit bumi secara geoseitris mengirimkan sinyal berisi informasi waktu secara akurat. Sumber waktu yang digunakan adalah jam atom yang sinkron di semua satelit. Sinyal tersebut akan diterima oleh penerima GPS yang berada di permukaan bumi, yang terdiri dari identitas satelit, lokasi satelit serta waktu sinyal dikirim. Selanjutnya penerima akan menghitung perbedaan-perbedaan waktu tersebut dan dikonversi ke dalam lokasi bujur dan lintang di mana penerima itu berada. Sistem ini dapat dikembangkan pada perangkat penerima untuk mengetahui lokasi yang diintegrasikan dengan sensor tambahan seperti gyroscope,

compass, accelerometer serta perangkat lunak pemetaan. Contohnya adalah telpon cerdas yang dilengkapi dengan sistem operasi dan aplikasi.

Pada sistem navigasi robot, GPS memiliki banyak kelebihan baik dari segi operasional maupun biaya yang relative murah. Namun begitu, kelemahan GPS terletak pada akurasi yang rendah (untuk penerapan navigasi robot), yaitu 10m-15m. Jika berada di lingkungan yang benyak bangunan, maka akurasi akan menurun atah bahkan tidak dapat menerima sinyal dari satelit. Pada banyak kasus, GPS tidak dapat digunakan di dalam ruangan.



Gambar 32 Cara kerja sistem navigasi berbasis satelit

Penerima GPS terhubung pada sebuah mikrokontroler atau prosesor untuk diolah sesuai keperluan. Antarmuka penerima GPS biasanya adalah komunikasi serial dengan protocol NMEA (National Marine Electronics Association). Struktur teks protokol NMEA dimulai dengan karakter $ dan setiap blok data dipisahkan oleh koma.

Berikut ini adalah contoh penerimaan data yang berasal dari perangkat penerima GPS. (sumber: https://en.wikipedia.org/wiki/NMEA_0183)

```
$GPGGA,092750.000,5321.6802,N,00630.3372,W,1,8,1.03,61.7,M,55.2,M,,*76
$GPGSA,A,3,10,07,05,02,29,04,08,13,,,,,1.72,1.03,1.38*0A
$GPGSV,3,1,11,10,63,137,17,07,61,098,15,05,59,290,20,08,54,157,30*70
$GPGSV,3,2,11,02,39,223,19,13,28,070,17,26,23,252,,04,14,186,14*79
$GPGSV,3,3,11,29,09,301,24,16,09,020,,36,,,*76
$GPRMC,092750.000,A,5321.6802,N,00630.3372,W,0.02,31.66,280511,,,A*43
$GPGGA,092751.000,5321.6802,N,00630.3371,W,1,8,1.03,61.7,M,55.3,M,,*75
$GPGSA,A,3,10,07,05,02,29,04,08,13,,,,,1.72,1.03,1.38*0A
$GPGSV,3,1,11,10,63,137,17,07,61,098,15,05,59,290,20,08,54,157,30*70
$GPGSV,3,2,11,02,39,223,16,13,28,070,17,26,23,252,,04,14,186,15*77
$GPGSV,3,3,11,29,09,301,24,16,09,020,,36,,,*76
$GPRMC,092751.000,A,5321.6802,N,00630.3371,W,0.06,31.66,280511,,,A*45
```

Format kalimat standard pada awal teks (diawali karakter $) memiliki arti tertentu. Contoh:

```
$GPGGA,092751.000,5321.6802,N,00630.3371,W,1,8,1.03,61.7,M,55.3,M,,*75
```

$GPGGA adalah pesan yang memberikan informasi lokasi tiga dimensi.

- **092751**– menunjukkan waktu di mana lokasi GPS Modul, 09:27:51 UTC
- 5321.6802,N – lintang sesuai angka (lintang utara)
- 00630.3371,W – bujur sesuai angka (bujur barat)
- **1** – Fix quality (0 = invalid; 1= GPS fix; 2 = DGPS fix; 3 = PPS fix; 4 = Real Time Kinematic; 5 = Float RTK; 6 = estimated (dead reckoning); 7 = Manual input mode; 8 = Simulation mode)
- **8** – jumlah satelit yang dilacak
- **1.03** – Horizontal dilution of position
- **61.7, M** – Altitude / Ketinggian, dalam meter di atas permukaan laut
- **55.3, M** – Tinggi geoid (permukaan laut rata-rata) di atas ellipsoid WGS84
- Field kosong – waktu dalam detik sejak pembaruan DGPS terakhir
- Field kosong – nomor ID stasiun DGPS.
- *75 – data checksum, selalu dimulai dengan *

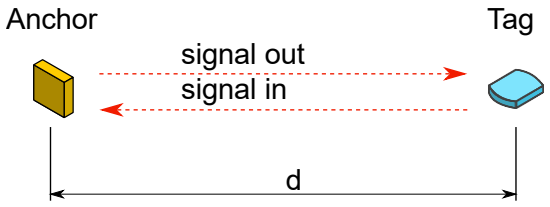
4.2. LPS (Local Positioning System)

LPS (*Local Positioning System*) merupakan sistem penentuan lokasi dalam daerah lokal atau ruang lingkup tertentu secara akurat mengetahui letak atau posisi. Sistem ini umumnya menggunakan UWB (*ultra wide band*) sebagai penerapannya. Alasannya menggunakan UWB yaitu karena menggunakan energi yang rendah untuk komunikasi jarak dekat atau pendek dan karena akurasi yang dibutuhkan sangat tinggi maka membutuhkan kecepatan pengiriman yang cukup tinggi juga. Dari hal tersebut, LPS yang digunakan menggunakan UWB untuk penerapan sistemnya.

LPS menggunakan UWB umumnya membutuhkan minimal tiga komponen *transmitter* atau komponen yang mengirimkan sinyal. Dan sebagai penerimanya atau *received* akan menerima data yang berupa lokasi dan posisi dirinya secara akurat. Dari hal itu, sebagai awalan membentuk sistem LPS pada suatu tempat, maka diperlukan kalibrasi awal pada komponen *transmitter* yang digunakan. Karena sistem LPS hampir sama dengan sistem GPS, maka untuk kalibrasi diperlukan sistem kontrol sebagai monitoring keadaan komponen *transmitter* atau dalam GPS seperti satelit yang digunakan. Ketika satelit LPS atau komponen *transmitter* yang ada sudah terkalibrasi, maka komponen received dapat digunakan untuk mencari nilai lokasinya.

Cara kerja LPS menggunakan prinsip Two-way Ranging, yaitu sebuah Anchor (mirip satelit pada GPS) mengirimkan sinyal ke Tag, lalu Tag membalas sinyal tersebut. Anchor

dipasang di posisi tetap, sedangkan Tag dipasang pada sebuah AGV. Anchor mengukur waktu antara sinyal keluar dan sinyal masuk, lalu menghitung jarak dengan memperhitungkan kecepatan cahaya.



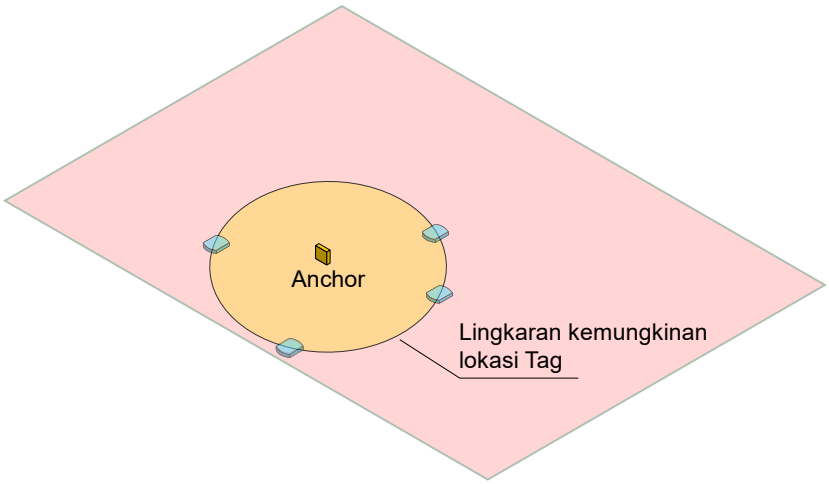
Gambar 33 Prinsip Two-way Ranging

$$d = \frac{c \times t}{2}$$

d= distance

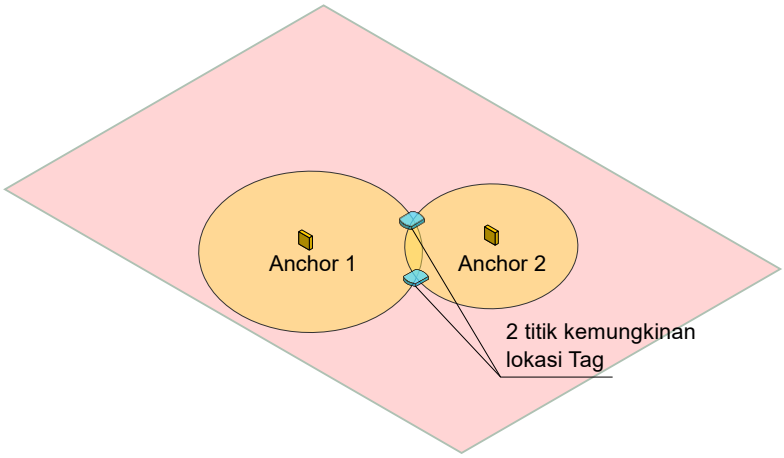
c= speed of light = 299,792,458 m / s

t= time difference



Gambar 34 Kalkulasi perkiraan posisi Tag dengan satu Anchor

Dengan hanya sebuah Anchor, sebuah Tag hanya dapat diketahui jaraknya saja, namun posisi tepatnya tidak diketahui karena berada di salah satu titik pada sebuah lingkaran. Dengan dua Anchor yang diletakkan dengan jarak tertentu dan sebuah Tag berada di antaranya, maka ada dua kemungkinan posisi yang dapat diketahui.



Gambar 35 Kalkulasi perkiraan posisi Tag dengan dua Anchor

Jika jarak d1, r1 dan r2 diketahui, maka jarak Tag terhadap Anchor 1 (d1) dapat dihitung.

Solusi: d1 = x

Persamaan lingkaran C1:

$$x^2 + y^2 = r_1^2$$

Persamaan lingkaran C2:

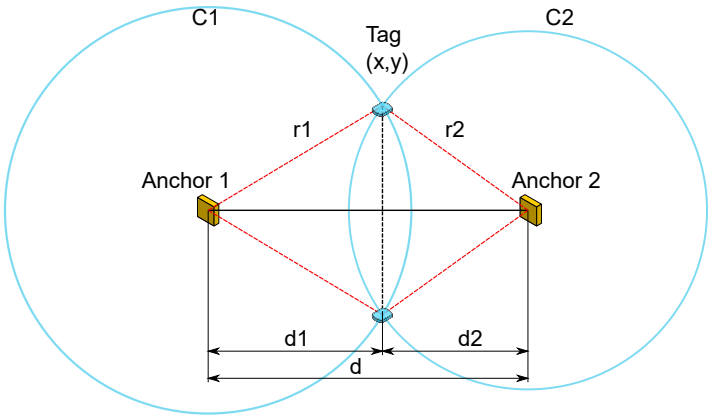
$$(x-d)^2 + y^2 = r_2^2$$

Titik perpotongan C1 dan C2 memiliki koordinat y yang sama sehingga eliminasi y pada persamaan-persamaan di atas menjadi:

$$x^2 + y^2 = r_1^2$$

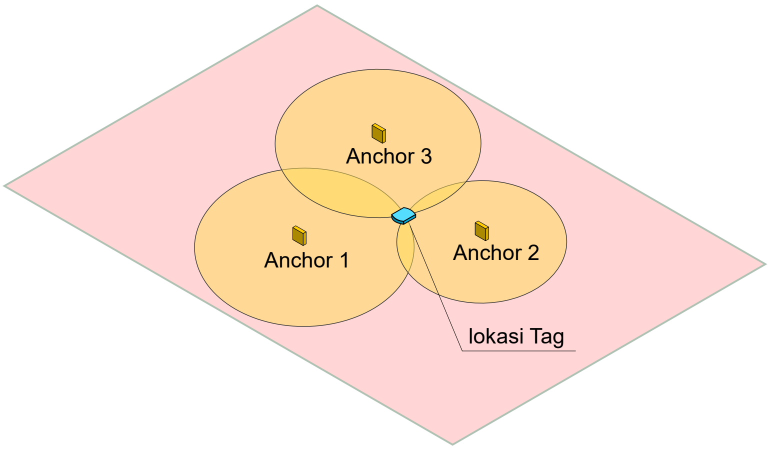
$$x^2 - 2dx + d^2 + y^2 = r_2^2$$

$$x = \frac{r_1^2 - r_2^2 + d^2}{2d}$$



Gambar 36 Geometri posisi Tag dengan dua Anchor

Dengan tiga buah Anchor yang terletak pada satu bidang datar, posisi sebuah Tag dapat diketahui koordinatnya dalam bidang dua dimensi (X,Y).

