

BAB II

LANDASAN TEORI

Bab ini membahas tentang landasan teori mengenai pengantar robotika, sensor jarak ultrasonik (PING), sensor api lilin (UVTron), sensor suhu api lilin (*Thermal Array TPA81*), I2C (*Inter-Integrated Circuit*), dan pengontrol mikro.

II.1 Pengantar Robotika

Keunggulan dalam teknologi robotika tidak dapat dipungkiri telah lama dijadikan ikon kebanggaan negara-negara maju di dunia. Kecanggihan teknologi yang dimiliki, gedung-gedung tinggi yang mencakar langit, tingkat kesejahteraan rakyatnya yang tinggi, kota-kota yang modern, belum terasa lengkap tanpa popularitas kepiawaian dalam dunia robotika.

II.1.1 Sejarah Robot ^[14]

Awal munculnya robot dapat diketahui dari bangsa Yunani kuno yang membuat patung yang dapat dipindah-pindahkan. Sekitar 270 BC, **Ctesibus**, seorang insinyur Yunani membuat organ dan jam air dengan komponen yang dapat dipindahkan. Zaman Nabi Muhammad SAW pun, telah membuat mesin perang yang menggunakan roda dan melontarkan bom.

Pada tahun 1770, **Pierre Jaquet Droz**, seorang pembuat jam berkebangsaan Swiss membuat tiga boneka mekanis. Uniknya, boneka tersebut dapat melakukan fungsi spesifik, yaitu dapat menulis, yang lainnya dapat memainkan musik dan organ, dan yang ketiga dapat menggambar.

Pada tahun 1898, **Nikola Tesla** membuat sebuah boat yang dikontrol melalui *radio remote control*, dan didemokan di Madison Square Garden. Namun usaha untuk membuat *autonomus* boat tersebut gagal karena masalah dana.

Awal kejayaan robot pada tahun 1970, ketika **Profesor Victor Scheinman** dari Universitas Stanford mendesain lengan standar. Saat ini konfigurasi kinematiknya dikenal sebagai standar lengan robot.

Ketika para pencipta robot pertama kali mencoba meniru manusia dan hewan, ditemukan bahwa hal tersebut sangatlah sulit dan membutuhkan tenaga penghitungan yang jauh lebih banyak dari yang tersedia pada masa itu. Jadi, penekanan perkembangan diubah ke bidang riset lainnya. Robot sederhana beroda digunakan untuk melakukan eksperimen dalam tingkah laku, navigasi, dan perencanaan jalur. Teknik navigasi tersebut telah berkembang menjadi sistem kontrol robot otonom yang tersedia secara komersial. Contoh paling mutakhir dari sistem kontrol navigasi otonom yang tersedia sekarang ini termasuk sistem navigasi berdasarkan laser dan VSLAM (*Visual Simultaneous Localization and Mapping*) dari *Active Media Robotics* dan *Evolution Robotics*.

Ketika para teknisi siap untuk mencoba robot berjalan kembali, dimulai dengan heksapoda dan platform berkaki banyak lainnya. Robot-robot tersebut meniru serangga dan arthropoda dari segi bentuk dan fungsi. Tren menuju jenis badan tersebut menawarkan fleksibilitas yang besar dan terbukti dapat beradaptasi dengan berbagai macam lingkungan, tetapi biaya dari penambahan kerumitan mekanik telah mencegah pengadopsiannya. Dengan lebih dari empat kaki, robot-robot ini stabil secara statis yang membuatnya bekerja lebih mudah.

Sementara itu, robot bersayap eksperimental dan contoh lain mengeksplorasi biomimikri juga dalam tahap pengembangan dini. Yang disebut "nanomotor" dan "kawat cerdas" diperkirakan dapat menyederhanakan daya gerak secara drastis, sementara stabilisasi dalam penerbangan nampaknya cenderung diperbaiki melalui giroskop yang sangat kecil.

Terakhir, pada tahun 2000 Honda memamerkan robot yang dibangun bertahun-tahun lamanya yang bernama ASIMO ditunjukkan pada Gambar 2.1, serta disusul oleh Sony yaitu robot anjing AIBO ditunjukkan pada Gambar 2.2.



Gambar 2.1 Robot ASIMO dari Honda



Gambar 2.2 Robot AIBO dari Sony

II.1.2 Defenisi Robot ^[3]

Robot memiliki banyak arti sesuai dengan pengertian pada tiap-tiap individu. Kata robot pada mulanya digunakan pada tahun 1921 oleh **Karl Capek** yang merupakan seorang penulis berkebangsaan Ceko. Naskah drama yang ditulisnya berjudul *RUR (Rossum's Universal Robot)*. Beliau mengambil kata robot dari bahasa Ceko, *robota*, yang berarti pekerja. Beliau bercerita tentang mesin yang menyerupai manusia, tetapi mampu bekerja terus - menerus tanpa lelah.

Kemudian pada tahun 1942 **Isacc Asimov** memperkenalkan istilah *robotic* pada cerpennya “*Runaround*”, yang mencantumkan aturan tentang robotika (*Three Rules of Robotic*) yaitu :

1. Robot tidak boleh mencederai manusia.
2. Robot harus mematuhi perintah yang diberikan manusia kecuali bertentangan dengan hukum pertama.
3. Robot harus mempertahankan keberadaannya sejauh tindakannya itu tidak melanggar hukum pertama.

Teknologi robotika berkembang pesat karena memiliki keuntungan yang dapat membantu pertumbuhan dalam bidang industri, antara lain mengurangi biaya tenaga kerja, meningkatkan ketepatan serta produktifitas, meningkatkan fleksibilitas, dan dapat bekerja di tempat-tempat yang berbahaya bagi manusia.

II.1.3 Keuntungan Penggunaan Robot ^[3]

Penggunaan robot didasarkan pada beberapa alasan, diantaranya adalah:

1. Produktivitas

Robot dapat mengurangi biaya produksi karena produktifitas robot mencapai 98%. Selain itu, robot dapat menekan persentase kegagalan produk lebih tinggi dibandingkan hasil kerja manusia.

2. Kecepatan

Robot bekerja lebih cepat daripada manusia. Sebagai perbandingan, sebuah robot dapat melakukan pekerjaan las dengan kecepatan 75cm/menit, sementara kecepatan seorang tukang las adalah 25cm/menit. Sebuah robot dapat mengecat dua lapis keseluruhan badan mobil dengan waktu 90detik, sedangkan seorang tukang cat membutuhkan waktu 30menit untuk melakukan pekerjaan serupa.

3. Kualitas

Robot menghasilkan suatu produk dengan kualitas lebih baik daripada manusia. Robot dapat melakukan proses pencetakan dengan ketepatan waktu operasi yang sangat presisi seperti yang telah diprogram dan menghasilkan cetakan berkualitas tinggi.

4. Kondisi Berbahaya

Robot dapat membantu manusia pada pekerjaan yang membahayakan keselamatan operator. Sebagai contoh, pada pekerjaan pengangkutan plat logam panas dari tungku ke mesin *press*, keberadaan robot memungkinkan manusia untuk tidak berhubungan langsung dengan bunga api yang terpentil dari logam panas. Sebuah robot tidak memerlukan perlindungan seperti manusia.

5. Kemudahan Pengendalian Manajemen

Catatan kerja robot dapat disimpan oleh komputer secara akurat. Selain itu, penugasan pekerjaan antara dua stasiun yang berurutan dapat diprogram dahulu dan penjadwalan pekerjaan tersebut dapat dikelola dengan ketat.

6. Ketahanan

Mesin statis akan ketinggalan zaman jika sifat suatu pekerjaan berubah. Sebaliknya, robot dapat diprogram ulang sesuai dengan tugas atau pekerjaan yang baru.

II.1.4 Klasifikasi Robot Berdasarkan Tingkat Kemampuan Melakukan Tugas ^[3]

Pengklasifikasian robot berdasarkan tingkat kemampuan melakukan tugasnya dibagi menjadi beberapa bagian yaitu :

1. *Sequence Controlled Robots*

Robot jenis ini melaksanakan suatu rangkaian kegiatan menurut perintah yang diberikan, misalnya: rangkaian mencuci dalam sebuah mesin cuci.

2. *Computed Trajectory Robots*

Robot jenis ini mengikuti jalur yang telah ditentukan antara posisi awal dan posisi akhir, sebagai contoh: mesin *numerical control*.

3. *Adaptive Robots*

Robot jenis ini memberikan reaksi dengan menggunakan sensor. Performansi robot dioptimalkan dengan penyesuaian kontrol pada parameter yang berubah-ubah.

4. *Intelligent Robots*

Robot jenis ini dilengkapi dengan sensor untuk merekam kondisi lingkungan sebagai pengetahuan dasar. Robot ini bekerja dengan bantuan sistem cerdas (*expert system*). Sistem cerdas ini terus dilatih oleh pemrogram sehingga performansi robot menjadi lebih baik seiring dengan pelatihan yang diberikan.

II.1.5 Klasifikasi Robot Berdasarkan Mobilitas ^[3]

Pengklasifikasian robot berdasarkan mobilitasnya dapat dibagi menjadi beberapa bagian, yaitu:

1. *Fixed Type*

Robot ini berada dalam kondisi diam atau kondisi beroperasi pada jalur gerak yang pendek.

2. *Mobile Robots*

Robot ini dilengkapi dengan roda, dapat bergerak pada daerah yang luas dan tidak terlalu sulit atau ekstrim.

3. *Walking Robots*

Robot ini mempunyai kaki yang dapat bergerak pada daerah yang sulit. *Walking robot* tertentu mempunyai kemampuan untuk memanjat dan menuruni lereng atau tangga. Masalah utama dari *walking robot* adalah posisi keseimbangan yang berbeda dalam melakukan beragam tugas.

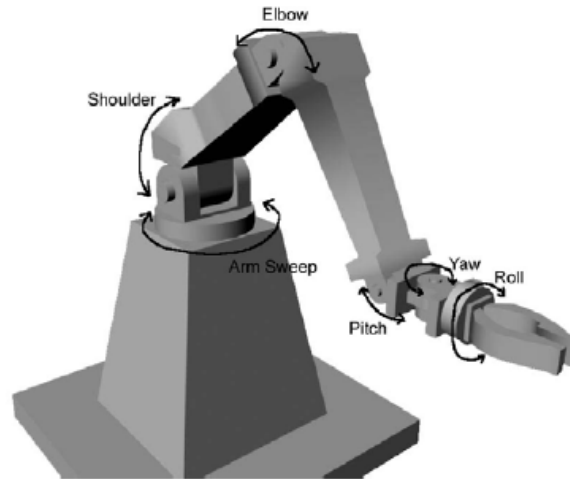
II.1.6 Derajat Kebebasan (*Degrees Of Freedom*) ^[5]

Degrees Of Freedom (DOF) adalah setiap titik sumbu gerakan mekanik pada robot, tidak terhitung untuk End Effector. DOF juga dapat diartikan sebagai jumlah fungsi gerak pada sebuah robot. *Degrees Of Movement* (DOM) adalah kebebasan / kemampuan untuk melakukan sebuah gerakan.

Sebagai contoh, Gambar 2.3 adalah sebuah robot dengan enam derajat kebebasan:

1. *Base Rotation* (dudukan untuk berputar)
2. *Shoulder Flex* (lengan atas / pundak)
3. *Elbow Flex* (lengan bawah)
4. *Wrist Pitch* (pergelangan angguk)

5. *Wrist Yaw* (pergelangan sisi)
6. *Wrist Roll* (pergelangan putar)



Gambar 2.3 Contoh Robot dengan 6 Derajat Kebebasan

II.1.7 Klasifikasi Robot Berdasarkan Konstruksi Robot ^[14]

Robot memiliki berbagai macam konstruksi. Diantaranya adalah:

a. *Mobile Robot* (bergerak)

Robot bergerak adalah robot yang dapat berpindah dari satu tempat ke tempat lain dengan menggunakan alat gerak seperti roda, kaki, dan lain lain.

Robot bergerak terdiri dari :

a.1. *Flying Robot*

Flying Robot adalah robot yang memiliki kemampuan untuk terbang di udara. Bentuk dan strukturnya dapat berupa helikopter yaitu memiliki baling-baling dalam dua arah yang berbeda. Saat ini mulai dikembangkan *micro flying robot*. Robot ini sangat kecil dan sangat ringan. Robot ini akan digunakan dalam berbagai bidang, antara lain adalah untuk melakukan pengamatan dan pencarian di wilayah-wilayah sempit maupun daerah yang berbahaya.

a.2. *Robot Beroda*

Robot Beroda adalah konstruksi robot yang ciri khasnya adalah memiliki aktuator berupa roda untuk menggerakkan keseluruhan badan robot tersebut,

sehingga robot tersebut dapat melakukan perpindahan posisi dari satu titik ke titik yang lain.

Robot beroda ini sangat disukai bagi yang mulai mempelajari robot. Hal ini karena untuk membuat robot beroda tidak memerlukan kerja fisik yang berat. Untuk dapat membuat sebuah robot beroda hanya diperlukan pengetahuan tentang pengontrol mikro dan sensor-sensor elektronika.

Rangka robot beroda dapat dengan mudah dibuat dengan menggunakan plywood/triplek, akrilik sampai menggunakan logam (aluminium). Robot beroda dapat dibuat sebagai pengikut garis (*Line Follower*) atau pengikut dinding (*Wall Follower*) ataupun pengikut cahaya.

a.3. Robot Humanoid

Robot Humanoid adalah sebuah robot yang secara keseluruhan memiliki bentuk menyerupai manusia. Secara umum robot humanoid memiliki batang tubuh dengan kepala, dua buah lengan dan dua buah kaki, meskipun beberapa bentuk dari robot humanoid hanya memiliki beberapa bagian tubuh saja, contohnya, dari bagian pinggang ke atas saja. Beberapa robot humanoid juga memiliki wajah beserta mata dan mulut. Androids adalah robot humanoid yang diciptakan menyerupai lelaki, dan Gynoids adalah robot humanoid yang dibuat menyerupai perempuan, istilah “Android” sering digunakan untuk menyatakan kedua jenis kelamin.

a.4. Robot Berkaki

Robot jenis ini tidak dilengkapi dengan roda melainkan bergerak dengan menggunakan kaki. Biasanya robot ini pada umumnya berbentuk menyerupai serangga atau anjing dan dilengkapi dengan empat atau enam kaki. Selain strukturnya yang mirip serangga atau anjing, pola gerak yang dilakukan juga biasanya menyerupai serangga atau anjing juga. Dengan empat kaki atau lebih, robot ini stabil secara statis yang membuatnya lebih mudah berjalan.