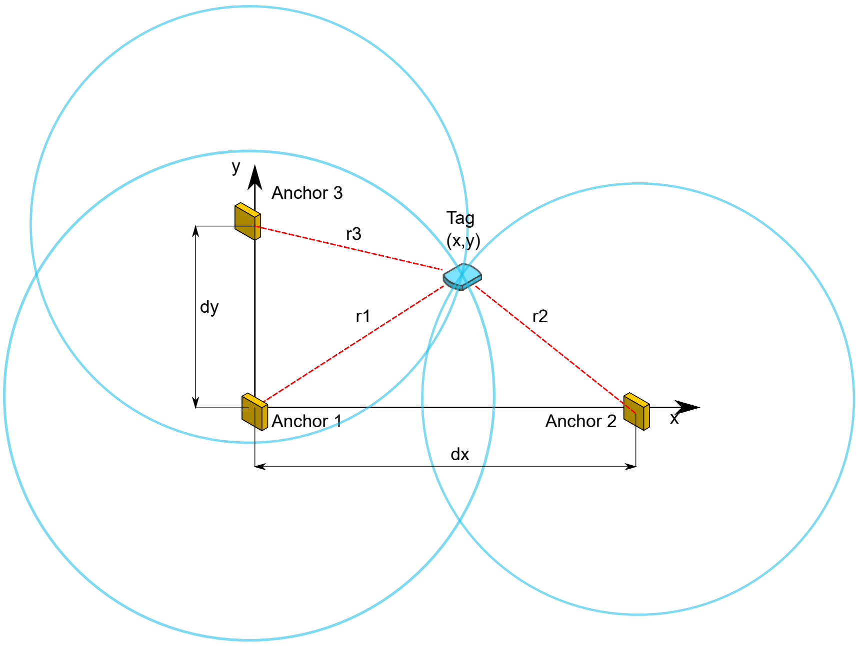


Gambar 37 Kalkulasi perkiraan posisi Tag dengan tiga Anchor

Pada bidang 2 dimensi, komponen posisi Tag (x,y) dapat dihitung dengan persamaan:

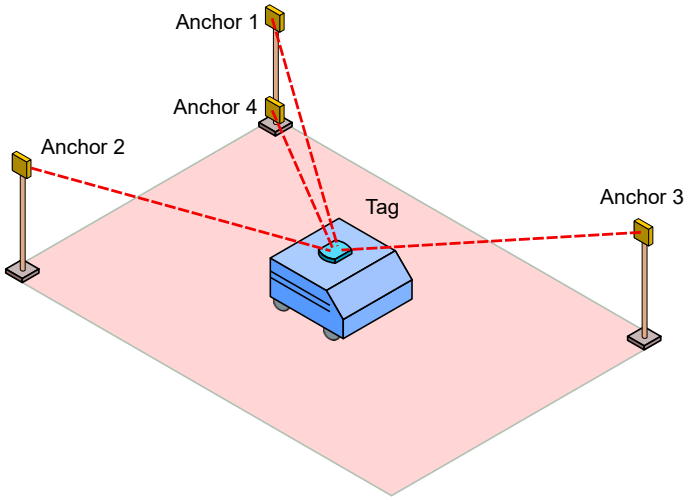
ὠسي ἰ̇ ط ي ἰ̇ ط ὨΩ̣ / ςΏ̣

ὡسي ἰ̇ ط ي ἰ̇ ط ὨΩ̣ / ςΏ̣



Gambar 38 Geometri posisi Tag dengan tiga Anchor

Apabila ingin mengetahui posisi sebuah Tag menurut koordinat kartesian tiga dimensi, maka diperlukan satu lagi Anchor yang diletakkan tegak lurus terhadap bidang yang dibentuk tiga Anchor sebelumnya.



Gambar 39 Kalkulasi perkiraan posisi Tag dengan empat Anchor

Ketika Tag melewati radius Anchor maksimal, maka Tag tidak akan menerima data dari Anchor. Untuk memperluas jangkauan, maka perlu ditambahkan lagi Anchor di area yang lebih luas. Sistem jaringan komputer dapat dibangun untuk memperluas jaringan LPS.

4.2.1. Nilai Data yang Didapat dari LPS

Untuk nilai data yang didapat dari sistem LPS, umumnya sudah menjadi nilai koordinat seperti x, y, dan z. Tetapi tergantung pada *transmitter* yang digunakan, semakin banyak *received* melacak *transmitter* yang ada maka fitur yang dapat digunakan akan semakin bertambah dan akurasi dalam menentukan posisi juga semakin detail.

```
POS ID 0x0, x(mm): 2737, y(mm): 571, z(mm): 0
POS ID 0x0, x(mm): 2697, y(mm): 630, z(mm): 0
POS ID 0x0, x(mm): 2718, y(mm): 580, z(mm): 0
POS ID 0x0, x(mm): 2718, y(mm): 571, z(mm): 0
POS ID 0x0, x(mm): 2699, y(mm): 583, z(mm): 0
POS ID 0x0, x(mm): 2700, y(mm): 569, z(mm): 0
POS ID 0x0, x(mm): 2704, y(mm): 608, z(mm): 0
POS ID 0x0, x(mm): 2683, y(mm): 567, z(mm): 0
POS ID 0x0, x(mm): 2683, y(mm): 601, z(mm): 0
```

Sebagai contoh nilai data yang didapat ketika menggunakan tiga *transmitter* akan menghasilkan koordinat dengan posisi dua dimensi atau hanya sumbu x dan sumbu y.

4.2.2. Orientasi Tiga Dimensi dari LPS

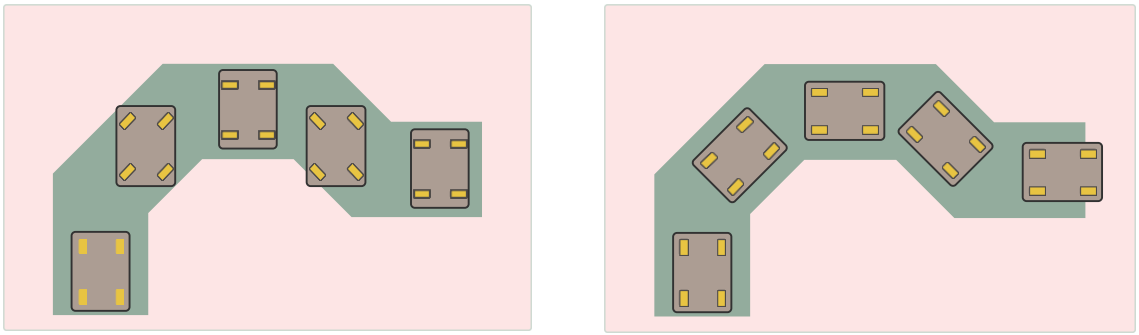
Selain mendapatkan nilai data berupa koordinat, sistem LPS juga bisa mendapatkan nilai lainnya yaitu orientasi tiga dimensi. Nilainya seperti keadaan *received* itu sendiri seperti arah, kecepatan, waktu dan semacamnya. Umumnya dalam penggunaan sehari-hari, nilai dari orientasi tiga dimensi yaitu arah karena akan mempengaruhi pergerakan setelah mendapatkan koordinat.

```
12,100353750,51,13,945,296,148,1793,2,2,1,42,48,-12,16374,103,-436,-378,-3,-2,-33,51,13,979,34,0,0,3,0
10,100353750,48,11,946,296,148,1793,-2,0,1,42,48,-12,16374,103,-436,-378,-1,0,-31,51,13,979,34,0,0,3,0
11,100353750,50,12,947,302,142,1785,7,-3,0,42,48,-12,16374,103,-436,-378,0,0,-31,51,13,979,34,0,0,3,0
10,100353750,51,13,948,302,142,1785,0,0,-1,42,48,-12,16374,103,-436,-378,-4,-1,-33,51,13,979,34,0,0,3,0
12,100353750,47,12,946,302,142,1785,-2,-1,2,42,48,-12,16374,103,-436,-378,-4,0,-35,51,13,979,34,0,0,3,0
10,100353750,47,14,943,302,142,1785,0,-2,1,42,48,-12,16374,103,-436,-378,-2,0,-35,51,13,979,34,0,0,3,0
11,100353750,49,13,944,302,142,1785,-2,3,0,42,48,-12,16374,103,-436,-378,-2,-1,-30,51,13,979,34,0,0,3,0
10,100353750,50,13,943,296,160,1793,-6,2,0,42,48,-12,16374,103,-436,-378,-1,0,-34,51,13,979,34,0,0,3,0
12,100353750,49,12,949,296,160,1793,0,0,1,42,48,-12,16374,103,-436,-378,-3,-1,-32,51,13,979,34,0,0,3,0
10,100353750,48,12,947,296,160,1793,6,0,2,42,48,-12,16374,103,-436,-378,-2,0,-33,51,13,979,34,0,0,3,0
11,100353750,48,13,946,296,160,1793,2,-2,0,42,48,-12,16374,103,-436,-378,-2,0,-33,51,13,979,34,0,0,3,0
10,100353750,49,13,945,296,142,1800,-2,4,0,42,48,-12,16374,103,-436,-378,-3,0,-29,51,13,979,34,0,0,3,0
12,100353750,47,14,949,296,142,1800,5,-3,1,42,48,-12,16374,103,-436,-378,-2,0,-31,51,13,979,34,0,0,3,0
10,100353750,49,13,947,296,142,1800,-2,0,-1,42,48,-12,16374,103,-436,-378,-2,0,-35,51,13,979,34,0,0,3,0
11,100353750,49,14,943,296,142,1800,0,2,1,42,48,-12,16374,103,-436,-378,-2,-1,-35,51,13,979,34,0,0,3,0
10,100353750,48,12,943,296,142,1800,5,0,2,42,48,-12,16374,103,-436,-378,-4,0,-31,51,13,979,34,0,0,3,0
11,100353750,47,13,947,308,137,1798,0,2,1,42,48,-12,16374,103,-436,-378,-4,-1,-31,51,13,979,34,0,0,3,0
11,100353750,47,12,947,308,137,1798,4,-2,1,42,48,-12,16374,103,-436,-378,-2,1,-33,51,13,979,34,0,0,3,0
10,100353750,49,14,945,308,137,1798,2,1,2,42,48,-12,16374,103,-436,-378,-2,0,-34,51,13,979,34,0,0,3,0
```

Dalam penerimaan datanya akan menghasilkan seperti pada gambar tersebut. Akan menghasilkan banyak data yang didapat, ketika menggunakan LPS. Walaupun seperti itu, data-data tersebut dapat disaring sesuai penggunaan yang diperlukan.

4.2.3. Integrasi LPS sebagai Sistem Kontrol AGV

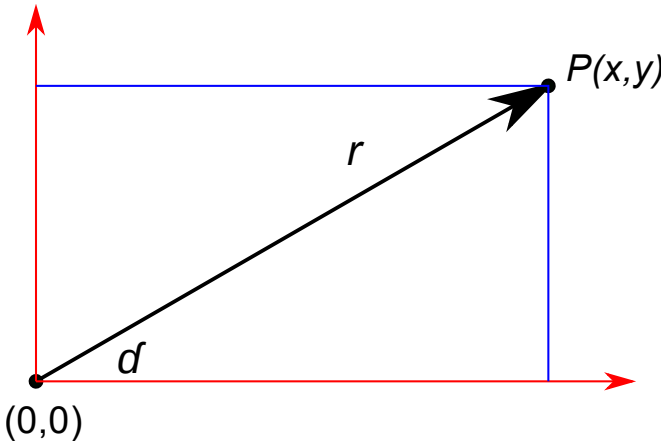
AGV dapat dikendalikan berdasarkan dua model gerakan yaitu gerakan dengan arah tetap (*fix orientation*) dan gerakan dengan arah sesuai jalur (*path orientation*). Penggunaan LPS pada AGV dibutuhkan data berupa koordinat pada bidang 2 dimensi (nilai x dan y) dalam kontrolnya. Untuk mengatur pergerakannya diperlukan sistem koordinat matematika dalam perhitungannya.



(a) (b)

Gambar 40 Fix orientation (a) dan Path orientation (b)

Pergerakan AGV dari satu titik ke titik lain (P0 Ą _



Gambar 41 Koordinat polar dan kartesian

Jika sudah ditentukan titik akhir berupa nilai x akhir dan y akhir maka perlu dibandingkan dengan nilai yang diketahui yaitu nilai x dan y dari sistem LPS yang ada. Nilai itu diolah untuk mengetahui sudut yang ada dengan rumus seperti berikut:

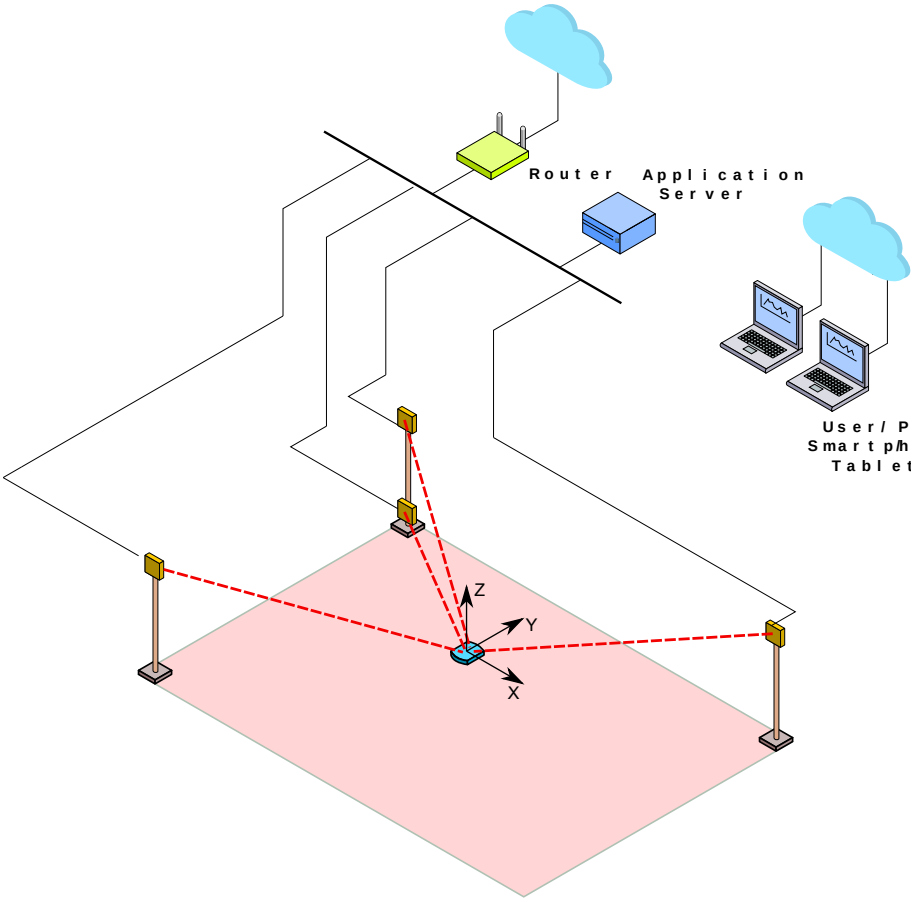
$$\hat{O}\ \acute{A}\text{–}\hat{I}\ \text{سي}\frac{\acute{\omega}}{\grave{\omega}}$$

$$\text{—سي}\ \grave{o}\ \ddot{\omega}^{\circ}\text{ع}\ \frac{\acute{\omega}}{\grave{\omega}}$$

Nilai sudut yang dihitung akan dibandingkan dengan nilai arah dari orientasi tiga dimensi komponen Tag yang ada. Ketika kurang dari sudut yang telah dihitung, maka pergerakan AGV harus melawan arah jarum jam. Tetapi sebaliknya, ketika melebihi sudut yang telah dihitung, maka pergerakan AGV searah jarum jam.

Setelah menyesuaikan arah yang telah dihitung dari nilai x dan y awal dengan x dan y akhir, maka arah akan tepat pada titik akhir yang dituju. Dari hal itu, perlu kontrol AGV untuk mendekati titik akhir yang ditentukan menyesuaikan arah tadi. Jika pergerakan AGV sudah

mendekati titik akhir, maka diperlukan sistem berhenti agar mencapai titik akhir dengan tepat. Setelah sudah sampai radius titik akhir, maka AGV harus diperintahkan untuk berhenti agar posisi akhir yang ditentukan sesuai dengan posisi AGV yang telah dikontrol.



Gambar 42 Sistem jaringan pada LPS

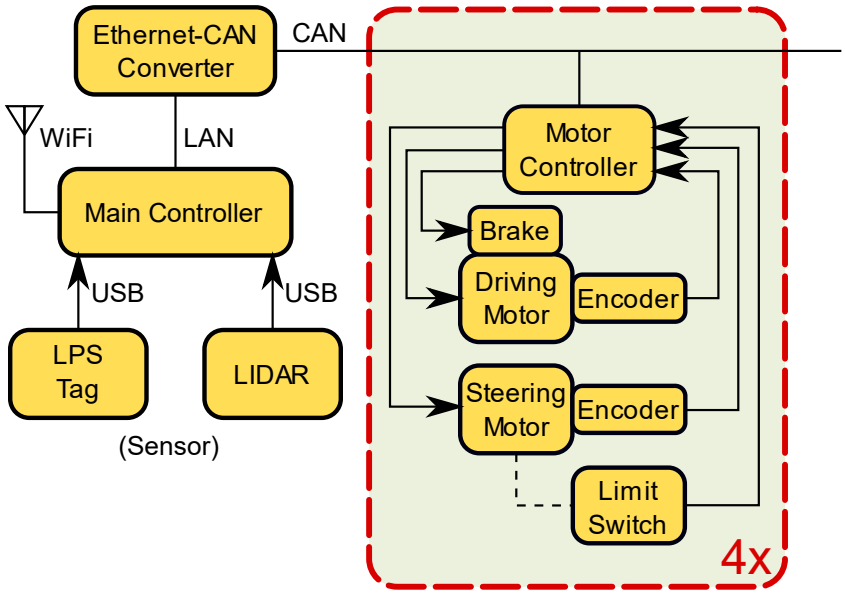
Dalam sistemnya diperlukan nilai x dan y serta arah yang diketahui melalui sistem LPS. Jika ingin mengontrol AGV untuk mencapai titik akhir (x_t, y_t), maka diperlukan perhitungan seperti penjelasan sebelumnya untuk menyesuaikan gerakannya.

Sistem jaringan pada LPS terdiri dari router, server, serta koneksi kabel ethernet ke masing-masing Anchor. Posisi Tag pada AGV dibaca dan dikirim ke server, untuk selanjutnya diolah sesuai aplikasi pada server. Di sisi client, sebuah PC/Laptop terdapat aplikasi navigasi yang berguna untuk merencanakan dan mengatur navigasi AGV.

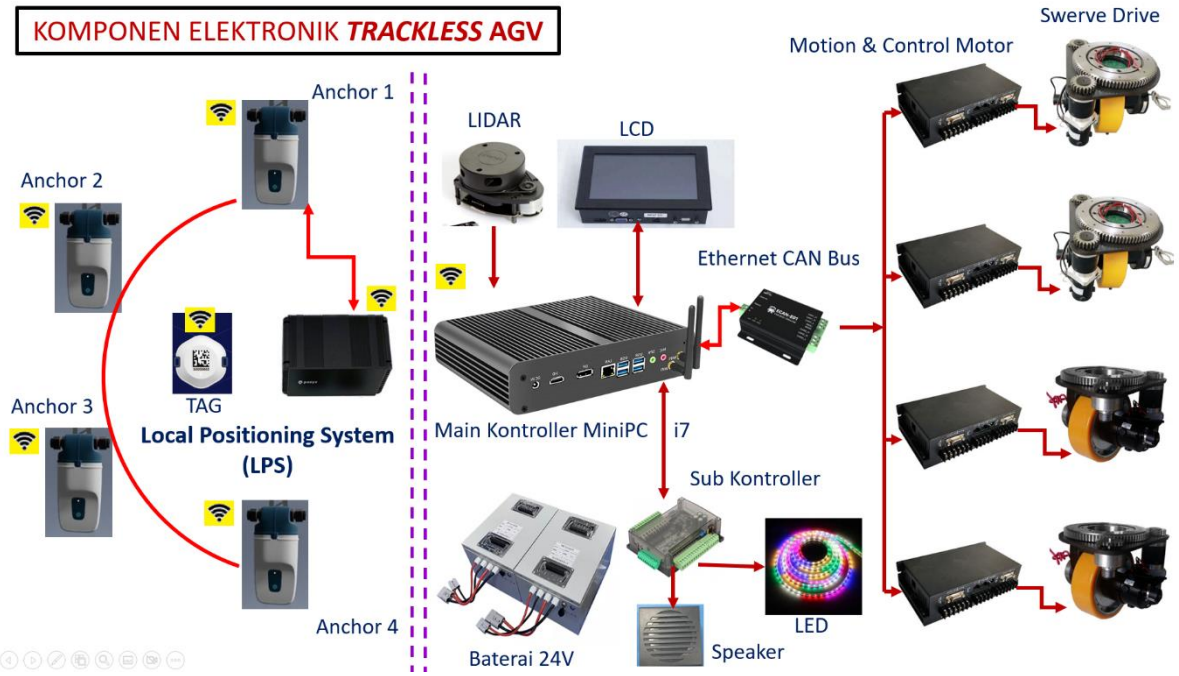
Bab 5. Konsep Pembuatan AGV

5.1. Pembuatan Rangkaian Elektrik AGV

Pembuatan rangkaian elektrik pada AGV ini digunakan *MiniPC* sebagai piranti proses yang akan mengolah data untuk diproses pada sistem navigasi dan sistem kendali swerve drive.



Gambar 43 Diagram blok sistem Trackless AGV



Gambar 44 Diagram block komponen-komponen Trackless AGV

5.2. Mini PC sebagai Main Controller,

bagian ini sebagai pengendali utama (*main controller*), yaitu mengolah masukkan dari rangkaian sensor untuk dilakukan operasi program yang selanjutnya memberikan respon ke rangkaian keluaran pada *driver actuator* yang selanjutkan akan menggerakkan motor.

Mini PC yang berperan sebagai "otak" pengendali AGV ini menggunakan prossesor i7. Program yang digunakan untuk mengendalikan AGV dikembangkan dengan menggunakan Microsoft Visual Studio dengan Bahasa pemprograman Visual Basic.NET.

5.3. Motor Swerve Drive

Modul ini merupakan actuator yang berperan sebagai perangkat keluaran berupa elektromekanik yang memiliki daya gerak. Ada dua motor berjenis BLDC (Brushless Direct Current) yang digunakan, masing-masing untuk penggerak dan kemudi. Setiap motor dipasangkan roda gigi **(***gear box*) yang berfungsi sebagai pereduksi putaran dan juga berfungsi menghasilkan kekutan putar (torsi) yang lebih besar. Setiap poros motor dihubungkan dengan encoder yang digunakan untuk masukan umpan balik untuk pengendalian loop tertutup. Pada motor penggerak dipasang juga sistem pengereman magnetic (magnetic brake).