Robot berkaki banyak digunakan sebagai alat penelitian di beberapa area. Para peneliti perlu untuk mengerti struktur tubuh dan kelakuan serangga (biomekanik) untuk membuat dan mempelajari robot berkaki.

Selain untuk penelitian, robot berkaki sangat populer dalam bidang hiburan juga. Sebagai contoh, Aibo, robot anjing produksi Sony yang dapat memahami perintah yang diucapkan manusia, dapat menunjukkan ekspresi ketika disentuh dan dibelai, dapat bermain bola dan masih banyak lagi yang dapat dilakukan oleh robot tersebut. Robot berkaki yang memiliki kecerdasan buatan akan sangat bermanfaat untuk melakukan misi luar angkasa yang sangat jauh dan berbahaya.

b. Manipulator Robot (lengan)

Robot Manipulator adalah konstruksi robot yang memiliki bentuk hanya sebuah lengan saja. Robot ini memiliki aktuator berupa motor untuk menggerakkan seluruh bagian robot tersebut.

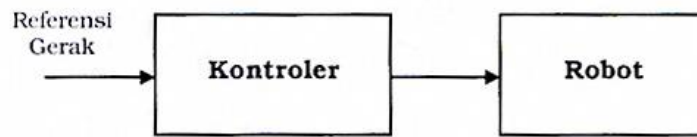
Robot manipulator ini sangat bermanfaat dan cukup banyak dipakai dalam bidang industri. Terutama dalam bagian *packing* dalam suatu pabrik. Selain untuk menghemat waktu, robot ini digunakan karena hasil yang diperoleh juga lebih baik daripada hasil yang dikerjakan oleh manusia.

c. Robot Jaringan

Robot Jaringan adalah pendekatan baru untuk melakukan control robot menggunakan jaringan internet dengan protocol TCP/IP. Perkembangan robot jaringan dipicu oleh kemajuan jaringan dan internet yang pesat. Dengan koneksi jaringan, proses kontrol dan pemantauan, termasuk akuisisi data bila ada, seluruhnya dilakukan melalui jaringan. Keuntungan lain, koneksi ini bisa dilakukan secara nirkabel.

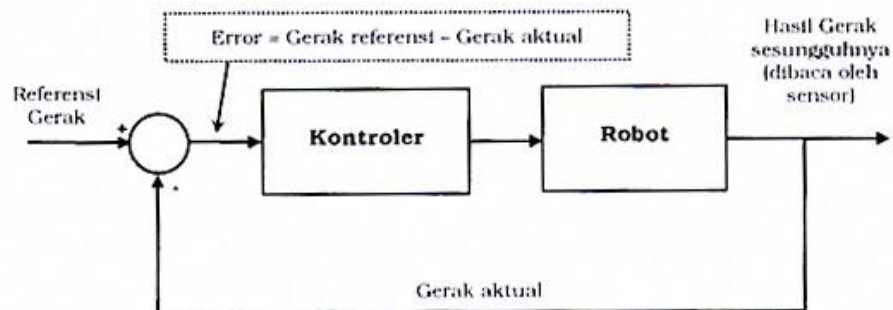
II.1.8 Sistem Kontrol Robot ^[3]

Sistem kontrol robot pada dasarnya terbagi menjadi dua kelompok, yaitu sistem kontrol loop terbuka dan loop tertutup. Diagram kontrol loop terbuka pada sistem robot ditunjukkan dalam Gambar 2.4, sedangkan diagram kontrol robot loop tertutup ditunjukkan dalam Gambar 2.5.



Gambar 2.4 Kontrol Robot Loop Terbuka

Pada Gambar 2.4, kontrol loop terbuka dapat dinyatakan sebagai sistem kontrol yang outputnya tidak diperhitungkan ulang oleh kontroler. Keadaan robot yang telah mencapai target seperti yang dikehendaki dan sesuai dengan referensi, tidak dapat mempengaruhi kinerja pengontrol. Kontrol ini sesuai untuk sistem operasi robot yang memiliki aktuator yang beroperasi berdasarkan umpan logika berbasis konfigurasi langkah sesuai urutan, misalnya motor stepper. Motor stepper tidak perlu dipasang sensor pada porosnya untuk mengetahui posisi akhir. Jika dalam keadaan berfungsi baik dan tidak ada masalah beban lebih maka motor stepper akan berputar sesuai dengan perintah pengontrol dan mencapai posisi target dengan tepat.



Gambar 2.5 Kontrol Robot Loop Tertutup

Pada Gambar 2.5, jika hasil gerak aktual telah sama dengan referensi maka *Input* kontroler akan nol, artinya kontroler tidak lagi memberikan sinyal aktuator kepada robot karena target akhir perintah gerak telah diperoleh. Jika semakin kecil *error* yang terhitung, maka semakin kecil pula sinyal pengemudian kontroler terhadap robot, sampai akhirnya mencapai kondisi tenang (*steady state*).

Referensi gerak dan hasil gerak aktual dapat berupa posisi, kecepatan, akselerasi, atau gabungan diantaranya. Kontrol bersifat konvergen jika dalam rentang waktu pengontrolan nilai *error* menuju nol, dan keadaan dikatakan stabil jika setelah konvergen pengontrol mampu menjaga agar *error* selalu nol.

Dua pengertian dasar antara konvergen dan stabil adalah sangat penting dalam kontrol loop tertutup. Stabil dan konvergen diukur berdasarkan referensi geraknya. Posisi akhir dianggap konvergen bila makin lama gerakan makin perlahan dan akhirnya diam pada posisi seperti yang dikehendaki referensi, dan dikatakan stabil jika posisi akhir yang diam ini dapat dipertahankan dalam masa-masa berikutnya.

II.1.9 Teknik Manuver ^[10]

Salah satu kemampuan yang harus dimiliki oleh *Autonomous Mobile Robot (AMR)* ialah mampu bergerak ke posisi tujuan walaupun dengan sedikit informasi tentang lingkungannya, yaitu dengan cara mengoptimalkan informasi tersebut dan mengumpulkan informasi sekelilingnya dengan menggunakan sensor yang dimilikinya. Perencanaan jalur dan manuverlah yang memegang peranan penting dalam memecahkan permasalahan di atas.

Manuver mengacu pada metodologi jalur yang terbentuk oleh robot dalam mengatasi rintangan yang tidak diharapkan. Hasil pergerakan tergantung pada posisi aktual robot dan pembacaan informasi dari sensor. Ada terdapat banyak macam algoritma manuver. Banyak metoda yang diusulkan yang dibedakan pada penggunaan sensor data dan strategi kontrol pergerakan dalam mengatasi rintangan.

Metode manuver untuk autonomous robot terdiri dari tiga metode dasar, yaitu :

1) Metode *Edge Detection*.

Dalam metode ini, sebuah algoritma disusun untuk menentukan posisi tepi vertikal dari rintangan setelah itu melakukan pengendalian wahana menyusuri rintangan sampai ke tepi vertikal di sisi lainnya. Garis yang menghubungkan

kedua tepi tersebut merepresentasikan batas dari rintangan yang ada di hadapan wahana.

2) Metode *Certainty Grid*.