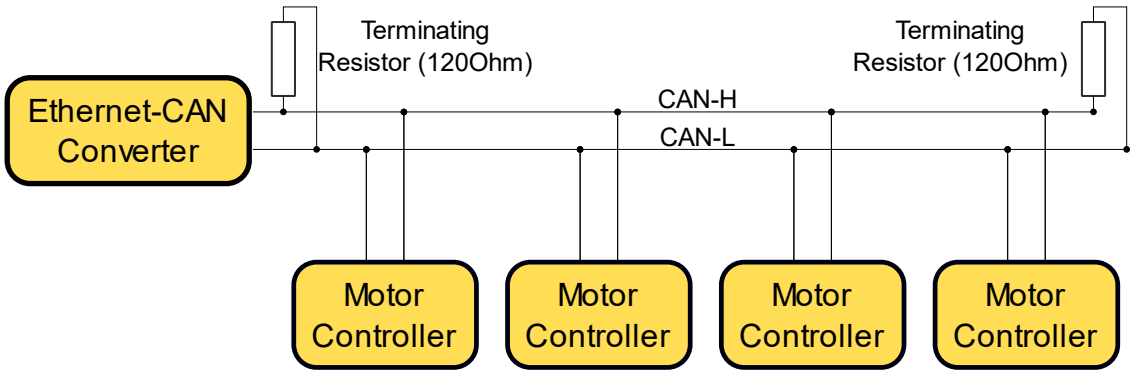
Motor Swerve Drive ini bertugas untuk menggerakkan AGV ke arah yang telah ditentukan oleh kontroler. Kebutuhan gaya untuk menggerakkan AGV harus disesuaikan dengan torsi motor DC yang digunakan, sebab apabila tidak sesuai maka AGV tidak dapat bergerak atau sebaliknya akan menjadi overspec yang berujung tidak efisien.

5.4. Pengendail gerakan dan penggerak motor BLDC (Motion Control and Motor Driver)

Pengendali motor yang digunakan adalah SYNTRON LS20530DG. Pada pengendali ini juga terdapat driver untuk menggerakkan motor BLDC. Satu unit alat ini dapat mengendalikan dua buah motor BLDC, menerima input encoder dari masing-masing motor, serta sebuah pengendali rem (brake) untuk motor penggerak.

5.5. Ethernet – CAN converter

Untuk berkomunikasi dengan modul SYNTRON, diperlukan sambungan kabel Controller Area Network (CAN) menuju ke main controller. Karena sebuah PC tidak terdapat koneksi CAN, maka diperlukan sebuah Ethernet-CAN converter (ECAN-01). Modul ini terdapat sebuah konektor RJ45 yang dihubungkan ke komputer, dan terdapat dua kanal CAN. Salah satu kanal CAN dapat dihubungkan ke beberapa modul motion controller.



Gambar 45 Rangkaian Controller Area Network

Rangkaian koneksi Ethernet-CAN ke motor controller dapat dilihat pada rangkaian di atas. Jaringan CAN memerlukan dua buah kabel, masing-masing CAN-H dan CAN-L yang dihubungkan secara parallel pada setiap perangkat. Pada ujung-ujung rangkaian wajib diberi resistor 120ohm agar komunikasi dapat berjalan. Setiap perangkat sudah terdapat resistor dan dilengkapi dengan saklar, sehingga pengguna cukup mengaktifkan saklar saja, on atau off (tidak perlu menambahkan resistor).

Dokumentasi cara penggunaan dan spesifikasi ECAN-01 dapat diakses pada situs: https://www.cdebyte.com/products/ECAN-E01

5.6. Baterai dan Charger

Baterai merupakan sumber listrik searah yang digunakan pada AGV. Ada dua buah baterai bertegangan 12V yang dirangkai secara seri, sehingga menghasilkan tegangan total 24volt. Perlu diperhatikan bahwa untuk menjaga keamanan rangkaian dalam mencegah kerusakan alat, maka perlu dipasang sekering-sekering pada titik-titk sebelum controller dan sebelum drive. Komponen pengaman lain yaitu Emergency-Stop yang dipasang pada tempat yang mudah dijangkau. Emergency-Stop memutus tegangan dari battery yang menuju drive untuk menghentikan motor apabila terjadi keadaan darurat.

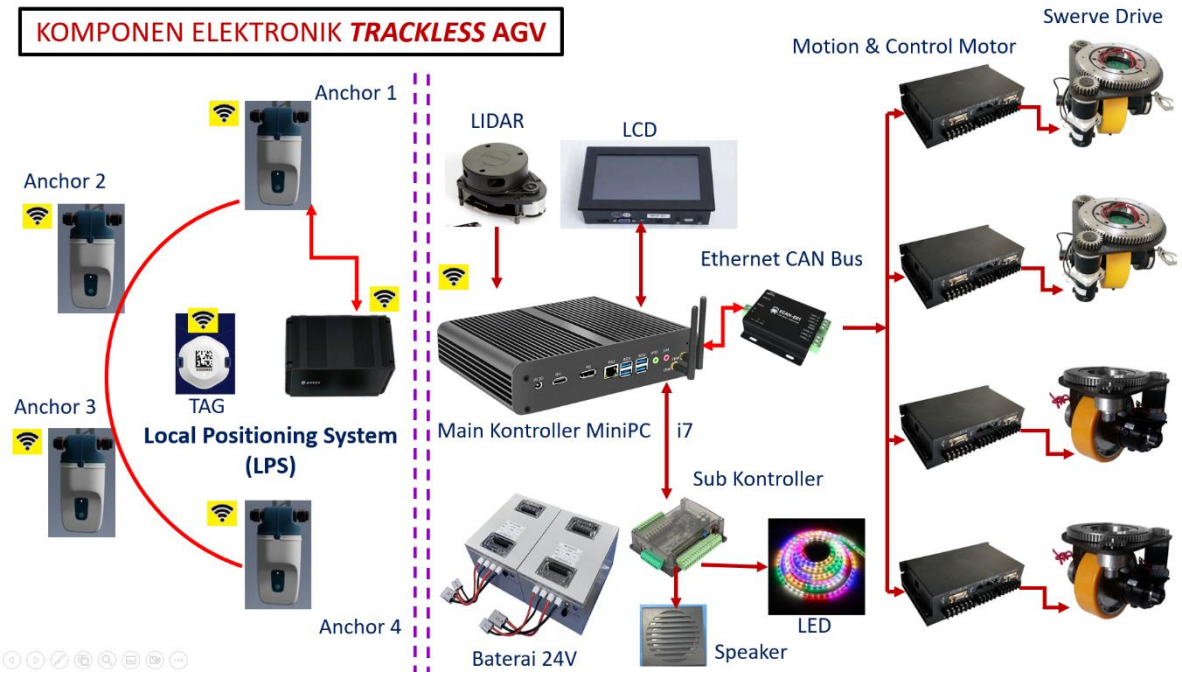
Pengisi daya (charger) merupakan alat penting dalam memberikan energi listrik dari 220volt AC ke battery. Selama pengisian, rangkaian tenaga ke AGV diputus sementara.

5.7. Perangkat Local Positioning System

Perangkat navigasi utama pada sistem AGV ini adalah Local Positiong System dengan teknologi transmisi radio menggunakan frekuensi Ultrawide band (UWB). Model alat yang digunakan yaitu Pozyx Enterprise RTLS. Spesifikasi alat yang digunakan yaitu:

- ¶ Real-time Location Systems (RTLS) berbasis teknologi Ultra-wideband (UWB) menggunakan smart algorithms and machine learning.
- ¶ Hyperaccurate up to 10cm, untuk pemposisian dalam ruangan
- ¶ Jangkauan area : +/- 300m2
- ¶ Pemprosesan data mencapai 1200 position/detik
- ¶ Dapat di-*scaled up* untuk jumlah Tag dan Anchor tak terbatas.

Dokumentasi lengkap cara penggunaan dan pemprograman dapat diakses pada situs: https://docs.pozyx.io/enterprise/



Gambar 46 Diagram blok sistem elektronik dan elektromekanik AGV

Bab 6. PENUTUP

Automated Guided Vehicle (AGV) tanpa jalur fisik merupakan sebuah teknologi di bidang otomasi sebagai alternatif model untuk mendukung transfer material di beberapa bidang seperti industri manufaktur, medical, logistic dan sebagainya. Local Positioning System (LPS) dapat membantu AGV dalam proses navigasi untuk mengetahui posisi secara realtime. Sebuah Tag yang dipasang pada sebuah AGV dapat berkomunikasi dengan Anchor yang mengukur waktu sinyal keluar dan sinyal masuk, sehingga dapat dikonversi ke dalam satuan jarak. Dengan beberapa Anchor, posisi sebuah AGV dapat dihitung dalam koordinat (X,Y,Z) pada ruang tiga dimensi. Selain itu, sebuah Tag juga terdapat gyroscope untuk menentukan orientasi AGV. Dengan adanya data posisi dan orientasi, maka proses pemandu AGV dapat dilakukan secara virtual melalui software.

DAFTAR PUSTAKA

[1] Ogata, Katsuhiko. 1997. Teknik Kontrol Automatik Jilid 1. Jakarta: Penerbit Erlangga

[2] Pakdaman,M.Sanaatiyan,M.M.,"Design and Implementation of Trackless Robot,", Second International Conference on Computer and Electrical Engineering TCCEE '09, vol.2, pp.585590, Dec.2009.

[3] https://en.wikipedia.org/wiki/PID_controller

[4] http://ctms.engin.umich.edu/CTMS/index.php?example=Introduction&section=ControlPID

[5] https://www.researchgate.net/figure/Vehicle-motion-according-to-the-direction-and-angular-speed-of-the-wheels-6_fig4_233867057/download

[6] https://www.researchgate.net/figure/H-Bridge-motor-drive-circuit-for-one-DC-motor-The-infrared-communication-module-is_fig3_221915427

[7] https://circuitdigest.com/microcontroller-projects/line-follower-robot-using-arduino

[8] https://easyeda.com/GATSAN/DIY_Line_Follower_Robot-8fc88c7a7021462eb5941adca1d404f1

[9] https://tutorial.cytron.io/2015/07/31/line-following-robot-using-lsa08-digital-mode/

[10] https://www.elektroindonesia.com/elektro

LAMPIRAN

Program AVR

Penjelasan di bagian ini akan diberikan program untuk pembuatan Trackless AGV untuk 2 roda penggerak dan 4 roda penggerak, sehingga akan menemukan perbedaan yang jelas antara keduanya, disarankan untuk diawali pembuatan Trackless AGV yang 2 roda untuk menghindari kerumitan dan ketidakpahaman akan program PID dan alur eksekusinya.

```
/*************************************************************
  PROGRAM UTAMA Trackless AGV
*************************************************************/
```

Program untuk mikrokontroler yang mendapatkan data LPS

```
#include <Pozyx.h>
#include <Pozyx_definitions.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd = LiquidCrystal_I2C(0x27, 16, 2);
uint16_t remote_id = 0x6811;
bool remote = false;
uint32_t last_millis;

#include <HCSR04.h>

UltraSonicDistanceSensor ultrasonik(8, 9);

boolean use_processing = false;

long simpanx;
long simpany;
long simpanx1;
long simpany1;

//set up nilai satelit pada posisi x dan y serta ketinggian jika diperlukan
long xanchors = 2800;
long yanchors = 4500;
long h = 900;

unsigned long a;
unsigned long b;
unsigned long c;
unsigned long d;

//set up satelit pada posisi x dan y serta ketinggian jika diperlukan sesuai alamat
satelitnya
const uint8_t num_anchors = 4;
uint16_t anchors[num_anchors] = {0x0D66, 0x0D5E,   0x6811,   0x6855};
int32_t anchors_x[num_anchors] = {0,     0,        xanchors, 600};
int32_t anchors_y[num_anchors] = {0,     0,        0,        yanchors};
int32_t heights[num_anchors] = {  0,     h,        h,        h};
```

```
uint8_t algorithm = POZYX_POS_ALG_UWB_ONLY;
uint8_t dimension = POZYX_3D;
int32_t height = h;

const int numReadings = 5;

long readings[numReadings];
long readIndex = 0;
long total = 0;
long average = 0;

const int numReadings1 = 5;

long readings1[numReadings1];
long readIndex1 = 0;
long total1 = 0;
long average1 = 0;

const int numReadings2 = 5;

long readings2[numReadings2];
long readIndex2 = 0;
long total2 = 0;
long average2 = 0;

const int numReadings3 = 5;

long readings3[numReadings3];
long readIndex3 = 0;
long total3 = 0;
long average3 = 0;

const int numReadings4 = 5;

long readings4[numReadings4];
long readIndex4 = 0;
long total4 = 0;
long average4 = 0;

const int numReadings5 = 5;

long readings5[numReadings5];
long readIndex5 = 0;
long total5 = 0;
long average5 = 0;

void setup() {
  Serial.begin(115200);


  lcd.begin();
  lcd.backlight();

  if (Pozyx.begin() == POZYX_FAILURE) {
    Serial.println(F("ERROR: Unable to connect to POZYX shield"));
    Serial.println(F("Reset required"));
    delay(100);
    abort();
  }

  if (Pozyx.begin(false, MODE_INTERRUPT, POZYX_INT_MASK_IMU) == POZYX_FAILURE) {
    Serial.println("ERROR: Unable to connect to POZYX shield");
    Serial.println("Reset required");
    delay(100);
    abort();
  }

  if (!remote) {
```

```
    remote_id = NULL;
  }

  Pozyx.clearDevices(remote_id);
  setAnchorsManual();
  Pozyx.setPositionAlgorithm(algorithm, dimension, remote_id);

  printCalibrationResult();
  delay(2000);

  Serial.println(F("Starting positioning: "));

  for (int thisReading = 0; thisReading < numReadings; thisReading++) {
    readings[thisReading] = 0;
  }
  for (int thisReading1 = 0; thisReading1 < numReadings1; thisReading1++) {
    readings1[thisReading1] = 0;
  }
  for (int thisReading2 = 0; thisReading2 < numReadings2; thisReading2++) {
    readings2[thisReading2] = 0;
  }
  for (int thisReading3 = 0; thisReading3 < numReadings3; thisReading3++) {
    readings3[thisReading3] = 0;
  }
  for (int thisReading4 = 0; thisReading4 < numReadings4; thisReading4++) {
    readings4[thisReading4] = 0;
  }

  for (int thisReading5 = 0; thisReading5 < numReadings5; thisReading5++) {
    readings5[thisReading5] = 0;
  }


  last_millis = millis();
  delay(10);

}

void loop() {

  sensor_raw_t sensor_raw;
  uint8_t calibration_status = 0;
  int dt;
  int status;
  if (remote) {
    status = Pozyx.getRawSensorData(&sensor_raw, remote_id);
    status &= Pozyx.getCalibrationStatus(&calibration_status, remote_id);
    if (status != POZYX_SUCCESS) {
      return;
    }
  } else {
    if (Pozyx.waitForFlag(POZYX_INT_STATUS_IMU, 10) == POZYX_SUCCESS) {
      Pozyx.getRawSensorData(&sensor_raw);
      Pozyx.getCalibrationStatus(&calibration_status);
    } else {
      uint8_t interrupt_status = 0;
      Pozyx.getInterruptStatus(&interrupt_status);
      return;
    }
  }

  a = millis();
  c = millis();

  coordinates_t position;
  if (remote) {
    status = Pozyx.doRemotePositioning(remote_id, &position, dimension, height,
algorithm);
```

```
  } else {
    status = Pozyx.doPositioning(&position, dimension, height, algorithm);
  }

  if (status == POZYX_SUCCESS) {
    printRawSensorData(sensor_raw);
    printCoordinates(position);

  } else {
    printErrorCode("positioning");
  }
}

void printCoordinates(coordinates_t coor) {
  uint16_t network_id = remote_id;
  if (network_id == NULL) {
    network_id = 0;
  }
  if (!use_processing) {



    total = total - readings[readIndex];
    readings[readIndex] = coor.x;
    total = total + readings[readIndex];
    readIndex = readIndex + 1;
    if (readIndex >= numReadings) {
      readIndex = 0;
    }
    average = total / numReadings;

    if (((average + 500) > coor.x) and ((average - 500) < coor.x)) {
      simpanx = coor.x;
    }

    total1 = total1 - readings1[readIndex1];
    readings1[readIndex1] = simpanx;
    total1 = total1 + readings1[readIndex1];
    readIndex1 = readIndex1 + 1;
    if (readIndex1 >= numReadings1) {
      readIndex1 = 0;
    }
    average1 = total1 / numReadings1;

    if (((average1 + 500) > simpanx) and ((average1 - 500) < simpanx)) {
      simpanx1 = simpanx;
    }


    total4 = total4 - readings4[readIndex4];
    readings4[readIndex4] = simpanx1;
    total4 = total4 + readings4[readIndex4];
    readIndex4 = readIndex4 + 1;
    if (readIndex4 >= numReadings4) {
      readIndex4 = 0;
    }
    average4 = total4 / numReadings4;


    Serial.print("x=");
    Serial.print(average4);
    Serial.print(";");


    total2 = total2 - readings2[readIndex2];
    readings2[readIndex2] = coor.y;
    total2 = total2 + readings2[readIndex2];
    readIndex2 = readIndex2 + 1;
```

```
    if (readIndex2 >= numReadings2) {
      readIndex2 = 0;
    }
    average2 = total2 / numReadings2;

    if (((average2 + 500) > coor.y) and ((average2 - 500) < coor.y)) {
      simpany = coor.y;
    }


    total3 = total3 - readings3[readIndex3];
    readings3[readIndex3] = simpany;
    total3 = total3 + readings3[readIndex3];
    readIndex3 = readIndex3 + 1;
    if (readIndex3 >= numReadings3) {
      readIndex3 = 0;
    }
    average3 = total3 / numReadings3;

    if (((average3 + 500) > simpany) and ((average3 - 500) < simpany)) {
      simpany1 = simpany;
    }


    total5 = total5 - readings5[readIndex5];
    readings5[readIndex5] = simpany1;
    total5 = total5 + readings5[readIndex5];
    readIndex5 = readIndex5 + 1;
    if (readIndex5 >= numReadings5) {
      readIndex5 = 0;
    }
    average5 = total5 / numReadings5;

    Serial.print("y=");
    Serial.print(average5);
    Serial.print(";");
    Serial.print("u=");
    Serial.print(ultrasonik.measureDistanceCm());
    Serial.print(";");



    Serial.println();


  } else {
    Serial.print("POS,0x");
    Serial.print(network_id, HEX);
    Serial.print(",");
    Serial.print(coor.x);
    Serial.print(",");
    Serial.print(coor.y);
    Serial.print(",");
    Serial.println(coor.z);
  }
}

void printErrorCode(String operation) {
  uint8_t error_code;
  if (remote_id == NULL) {
    Pozyx.getErrorCode(&error_code);
    Serial.print("ERROR ");
    Serial.print(operation);
    Serial.print(", local error code: 0x");
    Serial.println(error_code, HEX);
    return;
  }
  int status = Pozyx.getErrorCode(&error_code, remote_id);
```

```
  if (status == POZYX_SUCCESS) {
    Serial.print("ERROR ");
    Serial.print(operation);
    Serial.print(" on ID 0x");
    Serial.print(remote_id, HEX);
    Serial.print(", error code: 0x");
    Serial.println(error_code, HEX);
  } else {
    Pozyx.getErrorCode(&error_code);
    Serial.print("ERROR ");
    Serial.print(operation);
    Serial.print(", couldn't retrieve remote error code, local error: 0x");
    Serial.println(error_code, HEX);
  }
}

void printCalibrationResult() {
  uint8_t list_size;
  int status;

  status = Pozyx.getDeviceListSize(&list_size, remote_id);


  if (list_size == 0) {
    printErrorCode("configuration");
    return;
  }

  uint16_t device_ids[list_size];
  status &= Pozyx.getDeviceIds(device_ids, list_size, remote_id);

  Serial.println(F("Calibration result:"));
  Serial.print(F("Anchors found: "));
  Serial.println(list_size);

  coordinates_t anchor_coor;
  for (int i = 0; i < list_size; i++)
  {
    Serial.print("ANCHOR,");
    Serial.print("0x");
    Serial.print(device_ids[i], HEX);
    Serial.print(",");
    Pozyx.getDeviceCoordinates(device_ids[i], &anchor_coor, remote_id);
    Serial.print(anchor_coor.x);
    Serial.print(",");
    Serial.print(anchor_coor.y);
    Serial.print(",");
    Serial.println(anchor_coor.z);
  }
}

void setAnchorsManual() {
  for (int i = 0; i < num_anchors; i++) {
    device_coordinates_t anchor;
    anchor.network_id = anchors[i];
    anchor.flag = 0x1;
    anchor.pos.x = anchors_x[i];
    anchor.pos.y = anchors_y[i];
    anchor.pos.z = heights[i];
    Pozyx.addDevice(anchor, remote_id);
  }
  if (num_anchors > 4) {
    Pozyx.setSelectionOfAnchors(POZYX_ANCHOR_SEL_AUTO, num_anchors, remote_id);
  }

}

void printRawSensorData(sensor_raw_t sensor_raw) {
```

```
  Serial.print("t=");
  Serial.print(map(sensor_raw.euler_angles[0], 0, 5759, 5759, 0));
  Serial.print(";");
  if (c - d >= 200) {
    d = c;
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("x=");
    lcd.setCursor(2, 0);
    lcd.print(average4);
    lcd.setCursor(8, 0);
    lcd.print("y=");
    lcd.setCursor(10, 0);
    lcd.print(average5);
    lcd.setCursor(0, 1);
    lcd.print(map(map(sensor_raw.euler_angles[0], 0, 5759, 5759, 0), 0, 5759, 0,
359));
  }
}

void printCalibrationStatus(uint8_t calibration_status) {
  Serial.print(calibration_status & 0x03);
  Serial.print(",");
  Serial.print((calibration_status & 0x0C) >> 2);
  Serial.print(",");
  Serial.print((calibration_status & 0x30) >> 4);
  Serial.print(",");
  Serial.print((calibration_status & 0xC0) >> 6);
}
```