Dalam metode ini, lingkungan wahana direpresentasikan menjadi array dua dimensi dari elemen-elemen persegi yang disebut *sel/kisi*. Setiap *sel* memiliki nilai kepastian tertentu. Nilai kepastian ini diperbaharui oleh fungsi probabilitas yang diperoleh dari karakteristik yang dimiliki oleh sensor. Dari nilai kepastian setiap *sel* dalam lingkungan tersebut, dilakukan penghindaran tabrakan dengan mencari lokasi yang memiliki nilai kepastian yang kecil.

3) Metode *Potensial Field*.

Metode ini memanfaatkan gaya *virtual* yang diperoleh dari pendeteksian lingkungan dan pendeteksian letak target yang akan dituju. Gaya tolak *virtual* terjadi karena adanya pendeteksian rintangan oleh sensor. Gaya-gaya yang terjadi dari rintangan tersebut kemudian digabungkan menjadi sebuah resultan gaya tolak *virtual*. Gaya tarik *virtual* terjadi dari perhitungan posisi target terhadap wahana. Hasil gaya tarik *virtual* kemudian dijumlahkan dengan resultan gaya tolak *virtual* sehingga dihasilkan resultan gaya *virtual* dan sudut gaya *virtual*. Sudut resultan tersebut digunakan untuk melakukan belokan untuk menghindari tabrakan.

Dari ketiga metode dasar di atas masih diperoleh banyak sekali kekurangan yang membuat robot tidak sampai ke posisi tujuan yang diinginkan. Sebagai contoh pada metode *edge detection*, robot tidak akan sampai ke posisi tujuan jika tujuan berada pada ruang yang terpisah dari lainnya (seperti pulau), tetapi metode dasar ini cukup efektif untuk pola maze yang tetap.

II.1.10 KRCI 2009 ^[6]

Tema untuk Kontes Robot Cerdas Indonesia 2009 adalah Robot Cerdas Pemadam Api yang diselaraskan dengan tema yang ditentukan oleh Trinity College Fire - Fighting Robot Contest 2009.

Divisi yang diperlombakan pada KRCI 2009 ada empat divisi yaitu:

1. Divisi Senior Beroda

Suatu divisi dengan robot menggunakan roda sebagai alat geraknya dalam misi mencari dan memadamkan api pada arena lapangan dengan tingkat kesulitan sedang. Pada divisi ini yang diutamakan adalah kecepatan dan kemampuan robot dalam bernavigasi dan bermanuver dalam mencari dan memadamkan api di suatu arena dengan peta tertentu.

2. Divisi Senior Berkaki

Sama dengan Divisi Senior Beroda, hanya saja robot menggunakan kaki sebagai alat geraknya.

3. Divisi Expert Single

Suatu divisi dengan tingkat kesulitan yang tinggi dengan misi robot tidak saja mencari api lilin dan memadamkannya namun juga harus dapat menemukan posisi boneka bayi pada arena yang konfigurasi ruangnya acak.

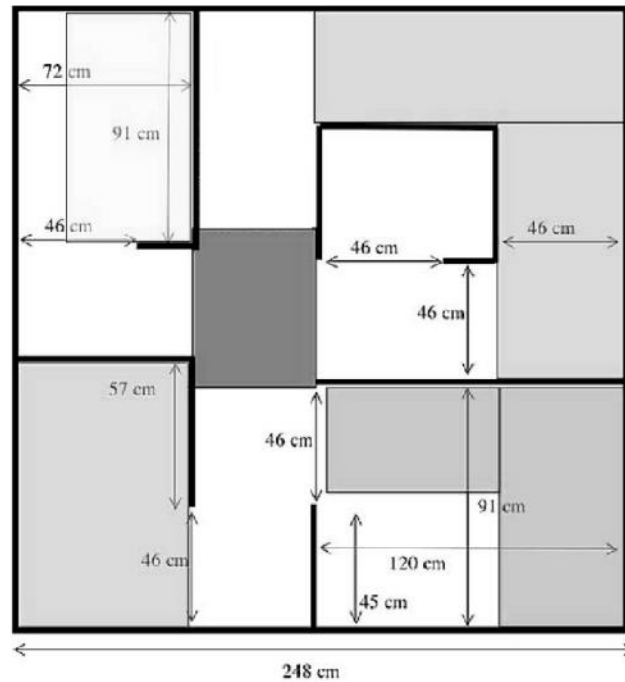
4. Divisi Expert Battle

Divisi ini mempertandingkan dua robot dari dua tim yang berbeda dalam satu arena. Pertandingan bertujuan memadamkan api dan mengangkat dan memindahkan boneka bayi masing-masing. Untuk menambah nilai masing-masing boleh merebut lilin dan/atau boneka lawan. Divisi ini menggantikan divisi Expert Swarm. Divisi ini mirip dengan pola-pola pertandingan KRI namun dengan mengambil tema KRCI.

Peraturan KRCI 2009 Divisi Senior Berkaki adalah sebagai berikut :

1. Dimensi robot maksimum yaitu $p = 46$ cm; $l = 31$ cm; dan $t = 27$ cm.
2. Untuk memadamkan api lilin diperbolehkan menggunakan kipas (dengan tiupan angin) dan/atau extinguisher (dengan cairan).
3. Api disimulasikan dengan api lilin dengan tinggi berkisar antara 15cm – 20cm (belum termasuk sumbu) dan diameter 2cm – 3cm

4. Denah lapangan berbentuk seperti pada Gambar 2.6.



Gambar 2.6 Denah Lapangan KRCI 2009

II.2 Motor Servo ^[4]

Motor servo adalah motor yang mampu bekerja dua arah (*clockwise* dan *counter – clockwise*), dengan arah dan sudut pergerakan rotornya dapat dikontrol hanya dengan memberikan sinyal PWM pada bagian pin kontrolnya.

Motor Servo merupakan sebuah motor DC yang memiliki rangkaian kontrol elektronika dan *internal gear*, untuk mengontrol pergerakan serta sudut angularnya. Motor servo dapat berputar dengan lambat, yang biasanya ditunjukkan oleh *rate* putarannya yang lambat. Akan tetapi, motor servo memiliki torsi yang kuat karena *internal gear* – nya.

Motor servo mempunyai beberapa ukuran dan tipe yang dapat digunakan untuk segala kebutuhan mulai dari mesin fotokopi, model pesawat terbang, dan model mobil. Aplikasi terbaru dari motor servo yaitu digunakan untuk aplikasi pembuatan robot. Motor servo pada awalnya dibuat untuk dioperasikan melewati

saluran radio kontrol dan biasanya mengacu pada radio kontrol. Motor servo tersebut bergerak dengan sinyal yang berasal dari penerima.

Adapun jenis-jenis motor servo yaitu :

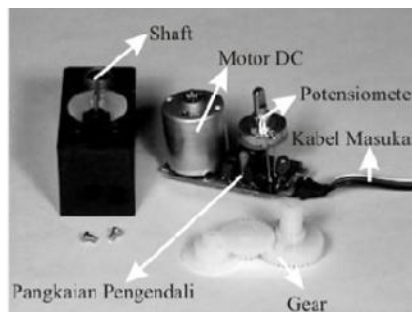
a) Motor Servo Standar 180°

Motor servo jenis ini mampu bergerak dua arah (*clockwise* dan *counter - clockwise*) dengan defleksi masing-masing sudut mencapai 90°, sehingga total defleksi sudut dari kanan-tengah-kiri adalah 180°.