Program untuk mikrokontroler arduino mega AGV

```
int ultrasonik;
int ultrasonik1;
int ultrasonik2;
int ultrasonik3;

int rodakananmaju = 5;
int rodakananmundur = 2;
int rodakirimaju = 6;
int rodakirimundur = 7;

int pushbutton = 9;

int start1 = 0;
int penukar = 1;

String data1;
String strMasukan;
String strMasukan1;
String data2;
int dataarah;
int arah;
int arah1;
long posisix;
long posisiy;
long posisixdata;
long posisiydata;

long posisiakhirx;
long posisiakhiry;
long arahakhir;

int pengunci = 0;
int pengunci1 = 0;
int pengunci2 = 0;
int pengunci3 = 0;

int dataolaharah;
int datakirimarah;

int timur;
int barat;

int j;
int k;
int l;

int kananmaju;
int kananmundur;
int kirimaju;
int kirimundur;

long waktu;

unsigned long waktusebelum;
unsigned long waktusesudah;

unsigned long waktusebelum1;
unsigned long waktusesudah1;

unsigned long waktusebelum2;
unsigned long waktusesudah2;

unsigned long waktusebelum3;
unsigned long waktusesudah3;
```

```
unsigned long waktusebelum4;
unsigned long waktusesudah4;

unsigned long waktusebelum5;
unsigned long waktusesudah5;

unsigned long waktusebelum6;
unsigned long waktusesudah6;

int counter;
int counter1 = 0;

#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include "Keypad.h"

LiquidCrystal_I2C lcd(0x27, 16, 2); // set the LCD address to 0x27 for a 16 chars
and 2 line display
const byte ROWS = 4;
const byte COLS = 4;

char hexaKeys[ROWS][COLS] = {
  {'5', '2', '0', '8'},
  {'4', '1', '*', '7'},
  {'B', 'A', 'D', 'C'},
  {'6', '3', '#', '9'}
};

byte rowPins[ROWS] = {22, 23, 24, 25};
byte colPins[COLS] = {26, 27, 28, 29};
String str[100];
int sistem = 0;
int counter3 = 0;

Keypad customKeypad = Keypad( makeKeymap(hexaKeys)
                              , rowPins, colPins, ROWS, COLS);

String dtTombol = "";

int lemarix[100];
int lemariy[100];

unsigned long sesudah;
unsigned long sebelum;

int simpanpinjam1;
int simpanpinjam2;

int selesai;


int sudut;
int sudut1;
int ubahsudut;
int ubahsudut1;
int suduthitung;

void setup() {
  pinMode(rodakananmaju, OUTPUT);
  pinMode(rodakananmundur, OUTPUT);
  pinMode(rodakirimaju, OUTPUT);
  pinMode(rodakirimundur, OUTPUT);
  pinMode(pushbutton, INPUT_PULLUP);
  Serial.begin(115200);
  Serial1.begin(115200);
  Serial2.begin(115200);
  lcd.begin();
```

```
  lcd.backlight();
  lcd.setCursor(2, 0);
  lcd.print("Tekan Keypad");
  posisiakhirx = 1000;
  posisiakhiry = 0;
  selesai = 7;
  analogWrite(rodakananmaju, 0);
  analogWrite(rodakananmundur, 0);
  analogWrite(rodakirimaju, 0);
  analogWrite(rodakananmundur, 0);
}

void loop() {
//sistem pushbutton untuk aktivasi AGV secara fisik
  if ((digitalRead(pushbutton) == LOW) and (penukar == 1)) {
    start1++;
    penukar = 0;
  } if ((digitalRead(pushbutton) == HIGH) and (penukar == 0)) {
    penukar = 1;
    pengunci = 0;
    pengunci1 = 0;
    counter = 0;
  }

//program untuk kalibrasi arah sesuai sumbu x+ / 0 derajat / timur
  waktu = millis();
  if (pengunci2 == 0) {
    if (waktu > 5000) {
      pengunci2 = 1;
      posisiakhirx = posisix;
      posisiakhiry = posisiy;
      timur = map(dataarah, 0, 5759, 0, 360);
    }
  }

//program untuk mendapatkan parameter posisi, arah, dan sensor safety berhenti
berupa ultrasonik
  if (Serial1.available())
  {
    char masukan = Serial1.read();
    strMasukan = strMasukan + masukan;
    if (masukan == 10 or masukan == 13)
    {
      int ix1 = strMasukan.indexOf("t=");
      int ix2 = strMasukan.indexOf(";");
      String data1 = strMasukan.substring(ix1 + 2, ix2);
      if (data1 != "") {
        dataarah = data1.toInt();
      }
      int ix3 = strMasukan.indexOf("x=", ix2);
      int ix4 = strMasukan.indexOf(";" , ix2 + 1);
      String data2 = strMasukan.substring(ix3 + 2, ix4);
      if (data2 != "") {
        posisix = data2.toInt();
      }
      int ix5 = strMasukan.indexOf("y=", ix4);
      int ix6 = strMasukan.indexOf(";" , ix4 + 1);
      String data3 = strMasukan.substring(ix5 + 2, ix6);
      if (data3 != "") {
        posisiy = data3.toInt();
      }
      int ix7 = strMasukan.indexOf("u=", ix6);
      int ix8 = strMasukan.indexOf(";", ix6 + 1);
      String data4 = strMasukan.substring(ix7 + 2, ix8);
      if (data4 != "") {
        ultrasonik = data4.toInt();
        if (ultrasonik1 == ultrasonik) {
          waktusebelum6 = millis();
```

```
          if (waktusebelum6 - waktusesudah6 >= 1000) {
            waktusesudah6 = waktusebelum6;
            ultrasonik1 = 100;
          }
        }
        if (ultrasonik > 1) {
          ultrasonik1 = ultrasonik;
        }
      }
      strMasukan = "";
    }
  }

//program untuk mendapatkan nilai kontrol dari aplikasi windows seperti letak akhir
dan kendali geraknya
  if (Serial2.available())
  {
    char masukan1 = Serial2.read();
    strMasukan1 = strMasukan1 + masukan1;
    if (masukan1 == 10 or masukan1 == 13)
    {
      int ix11 = strMasukan1.indexOf("X=");
      int ix21 = strMasukan1.indexOf(";");
      String data11 = strMasukan1.substring(ix11 + 2, ix21);
      if (data11 != "") {
        simpanpinjam1 = data11.toInt();
        if (simpanpinjam1 != 0) {
          posisiakhirx = simpanpinjam1;
        }
      }
      int ix31 = strMasukan1.indexOf("Y=", ix21);
      int ix41 = strMasukan1.indexOf(";" , ix21 + 1);
      String data21 = strMasukan1.substring(ix31 + 2, ix41);
      if (data21 != "") {
        simpanpinjam2 = data21.toInt();
        if (simpanpinjam2 != 0) {
          posisiakhiry = simpanpinjam2;
        }
      }
      int ix51 = strMasukan1.indexOf("B=");
      int ix61 = strMasukan1.indexOf(";");
      String data31 = strMasukan1.substring(ix51 + 2, ix61);
      if (data31 != "") {
        start1 = data31.toInt();
      }
      strMasukan1 = "";
    }
  }

//perhitungan sistem arah
  arah = map(dataarah, 0, 5759, 0, 359);
  waktusebelum2 = millis();
  if (waktusebelum2 - waktusesudah2 >= 500) {
    waktusesudah2 = waktusebelum2;
    sudut = (atan2((posisiakhiry - posisiy), (posisiakhirx - posisix)) * (180 /
PI));
  }
  if (sudut < 0) {
    sudut1 = 360 + sudut;
  } else {
    sudut1 = sudut;
  }
  suduthitung = sudut1 + timur;
  if (suduthitung >= 360) {
    ubahsudut = suduthitung - 360;
  } else {
    ubahsudut = suduthitung;
  }
```

```
  dataolaharah = arah - timur;
  if (dataolaharah < 0) {
    datakirimarah = 360 + dataolaharah;
  } else {
    datakirimarah = dataolaharah;
  }

//data yang dikirim untuk node-MCU data monitoring letak AGV ke aplikasi windows
  sebelum = millis();
  if (sebelum - sesudah >= 200) {
    if (pengunci2 == 1) {
      sesudah = sebelum;
      Serial2.print("t=");
      Serial2.print(datakirimarah);
      Serial2.print(";x=");
      Serial2.print(posisix);
      Serial2.print(";y=");
      Serial2.print(posisiy);
      Serial2.print(";s=");
      Serial2.print(selesai);
      Serial2.print(";");
      Serial2.println();
    }
  }

//program untuk kendali nilai kecepatan menurut pergerakannya
  analogWrite(rodakananmaju, kananmaju);
  analogWrite(rodakananmundur, kananmundur);
  analogWrite(rodakirimaju, kirimaju);
  analogWrite(rodakirimundur, kirimundur);

//program untuk mengendalikan melalui keypad
  char key = customKeypad.getKey();
  if (key) {
    if (key == '#') {
      lcd.clear();
      lcd.setCursor(1, 0);
      dtTombol = "";
      lcd.setCursor(2, 0);
      counter3 = 0;
      lcd.print("Delete");
      for (int i = 0 ; i < 20 ; i++) {
        str[i] = "";
      }
    }
    if (key == 'D') {
      lcd.setCursor(2, 0);
      lcd.print("Kirim Perintah     ");
      for (int i = 0; i < 10; i++) {
        lemarix[i] = str[(i * 2)].toInt();
        lemariy[i] = str[(i * 2) + 1].toInt();
      }
      counter1 = 0;
      pengunci3 = 0;
      selesai = 1;
    }
    if (sistem == 1) {
      if (key == '0' or key == '1' or key == '2' or key == '3' or key == '4' or key
== '5' or key == '6' or key == '7' or key == '8' or key == '9') {
        dtTombol = dtTombol + key;
        lcd.setCursor(0, 1);
        lcd.print(String(counter3) + '=' + dtTombol);
        str[counter3] = str[counter3] + key;
      }
    }
    if (key == 'C') {
      lcd.setCursor(0, 1);
      lcd.print("                                ");
```

```
      str[counter3] = "";
      dtTombol = "";
    }
    if (key == 'B') {
      lcd.clear();
      lcd.setCursor(1, 0);
      dtTombol = "";
      lcd.setCursor(2, 0);
      lcd.print("Data Dimasukan     ");
      sistem = 0;
      Serial.print(counter3);
      Serial.print("=");
      Serial.println(str[counter3]);
      counter3++;
    }
    if (key == 'A') {
      lcd.setCursor(2, 0);
      lcd.print("Sistem Input      ");
      sistem = 1;
      lcd.setCursor(0, 1);
      lcd.print(String(counter3) + '=' + dtTombol);
    }
  }

//program untuk mendapatkan banyak titik melalui keypad
  if (counter1 <= 100) {
    if (selesai == 1 && pengunci3 == 0) {
      pengunci3 = 1;
      if (counter1 < (counter3 / 2)) {
        posisiakhirx = lemarix[counter1];
        posisiakhiry = lemariy[counter1];
      }
      counter1++;
    } else if (selesai == 7 && pengunci3 == 1) {
      pengunci3 = 0;
    }
  }

//program untuk kontrol pergerakan AGV sesuai parameter koordinat x dan y serta
arah
  if (ultrasonik1 <= 40) {
    berhentiagv();
  } else {
    waktusebelum5 = millis();
    if (start1 % 2 == 1) {
      if ((posisiakhirx - 200) <= posisix and posisix <= (posisiakhirx + 200) and
(posisiakhiry - 200) <= posisiy and posisiy <= (posisiakhiry + 200)) {
        berhentiagv();
        waktusebelum5 = millis();
        waktusesudah5 = millis();
        selesai = 1;
      } else {
        if (waktusebelum5 - waktusesudah5 >= 1000) {
          selesai = 7;
          if (((ubahsudut <= 360) and (ubahsudut >= 270)) and ((arah <= 90) and
(arah >= 0))) {
            belokkananagv();
          } else if (((arah <= 360) and (arah >= 270)) and ((ubahsudut <= 90) and
(ubahsudut >= 0))) {
            belokkiriagv();
          } else {
            if ((ubahsudut - 2) <= arah and arah <= (ubahsudut + 2)) {
              majuagv();

            } else if ((ubahsudut - 30) <= arah and (ubahsudut - 2) > arah) {
              kiriagv();
            } else if (arah <= (ubahsudut + 30) and arah > (ubahsudut + 2)) {
              kananagv();
```

```
            } else {
              if ((ubahsudut - 30) > arah) {
                belokkiriagv();
              } else if (arah > (ubahsudut + 30)) {
                belokkananagv();
              }
            }
          }
        }
      }
    } else {
      berhentiagv();
    }
  }
}

void berhentiagv() {
  kananmaju = 0;
  kananmundur = 0;
  kirimaju = 0;
  kirimundur = 0;
}

void majuagv() {
  kananmundur = 0;
  kirimundur = 0;
  if (kananmaju == 0) {
    kananmaju = kirimaju;
  } else if (kirimaju == 0) {
    kirimaju = kananmaju;
  } else {
    kananmaju = kirimaju;
  }
  waktusebelum = millis();
  if (waktusebelum - waktusesudah >= 50) {
    waktusesudah = waktusebelum;
    kirimaju++;
  }
}

void kananagv() {
  kananmundur = 0;
  kirimundur = 0;
  if (kananmaju == 0) {
    kananmaju = kirimaju;
  } else if (kirimaju == 0) {
    kirimaju = kananmaju;
  }
  waktusebelum3 = millis();
  if (waktusebelum3 - waktusesudah3 >= 50) {
    waktusesudah3 = waktusebelum3;
    kirimaju++;
  }
}

void kiriagv() {
  kananmundur = 0;
  kirimundur = 0;
  if (kananmaju == 0) {
    kananmaju = kirimaju;
  } else if (kirimaju == 0) {
    kirimaju = kananmaju;
  }
  waktusebelum4 = millis();
  if (waktusebelum4 - waktusesudah4 >= 50) {
    waktusesudah4 = waktusebelum4;
    kananmaju++;
  }
```

```
}

void belokkananagv() {
  kananmaju = 0;
  kananmundur = 40;
  kirimaju = 100;
  kirimundur = 0;
}

void belokkiriagv() {
  kananmaju = 100;
  kananmundur = 0;
  kirimaju = 0;
  kirimundur = 40;
}
```

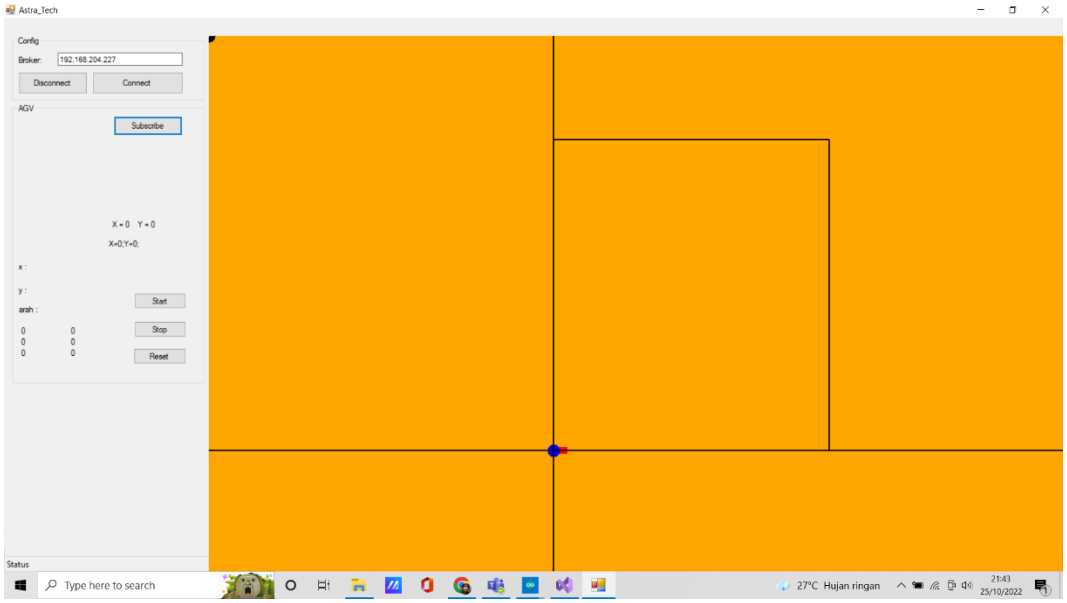
Program untuk mikrokontroler Node-MCU

```
#include <ESP8266WiFi.h>
#include <PubSubClient.h>
String str;
String data1;
String data2;
String data3;
int dataarah;
long posisix;
long posisiy;

int a;
int b;
int c;
int d;

String strMasukan;
int selesai;
long datainput;

String kata;

long datamonitor;

//set up untuk alamat WIFI dan broker MQTT yang dipakai
const char* ssid = "BEJOTECH";
const char* password = "";
const char* mqtt_server = "192.168.149.227";

WiFiClient espClient;
PubSubClient client(espClient);
#define MSG_BUFFER_SIZE  (50)
char msg[MSG_BUFFER_SIZE];

unsigned long sebelum;
unsigned long sesudah;
unsigned long sebelum1;
unsigned long sesudah1;
unsigned long sebelum2;
unsigned long sesudah2;
unsigned long sebelum3;
unsigned long sesudah3;
int datalama;

void setup_wifi() {

  delay(10);
  //  Serial.println();
  //  Serial.print("Connecting to ");
  //  Serial.println(ssid);

  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    //    Serial.print(".");
  }

  randomSeed(micros());

  //  Serial.println("");
  //  Serial.println("WiFi connected");
  //  Serial.println("IP address: ");
  //  Serial.println(WiFi.localIP());
```

```
}

//program untuk menerima data yang dikirimkan dari program windows untuk mengontrol
AGV
void callback(char* topic, byte* payload, unsigned int length) {
  for (int i = 0; i < length; i++) {
    kata = kata + (char)payload[i];
  String kalimat = topic;
  if (kalimat == "pozyxinput") {
    Serial.println(kata);
    //  datainput = kata.toInt();
  }
  kata = "";
}

void reconnect() {
  while (!client.connected()) {
    //    Serial.print("Attempting MQTT connection...");
    String clientId = "ESP8266Client-";
    clientId += String(random(0xffff), HEX);
    if (client.connect(clientId.c_str())) {
      //      Serial.println("connected");
      client.subscribe("pozyxinput");
    } else {
      //      Serial.print("failed, rc=");
      //      Serial.print(client.state());
      //      Serial.println(" try again in 5 seconds");
      delay(5000);
    }
  }
}

void setup() {
  Serial.begin(115200);
  setup_wifi();
  client.setServer(mqtt_server, 1883);
  client.setCallback(callback);
  d = 1;
}

void loop() {
//program untuk menerima data monitor lokasi AGV secara serial
  if (Serial.available())
  {
    char masukan = Serial.read();
    strMasukan = strMasukan + masukan;
    if (masukan == 10 or masukan == 13)
    {
      int ix1 = strMasukan.indexOf("t=");
      int ix2 = strMasukan.indexOf(";");
      String data1 = strMasukan.substring(ix1 + 2, ix2);
      if (data1 != "") {
        dataarah = data1.toInt();
      }
      int ix3 = strMasukan.indexOf("x=", ix2);
      int ix4 = strMasukan.indexOf(";" , ix2 + 1);
      String data2 = strMasukan.substring(ix3 + 2, ix4);
      if (data2 != "") {
        posisix = data2.toInt();
      }
      int ix5 = strMasukan.indexOf("y=", ix4);
      int ix6 = strMasukan.indexOf(";" , ix4 + 1);
      String data3 = strMasukan.substring(ix5 + 2, ix6);
      if (data3 != "") {
        posisiy = data3.toInt();
      }
      int ix7 = strMasukan.indexOf("s=", ix6);
      int ix8 = strMasukan.indexOf(";" , ix6 + 1);
```

```
      String data4 = strMasukan.substring(ix7 + 2, ix8);
      if (data4 != "") {
        selesai = data4.toInt();
      }
      strMasukan = "";
    }

  }

  if (!client.connected()) {
    reconnect();
  }
  client.loop();

  //sistem pengiriman data monitoring AGV ke aplikasi windows
  sesudah = millis();
  if (sesudah - sebelum >= 100) {
    sebelum = sesudah;
    if (d == 1) {
      snprintf (msg, MSG_BUFFER_SIZE, "%ld", posisix);
      client.publish("pozyxmonitorx", msg);
      a = 1;
      sesudah1 = millis();
      sebelum1 = millis();
      d = 0;
    }
  }
  sesudah1 = millis();
  if (sesudah1 - sebelum1 >= 100) {
    sebelum1 = sesudah1;
    if (a == 1) {
      snprintf (msg, MSG_BUFFER_SIZE, "%ld", posisiy);
      client.publish("pozyxmonitory", msg);
      b = 1;
      sesudah2 = millis();
      sebelum2 = millis();
      a = 0;
    }
  }
  sesudah2 = millis();
  if (sesudah2 - sebelum2 >= 100) {
    sebelum2 = sesudah2;
    if (b == 1) {
      snprintf (msg, MSG_BUFFER_SIZE, "%ld", dataarah);
      client.publish("pozyxmonitorarah", msg);
      c = 1;
      sesudah3 = millis();
      sebelum3 = millis();
      b = 0;
    }
  }
  sesudah3 = millis();
  if (sesudah3 - sebelum3 >= 100) {
    sebelum3 = sesudah3;
    if (c == 1) {
      snprintf (msg, MSG_BUFFER_SIZE, "%ld", selesai);
      client.publish("pozyxmonitorselesai", msg);
      d = 1;
      sesudah = millis();
      sebelum = millis();
      c = 0;
    }
  }

}
```