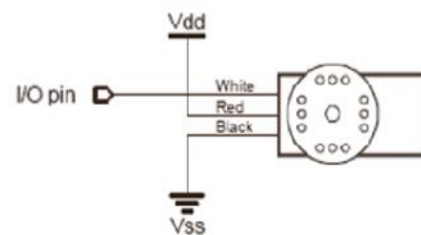
b) Motor Servo *Continuous*

Motor servo jenis ini mampu bergerak dua arah (*clockwise* dan *counter - clockwise*) tanpa batasan defleksi sudut putar (dapat berputar secara kontinu).

Untuk beroperasi, motor servo membutuhkan catu daya 5V DC. Kabel merah dihubungkan ke Vdd (+), kabel hitam dihubungkan ke Vss (ground), dan sinyal kontrol PWM dihubungkan dengan kabel putih. Pada Gambar 2.7 dan Gambar 2.8 menunjukkan bentuk beserta koneksi kabel motor servo.



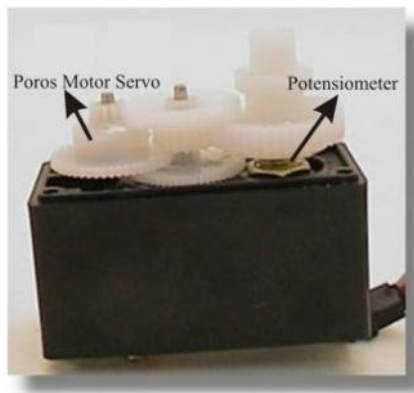
Gambar 2.7 Struktur Dalam Motor Servo



Gambar 2.8 Koneksi Kabel Motor Servo

Secara umum, motor servo terdiri atas sebuah rangkaian pengontrol, sebuah potensiometer, dan sejumlah *gear* untuk memperbesar torsi. Motor dan potensiometer dihubungkan ke rangkaian pengontrol dan ketiganya membentuk sistem loop tertutup. Untuk mengaktifkan motor, sebuah sinyal digital dikirim ke rangkaian pengontrol. Potensiometer ini bertugas memonitor sudut poros motor servo. Motor akan berputar ke arah sudut yang diinginkan sampai sudut poros motor servo sesuai dengan yang diinginkan, kemudian motor akan berhenti

bergerak. Pada umumnya motor servo didesain untuk rotasi yang terbatas yaitu berkisar antara 0° sampai 180° . Sebuah motor servo yang normal tidak dapat berputar lebih jauh, dikarenakan sebuah penghalang mekanik yang diselipkan pada roda gigi keluaran utama. Motor servo dapat berotasi secara penuh dengan melakukan suatu modifikasi.



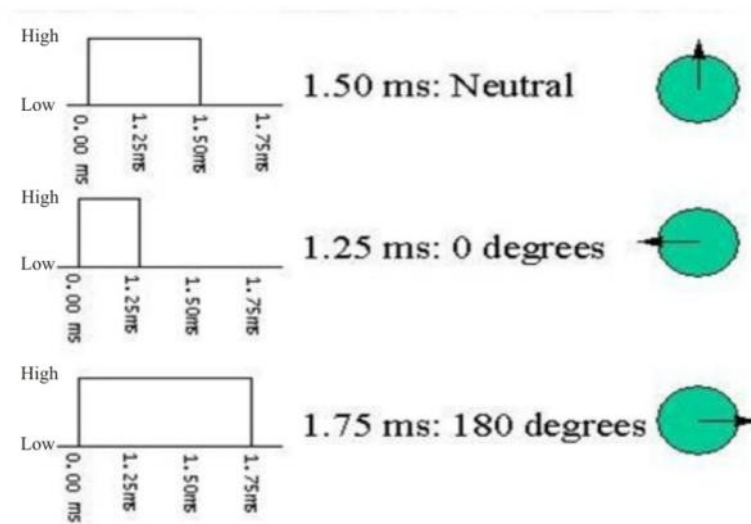
Gambar 2.9 Potensiometer Motor Servo

Gambar 2.9 adalah bagian dalam motor servo yang memperlihatkan poros motor servo yang terhubung dengan potensiometer. *Gear* di tengah digunakan sebagai penghubung antara *gear* poros motor servo, dan *gear* potensiometer. Setiap perubahan sudut poros motor servo akan mengubah posisi potensiometer. Hal ini membuat posisi poros motor servo dapat diketahui dengan tepat.

Besar daya yang dibutuhkan motor servo sebanding dengan sudut yang perlu ditempuh poros. Motor servo menerapkan sistem kontrol proporsional. Jika poros perlu berputar jauh maka poros akan bergerak dengan kecepatan penuh, namun jika sudut tempuh poros dekat maka poros akan bergerak perlahan.

Untuk mendapatkan sudut yang diinginkan maka motor servo harus mendapatkan *input* lebar pulsa yang tepat. Motor servo diharapkan menerima pulsa setiap 20 milidetik. Sudut keluaran motor servo ditentukan oleh lebar pulsa yang diterima motor servo. Misalnya untuk menghasilkan sudut 90° (posisi netral) maka motor servo perlu menerima pulsa setiap 20 milidetik dengan masing-masing lebar pulsa sebesar 1,5 milidetik. Jika pulsa yang diterima kurang dari 1,5

milidetik maka poros akan berputar ke arah 0° . Sebaliknya, jika pulsa yang diterima motor servo lebih dari 1,5 milidetik maka poros akan berputar ke arah 180° .



Gambar 2.10 Contoh Posisi dan Lebar Pulsa yang Diberikan

Gambar 2.10 adalah sinyal untuk mengontrol sudut motor servo. Sinyal ini berupa sinyal persegi (*square*) dengan lebar pulsa *high* berkisar antara 0,6 milidetik sampai 2,4 milidetik. Pulsa dengan pulsa *high* sebesar 0,6 milidetik akan membuat poros motor servo berada pada posisi 0° . Pulsa dengan pulsa *high* sebesar 1,5 milidetik akan membuat poros motor servo berada pada posisi 90° dan pulsa dengan pulsa *high* sebesar 2,4 milidetik akan membuat poros motor servo berada pada posisi 180° . Poros motor servo dapat menduduki posisi lainnya dengan memberikan lebar pulsa *high* dengan perbandingan perubahan satu derajat poros setiap perubahan sepuluh mikrodetik pulsa *high*.

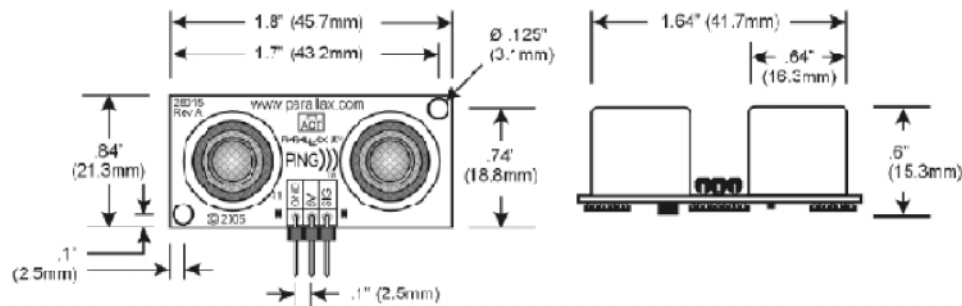
II.3 Sensor

Sensor adalah piranti yang mengubah suatu nilai fisik (*input*) menjadi nilai fisik yang lain (*output*). *Output* yang dihasilkan biasanya berupa sinyal elektrik. Pada Tugas Akhir ini digunakan beberapa jenis sensor yaitu sensor jarak

ultrasonik, sensor api lilin (Hamamatsu UVTron), dan sensor *thermal infrared* (Devantech Thermal Array TPA81).