Program untuk aplikasi windows




```
Imports uPLibrary.Networking.M2Mqtt
Imports uPLibrary.Networking.M2Mqtt.Messages
Imports System.Text
Imports System.Threading


Public Class Form1

    Dim client As MqttClient
    Dim Msg As StringBuilder = New StringBuilder(4096)

    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
        If (TextBox1.Text.Length <> 0) Then


            Try
                client = New MqttClient(TextBox1.Text)

                Dim clientId As String = Guid.NewGuid().ToString()

                AddHandler client.MqttMsgPublishReceived, AddressOf
Client_MqttMsgPublishReceived
                AddHandler client.ConnectionClosed, AddressOf Client_Disconnect

                client.Connect(clientId)

                If client.IsConnected Then
                    ComboBox1.SelectedIndex = 0
                    ComboBox2.SelectedIndex = 0

                    ToolStripStatusLabel1.Text = "Connected to " + "'" +
TextBox1.Text + "'"
                Else
                    ToolStripStatusLabel1.Text = "Disconnected"
                End If

            Catch ex As Exception
```

```
                ToolStripStatusLabel1.Text = "Error"
                MsgBox(ex.Message(), MsgBoxStyle.Critical)
            End Try
        Else
            ToolStripStatusLabel1.Text = "Please enter a valid Broker address "

        End If
    End Sub

    Private Sub Button4_Click(sender As Object, e As EventArgs) Handles
Button4.Click
        If (client IsNot Nothing AndAlso client.IsConnected()) Then
            client.Disconnect()
        Else
            ToolStripStatusLabel1.Text = "Error"
        End If
    End Sub
    Private Sub Client_Disconnect(sender As Object, e As EventArgs)
        ToolStripStatusLabel1.Text = "Connection Lost"
    End Sub


    Private Sub Client_MqttMsgPublishReceived(ByVal sender As Object, ByVal e As
MqttMsgPublishEventArgs)


        Msg.Append(e.Topic.ToString() + Encoding.Default.GetString(e.Message))
        Msg.AppendLine()

        SetText(Msg.ToString)

    End Sub
    Dim simpan1 As Long
    Dim simpan2 As Long
    Dim simpan3 As Long
    Dim simpan4 As Long

    Delegate Sub SetTextCallback(newString As String)
    Private Sub SetText(ByVal newString As String)
        ' Calling from another thread? -> Use delegate
        If Me.RichTextBox2.InvokeRequired Then
            Dim d As New SetTextCallback(AddressOf SetText)
            ' Execute delegate in the UI thread, pass args as an array
            Me.Invoke(d, New Object() {newString})
        Else ' Same thread, assign string to the textbox
            str = ""
            Msg.Clear()
            RichTextBox2.Clear()
            Me.RichTextBox2.Text = newString
            str = newString

            If (str.IndexOf("pozyxmonitorx") = 0) Then
                simpan1 = CInt(str.Substring(Len("pozyxmonitorx")))
                If simpan1 <> 0 Then
                    posisix = simpan1
                    Label10.Text = posisix
                End If

            ElseIf (str.IndexOf("pozyxmonitory") = 0) Then
                simpan2 = CInt(str.Substring(Len("pozyxmonitory")))
                If simpan2 <> 0 Then
                    posisiy = simpan2
                    Label11.Text = posisiy
                End If
            ElseIf (str.IndexOf("pozyxmonitorarah") = 0) Then
                simpan3 = CInt(str.Substring(Len("pozyxmonitorarah")))
```

```
                If simpan3 <> 0 Then
                    arah = simpan3
                    Label12.Text = arah
                End If
            ElseIf (str.IndexOf("pozyxmonitorselesai") = 0) Then
                simpan4 = CInt(str.Substring(Len("pozyxmonitorselesai")))
                If simpan4 <> 0 Then
                    selesai = simpan4
                End If
            End If
                Msg.Clear()
            RichTextBox2.Clear()
            pen3 = New Pen(Color.Blue, 10)
            pen4 = New Pen(Color.Red, 10)
            pen1 = New Pen(Color.Black, 10)
            surface = Panel1.CreateGraphics()
            rectangle2 = New Rectangle(xakhir - 4, yakhir - 4, 8, 8)
            rectangle3 = New Rectangle((posisix / 10 + 500) - 4, (600 - (posisiy /
10)) - 4, 8, 8)
            surface.FillRectangle(brush1, rectangle1)
            pen2 = New Pen(Color.Black, 2)
            surface.DrawLine(pen2, 500, 0, 500, 900)
            surface.DrawLine(pen2, 0, 600, 1900, 600)
            surface.DrawLine(pen2, 500 + 400, 150, 500 + 400, 600)
            surface.DrawLine(pen2, 500, 600 - 450, 900, 600 - 450)
            arahx = 20 * Math.Cos(Math.PI * arah / 180.0)
            arahy = 20 * Math.Sin(Math.PI * arah / 180.0)
            posisixarah = (posisix / 10 + 500)
            posisiyarah = (600 - (posisiy / 10))
            surface.DrawLine(pen4, posisixarah, posisiyarah, posisixarah + arahx,
posisiyarah - arahy)
            surface.DrawEllipse(pen1, rectangle2)
            surface.DrawEllipse(pen3, rectangle3)
        End If
    End Sub
    Dim posisixarah As Long = 0
    Dim posisiyarah As Long = 0
    Dim posisix As Integer = 0
    Dim posisiy As Long = 0
    Dim arah As Long = 0
    Dim selesai As Long = 0
    Dim arahx As Long = 0
    Dim arahy As Long = 0
    Dim str As String

    Private Sub Button2_Click(sender As Object, e As EventArgs) Handles
Button2.Click
        If (client IsNot Nothing AndAlso client.IsConnected()) Then

            Try
                Dim Topic() As String = {"pozyxmonitorx"}
                Dim Qos() As Byte = {ComboBox1.SelectedIndex}
                client.Subscribe(Topic, Qos)
                Dim Topic1() As String = {"pozyxmonitory"}
                Dim Qos1() As Byte = {ComboBox1.SelectedIndex}
                client.Subscribe(Topic1, Qos1)
                Dim Topic2() As String = {"pozyxmonitorarah"}
                Dim Qos2() As Byte = {ComboBox1.SelectedIndex}
                client.Subscribe(Topic2, Qos2)
                Dim Topic3() As String = {"pozyxmonitorselesai"}
                Dim Qos3() As Byte = {ComboBox1.SelectedIndex}
                client.Subscribe(Topic3, Qos3)

            Catch ex As Exception
                ToolStripStatusLabel1.Text = "Error"
                MsgBox(ex.Message, MsgBoxStyle.Critical)
            End Try
        Else
```

```
            ToolStripStatusLabel1.Text = "Disconnected !! subscription procedure is
not valid"
        End If
    End Sub

    Private Sub Button3_Click(sender As Object, e As EventArgs) Handles
Button3.Click
        If (client IsNot Nothing AndAlso client.IsConnected()) Then
            Try
                Dim Qos As Byte = ComboBox2.SelectedIndex
                client.Publish("pozyxinput",
Encoding.Default.GetBytes(RichTextBox1.Text), Qos, False)
                ToolStripStatusLabel1.Text = "Publish to " + "{" + "pozyxinput" +
"}"
            Catch ex As Exception
                ToolStripStatusLabel1.Text = "Error"
                MsgBox(ex.Message, MsgBoxStyle.Critical)
            End Try


        Else
            ToolStripStatusLabel1.Text = "Disconnected !! Publish procedure is not
valid"
        End If


    End Sub

    Private Sub Button5_Click(sender As Object, e As EventArgs) Handles
Button5.Click
        Msg.Clear()
        RichTextBox2.Clear()
    End Sub

    Private Sub Panel1_MouseMove(sender As Object, e As MouseEventArgs) Handles
Panel1.MouseMove
        Label8.Text = "X = " & (e.X - 500) * 10 & "    Y = " & (600 - e.Y) * 10
        pen2 = New Pen(Color.Black, 2)
        surface = Panel1.CreateGraphics()
        surface.DrawLine(pen2, 500, 0, 500, 900)
        surface.DrawLine(pen2, 0, 600, 1900, 600)
        surface.DrawLine(pen2, 500 + 400, 150, 500 + 400, 600)
        surface.DrawLine(pen2, 500, 600 - 450, 900, 600 - 450)

    End Sub
    Dim xakhir As Long
    Dim yakhir As Long
    Dim titikakhir(100, 1) As Long
    Dim akhir(100, 1) As Long
    Dim urutan As ULong

    Dim kunci As Integer = 0
    Private Sub Panel1_MouseClick(sender As Object, e As MouseEventArgs) Handles
Panel1.MouseClick

        titikakhir(urutan, 0) = (e.X - 500) * 10
        titikakhir(urutan, 1) = (600 - e.Y) * 10
        akhir(urutan, 0) = e.X
        akhir(urutan, 1) = e.Y
        urutan = urutan + 1

        If kunci = 0 Then

            Try
                Dim Qos As Byte = ComboBox2.SelectedIndex
                client.Publish("pozyxinput", Encoding.Default.GetBytes("X=" &
titikakhir(0, 0) & ";Y=" & titikakhir(0, 1) & ";"), Qos, False)
```

```
                ToolStripStatusLabel1.Text = "Publish to " + "{" + "pozyxinput" +
"}"
            Catch ex As Exception
                ToolStripStatusLabel1.Text = "Error"
                MsgBox(ex.Message, MsgBoxStyle.Critical)
            End Try
            kunci = 1
        End If

        xakhir = e.X
        yakhir = e.Y
        pen1 = New Pen(Color.Black, 10)
        surface = Panel1.CreateGraphics()
        rectangle2 = New Rectangle(xakhir - 4, yakhir - 4, 8, 8)
        surface.FillRectangle(brush1, rectangle1)
        pen2 = New Pen(Color.Black, 2)
        surface = Panel1.CreateGraphics()
        surface.DrawLine(pen2, 500, 0, 500, 900)
        surface.DrawLine(pen2, 0, 600, 1900, 600)
        surface.DrawEllipse(pen1, rectangle2)


        surface.DrawLine(pen2, 500, 0, 500, 900)
        surface.DrawLine(pen2, 0, 600, 1900, 600)
        pen3 = New Pen(Color.Blue, 10)
        pen4 = New Pen(Color.Red, 10)
        pen1 = New Pen(Color.Black, 10)
        surface = Panel1.CreateGraphics()
        rectangle2 = New Rectangle(xakhir - 4, yakhir - 4, 8, 8)
        rectangle3 = New Rectangle((posisix / 10 + 500) - 4, (600 - (posisiy / 10))
- 4, 8, 8)
        surface.FillRectangle(brush1, rectangle1)
        pen2 = New Pen(Color.Black, 2)
        surface.DrawLine(pen2, 500, 0, 500, 900)
        surface.DrawLine(pen2, 0, 600, 1900, 600)
        arahx = 20 * Math.Cos(Math.PI * arah / 180.0)
        arahy = 20 * Math.Sin(Math.PI * arah / 180.0)
        posisixarah = (posisix / 10 + 500)
        posisiyarah = (600 - (posisiy / 10))
        surface.DrawLine(pen4, posisixarah, posisiyarah, posisixarah + arahx,
posisiyarah - arahy)
        surface.DrawEllipse(pen1, rectangle2)
        surface.DrawEllipse(pen3, rectangle3)


        surface.DrawLine(pen2, 500, 0, 500, 900)
        surface.DrawLine(pen2, 0, 600, 1900, 600)
        surface.DrawLine(pen2, 500 + 400, 150, 500 + 400, 600)
        surface.DrawLine(pen2, 500, 600 - 450, 900, 600 - 450)
    End Sub
    Dim rectangle2 As Rectangle
    Dim rectangle3 As Rectangle
    Dim surface As Graphics
    Dim pen1 As Pen
    Dim pen2 As Pen
    Dim pen3 As Pen
    Dim pen4 As Pen
    Dim rectangle1 As Rectangle = New Rectangle(0, 0, 1593, 926)
    Dim brush1 As Brush = Brushes.Orange
    Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load

        Dim surface As Graphics = Panel1.CreateGraphics()
        surface.FillRectangle(brush1, rectangle1)
        pen1 = New Pen(Color.Black, 2)
        surface = Panel1.CreateGraphics()
        surface.DrawLine(pen1, 500, 0, 500, 900)
        Timer1.Enabled = True
    End Sub
```

```
    Dim kunci1 As Integer = 0
    Dim kunci2 As Integer = 0
    Private Sub Timer2_Tick(sender As Object, e As EventArgs) Handles Timer2.Tick
        Timer1.Enabled = True
        Timer2.Enabled = False

        If selesai = 1 Then
            kunci1 = 1
        Else
            kunci1 = 0
        End If

        If kunci1 = 1 And kunci2 = 0 Then

            For c = 0 To urutan - 1
                titikakhir(c, 0) = titikakhir(c + 1, 0)
                titikakhir(c, 1) = titikakhir(c + 1, 1)
            Next

            Label9.Text = "X=" & titikakhir(0, 0) & ";Y=" & titikakhir(0, 1) & ";"

            Try
                Dim Qos As Byte = ComboBox2.SelectedIndex
                client.Publish("pozyxinput",
Encoding.Default.GetBytes(Label9.Text), Qos, False)
                ToolStripStatusLabel1.Text = "Publish to " + "{" + "pozyxinput" +
"}"
            Catch ex As Exception
                ToolStripStatusLabel1.Text = "Error"
                MsgBox(ex.Message, MsgBoxStyle.Critical)
            End Try
            Try
                Dim Qos As Byte = ComboBox2.SelectedIndex
                client.Publish("pozyxinput", Encoding.Default.GetBytes("B=1;"),
Qos, False)
                ToolStripStatusLabel1.Text = "Publish to " + "{" + "pozyxinput" +
"}"
            Catch ex As Exception
                ToolStripStatusLabel1.Text = "Error"
                MsgBox(ex.Message, MsgBoxStyle.Critical)
            End Try
            kunci2 = 1
        ElseIf kunci1 = 0 And kunci2 = 1 Then

            kunci2 = 0
        End If

        Label9.Text = "X=" & titikakhir(0, 0) & ";Y=" & titikakhir(0, 1) & ";"

        Label13.Text = titikakhir(0, 0)
        Label18.Text = titikakhir(0, 1)
        Label14.Text = titikakhir(1, 0)
        Label17.Text = titikakhir(1, 1)
        Label15.Text = titikakhir(2, 0)
        Label16.Text = titikakhir(2, 1)

        surface.DrawLine(pen2, 500, 0, 500, 900)
        surface.DrawLine(pen2, 0, 600, 1900, 600)
        pen3 = New Pen(Color.Blue, 10)
        pen4 = New Pen(Color.Red, 10)
        pen1 = New Pen(Color.Black, 10)
        surface = Panel1.CreateGraphics()
        rectangle2 = New Rectangle(xakhir - 4, yakhir - 4, 8, 8)
        rectangle3 = New Rectangle((posisix / 10 + 500) - 4, (600 - (posisiy / 10))
- 4, 8, 8)
        surface.FillRectangle(brush1, rectangle1)
```

```
        pen2 = New Pen(Color.Black, 2)
        surface.DrawLine(pen2, 500, 0, 500, 900)
        surface.DrawLine(pen2, 0, 600, 1900, 600)
        arahx = 20 * Math.Cos(Math.PI * arah / 180.0)
        arahy = 20 * Math.Sin(Math.PI * arah / 180.0)
        posisixarah = (posisix / 10 + 500)
        posisiyarah = (600 - (posisiy / 10))
        surface.DrawLine(pen4, posisixarah, posisiyarah, posisixarah + arahx,
posisiyarah - arahy)
        surface.DrawEllipse(pen1, rectangle2)
        surface.DrawEllipse(pen3, rectangle3)


        surface.DrawLine(pen2, 500, 0, 500, 900)
        surface.DrawLine(pen2, 0, 600, 1900, 600)
        surface.DrawLine(pen2, 500 + 400, 150, 500 + 400, 600)
        surface.DrawLine(pen2, 500, 600 - 450, 900, 600 - 450)

    End Sub
    Private Sub Timer1_Tick(sender As Object, e As EventArgs) Handles Timer1.Tick
        Timer2.Enabled = True
        Timer1.Enabled = False
    End Sub
    Dim c As Integer
    Private Sub Button6_Click(sender As Object, e As EventArgs) Handles
Button6.Click
        urutan = 0
        kunci = 0
        For c = 0 To 10 - 1
            titikakhir(c, 0) = 0
            titikakhir(c, 1) = 0
        Next
    End Sub

    Private Sub Button7_Click(sender As Object, e As EventArgs) Handles
Button7.Click
        Try
            Dim Qos As Byte = ComboBox2.SelectedIndex
            client.Publish("pozyxinput", Encoding.Default.GetBytes("B=1;"), Qos,
False)
            ToolStripStatusLabel1.Text = "Publish to " + "{" + "pozyxinput" + "}"
        Catch ex As Exception
            ToolStripStatusLabel1.Text = "Error"
            MsgBox(ex.Message, MsgBoxStyle.Critical)
        End Try
    End Sub

    Private Sub Button8_Click(sender As Object, e As EventArgs) Handles
Button8.Click
        Try
            Dim Qos As Byte = ComboBox2.SelectedIndex
            client.Publish("pozyxinput", Encoding.Default.GetBytes("B=0;"), Qos,
False)
            ToolStripStatusLabel1.Text = "Publish to " + "{" + "pozyxinput" + "}"
        Catch ex As Exception
            ToolStripStatusLabel1.Text = "Error"
            MsgBox(ex.Message, MsgBoxStyle.Critical)
        End Try
    End Sub


End Class
```