II.3.1 Sensor Jarak Ultrasonik (PING)^[9]

Sensor jarak ultrasonik sangat cocok dipakai untuk aplikasi-aplikasi yang perlu melakukan pengukuran jarak. Secara umum, sensor ini sangat populer untuk dipakai pada aplikasi-aplikasi robotika. Selain itu, sensor ini juga bisa bermanfaat dalam sistem *security* dan sebagai alternatif lain pengganti sensor jarak *infrared*. Sensor jarak ultrasonik yang dipakai dalam tugas akhir ini adalah sensor “ Parallax's PING)))TM Ultrasonic Range Finder ” atau biasa disebut dengan sensor PING. Berdasarkan *datasheet*, sensor PING ini dapat digunakan untuk mengukur jarak benda sejauh 3cm sampai 300cm. Dimensi dan kaki pin sensor PING ditunjukkan pada Gambar 2.11.

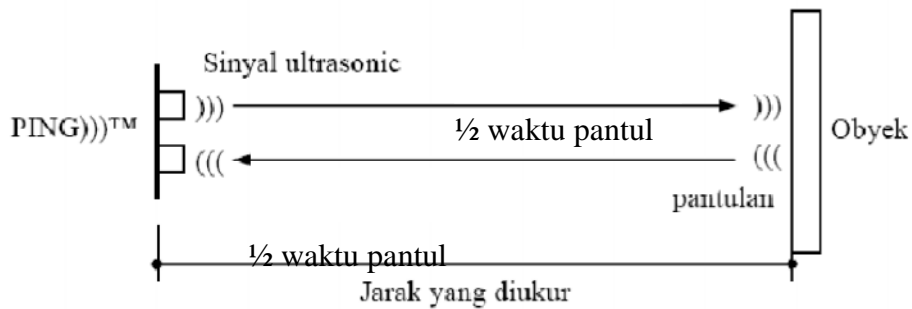


Gambar 2.11 Dimensi Sensor PING

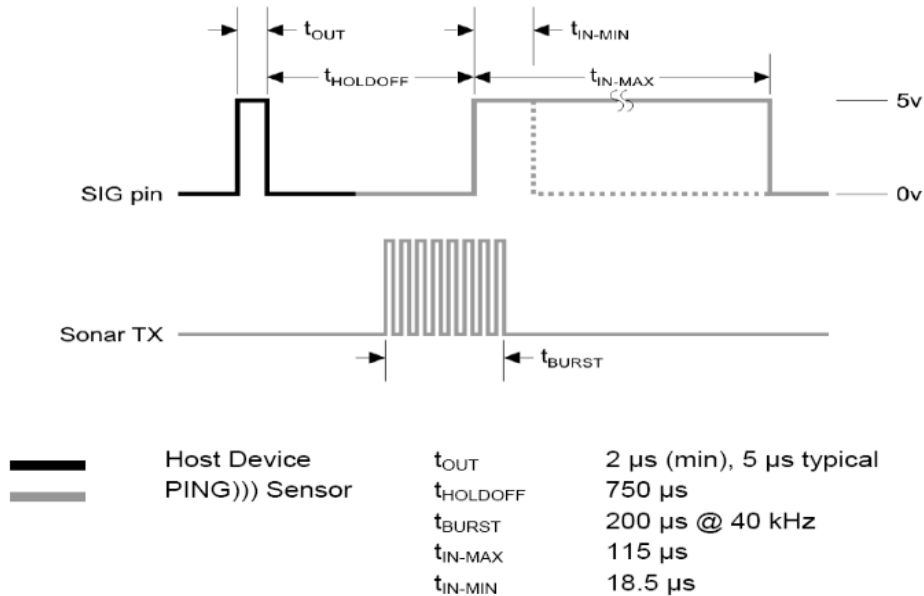
Sensor PING mendeteksi jarak objek dengan cara memancarkan gelombang ultrasonik (40kHz) selama t_{BURST} (200 μ s) kemudian mendeteksi pantulannya. Sensor PING memancarkan gelombang ultrasonik sesuai dengan sinyal kontrol dari pengontrol mikro (sinyal kontrol berupa pulsa *trigger* dengan t_{OUT} min 2 μ s). Gelombang ultrasonik ini melalui udara dengan kecepatan 344m/s, mengenai objek, kemudian memantul kembali ke sensor. Sensor PING mengeluarkan pulsa *output high* pada pin SIG setelah memancarkan gelombang ultrasonik. Setelah gelombang pantulan terdeteksi, sensor PING akan membuat

output low pada pin SIG. Lebar pulsa high (t_{IN}) akan sama dengan lama waktu dipancarkan sampai diterima kembali pantulan gelombang ultrasonik akibat objek. Maka jarak yang diukur dapat dilihat pada Persamaan 2.1. Ilustrasi cara kerja sensor PING ditunjukkan pada Gambar 2.12. Untuk dapat lebih memahami penggunaan cara kerja sensor PING, pada Gambar 2.13 ditunjukkan diagram waktu sensor PING.

$$Jarak = \left[\frac{(t_{IN}) \times (344) m/s}{2} \right] meter \dots\dots\dots (2.1)$$



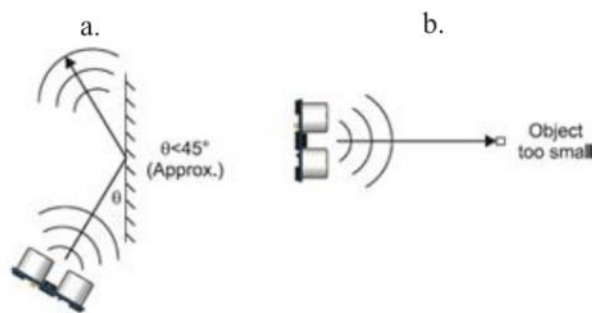
Gambar 2.12 Gambar Ilustrasi Cara Kerja Sensor PING



Gambar 2.13 Diagram Waktu Sensor PING

Hal-hal yang dapat menyebabkan sensor PING tidak teliti untuk mengukur jarak suatu objek adalah sebagai berikut :

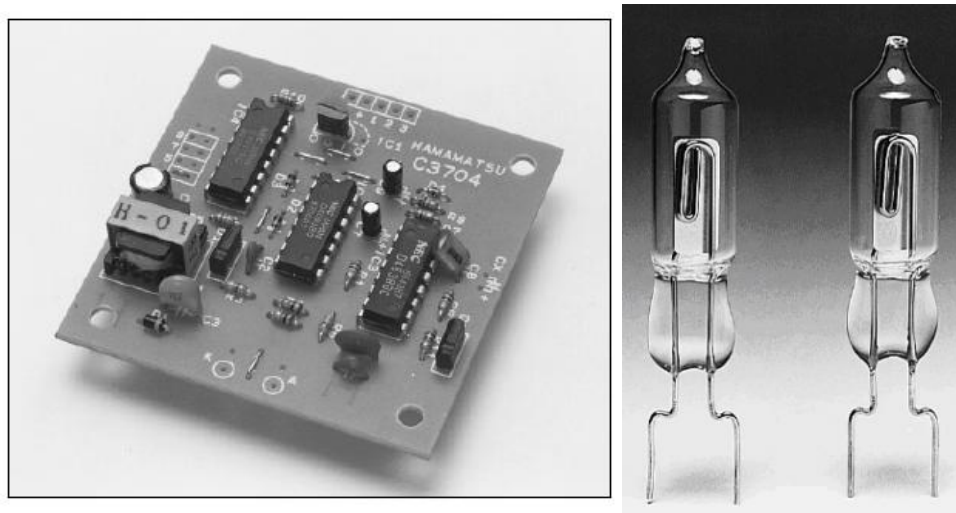
1. Jarak objek tersebut diluar jangkauan ukur ($< 3\text{cm}$ atau $> 300\text{cm}$).
 2. Sudut objek terhadap sensor () $< 45^\circ$, sudut yang terlalu kecil antara sensor jarak ultrasonik dan objek sehingga gelombang ultrasonik tidak dapat dipantulkan kembali ke sensor, ditunjukkan pada Gambar 2.14a.
 3. Objek tersebut terlalu kecil untuk memantulkan gelombang ultrasonik kembali ke sensor, ditunjukkan pada Gambar 2.14b.
 4. Jenis material objek yang bersifat meredam suara, bentuk permukaan yang tidak beraturan, sudut ruangan, dan lain- lain sehingga mungkin tidak dapat memantulkan gelombang ultrasonik yang cukup untuk dideteksi dengan teliti.
- Selain itu, sensor PING dapat digunakan untuk mendeteksi permukaan air, tetapi tidak dianjurkan untuk digunakan di lingkungan yang basah.



Gambar 2.14 Gambar Posisi Objek terhadap Sensor PING

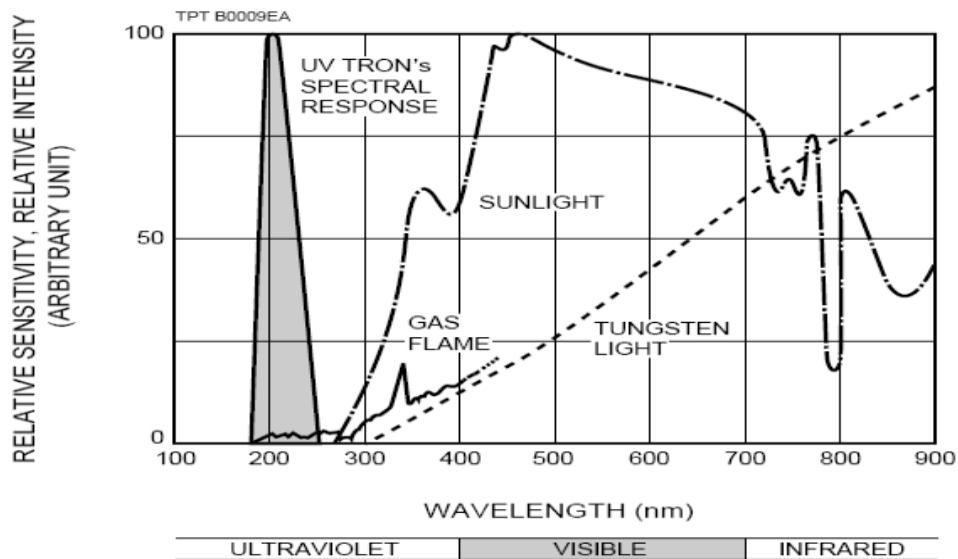
II.3.2 Sensor Api (UVTron) ^[8]

Sistem informasi pada robot untuk mendapatkan kondisi ada tidaknya api lilin pada ruangan merupakan masalah tersendiri dalam penyelesaiannya. Salah satu pemecahan masalah tersebut adalah dengan pemasangan sensor yang bekerja dengan mendeteksi ada atau tidaknya panas api. Sensor ini memberikan sinyal aktif apabila mendeteksi adanya api dalam ruangan. Tipe sensor yang dipilih adalah Hamamatsu R2868 berbentuk bohlam dengan dilengkapi sebuah rangkaian, seperti pada Gambar 2.15.

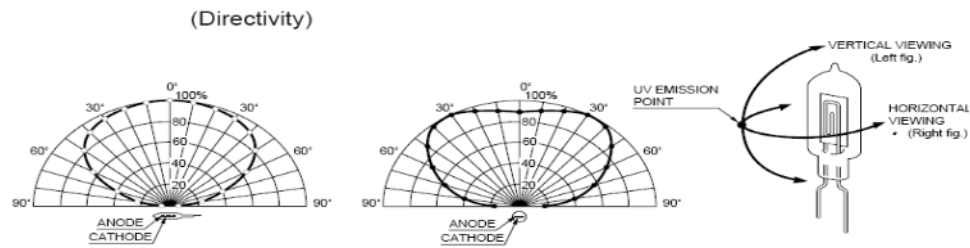


Gambar 2.15 UVTron Hamamatsu R2868

Prinsip kerja Hamamatsu R2868 adalah mendeteksi adanya gelombang ultraviolet pada *range* 185-260nm, di area tersebut adalah sinar ultraviolet yang dihasilkan oleh api, ditunjukkan pada Gambar 2.16. Daerah deteksi Hamamatsu R2868, seperti pada Gambar 2.17.



Gambar 2.16 Spektrum Respon Uvtron

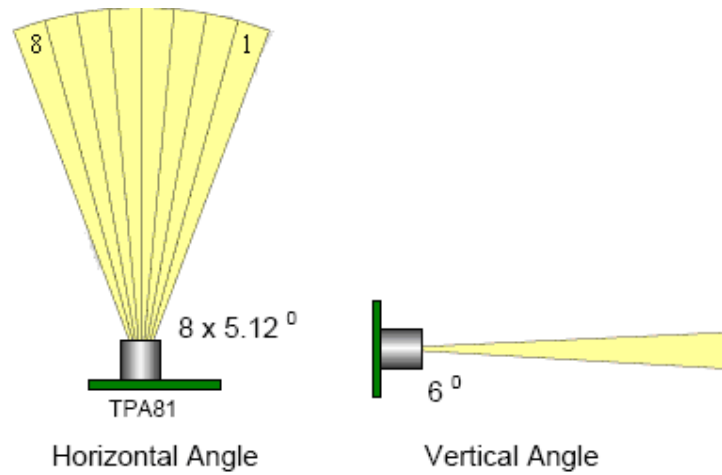


Gambar 2.17 Derajat Sensitivitas Hamamatsu R2868

Hamamatsu R2868 memerlukan sebuah modul rangkaian untuk mengaktifkannya. Modul ini bernama *UVTron Driving Circuit* dengan seri C3704-02. Bekerja pada tegangan 5 volt dengan konsumsi arus sebesar 300 mikro *Ampere* yang akan menghasilkan keluaran digital logika *high* jika terdeteksi api dan logika *low* jika tidak ada api. Dalam perlakuannya, Hamamatsu R2868 tidak boleh dipegang secara langsung oleh tangan manusia karena dapat mengurangi sensitivitas dari sensor api (UVTron) tersebut.

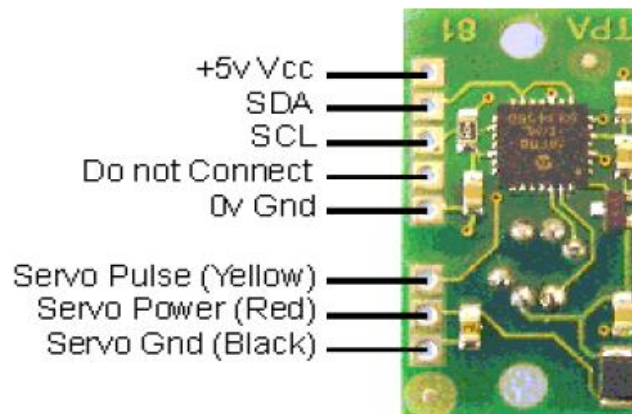
II.3.3 Sensor Suhu Infra Merah (*Thermal Array TPA81*)^[17]

TPA81 adalah sebuah *thermopile array* yang mendeteksi inframerah pada panjang gelombang $2\mu\text{m}$ - $22\mu\text{m}$. *Thermal Array* berguna untuk mengukur sumber panas yang berbeda dalam sekali pengambilan data suhu pada suatu jangkauan daerah tertentu, yaitu sekitar 41° yang dibagi ke dalam delapan titik (*pixel*). Pengukuran berbentuk baris dengan masing-masing titik memiliki sudut deteksi sekitar 6° yang ditunjukkan pada Gambar 2.18. TPA81 dapat mengukur suhu sebuah nyala api lilin sampai 2m. Selain dapat mengukur suhu, sensor ini juga mempunyai keluaran untuk menggerakkan motor servo. Gerakan motor servo ini dimaksudkan agar sensor ini dapat bergerak ke kiri dan ke kanan sewaktu mengukur suhu objek yang berada di lingkungannya.



Gambar 2.18 Sudut Pandang TPA81

TPA 81 memiliki bentuk dengan ukuran 3.4 x 1.7cm terdiri dari 8 buah kaki pin seperti Gambar 2.19.



Gambar 2.19 Kaki Pin TPA81

II.3.3.1 Komunikasi pada TPA 81 ^[17]

Komunikasi pada TPA81 menggunakan I2C bus. Jalur SCL dan SDA masing-masing harus memiliki resistor pull-up dengan nilai 1.8k ohm.

Pengalamatan TPA81 dapat diatur ke dalam delapan buah pengalaman I2C bus. Pengalaman itu terdiri dari 0xD0, 0xD2, 0xD4, 0xD6, 0xD8, 0xDA, 0xDC, dan 0xDE. Secara *default*, TPA81 diberi pengalaman asal 0xD0.

II.3.3.2 Register pada TPA 81 ^[17]

TPA 81 memiliki 10 macam register, ditunjukkan pada Tabel 2.1.

Tabel 2.1 Register pada TPA 81

| Register | Read | Write |
|----------|------------------------|---------------------------------|
| 0 | Software Revision | Command Register |
| 1 | Ambient Temperature °C | Servo Range (V6 or higher only) |
| 2 | Pixel 1 Temperature °C | N/A |
| 3 | Pixel 2 | N/A |
| 4 | Pixel 3 | N/A |
| 5 | Pixel 4 | N/A |
| 6 | Pixel 5 | N/A |
| 7 | Pixel 6 | N/A |
| 8 | Pixel 7 | N/A |
| 9 | Pixel 8 | N/A |

Hanya register 1 dan 0 yang dapat digunakan untuk *write* pada komunikasi I2C. Register 0 adalah *command register* dan digunakan untuk mengatur posisi servo dan mengubah pengalaman TPA81 pada komunikasi I2C, tidak dapat digunakan untuk *read*. Sedangkan register 1 membaca suhu lingkungan.

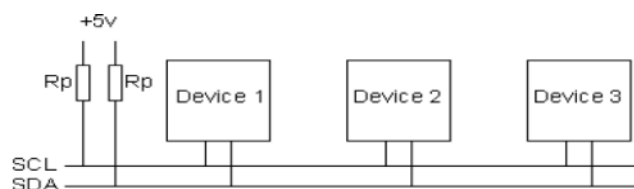
Terdapat sembilan temperatur yang dapat terbaca TPA81, semua dalam derajat Celcius (°C) . Register 1 adalah suhu lingkungan sekitar sensor. Register 2 - 9 adalah suhu-suhu pada 8 titik, seperti pada Gambar 2.18.

II.4 I2C (*Inter-Integrated Circuit*) ^[16]

I2C merupakan singkatan dari *Inter-Integrated Circuit*, yang disebut dengan *I-squared-C* atau *I-two-C*. I2C merupakan protokol yang digunakan pada *multi-master serial computer bus* yang dikembangkan oleh Philips. Protokol ini untuk saling berkomunikasi antar perangkat *low-speed* lainnya yang diaplikasikan pada *motherboard*, *embedded system*, atau *cellphone*. Jalur I2C bus hanya

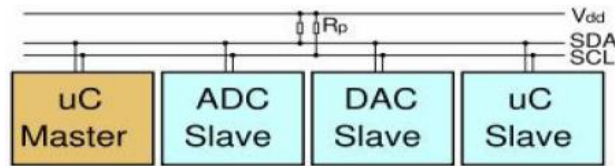
merupakan dua jalur yang disebut dengan *SDA line* dan *SCL line*, dengan *SCL line* merupakan jalur untuk *clock* dan *SDA line* merupakan jalur untuk data. Semua peralatan yang akan digunakan, dihubungkan seluruhnya pada jalur *SDA line* dan *SCL line* dari I2C tersebut. Jenis komunikasi yang dilakukan antar perangkat dengan menggunakan protokol I2C mempunyai sifat *serial synchronous half duplex bidirectional*, dengan data yang ditransmisikan dan diterima hanya melalui satu jalur data *SDA line* (bersifat *serial*). Setiap penggunaan jalur data bergantian antar perangkat (bersifat *halfduplex*) dan data dapat ditransmisikan dari dan ke sebuah perangkat (bersifat *bidirectional*).

Sumber *clock* yang digunakan pada I2C *bus* hanya berasal dari satu perangkat *master* melalui jalur *clock SCL line* (bersifat *synchronous*). Kedua jalur *SDA* dan *SCL* merupakan *driver* yang bersifat *open drain*, yang berarti bahwa IC yang digunakan dapat men-*drive* keluaran *low*, tetapi tidak dapat men-*drive* menjadi *high*, untuk dapat mendapatkan data *high* maka harus menyediakan resistor *pull-up* pada tegangan *power supply* sebesar 5 volt terhadap jalur *SDA* dan *SCL*. Jika tidak dipasang resistor *pull-up*, jalur *SDA* dan *SCL* akan mendekati *low* (mendekati 0 volt) dan jalur I2C *bus* tidak dapat bekerja. Nilai resistor yang dapat digunakan berkisar 1K8 ohm hingga 47K ohm. Satu set resistor *pull-up* sudah cukup untuk digunakan pada semua jalur I2C *bus*, pemasangan resistor *pull-up* dapat dilihat pada Gambar 2.20.



Gambar 2.20 Pemasangan Resistor *Pull-Up* pada I2C *Bus*

Semua perangkat yang terdapat dalam jalur I2C *bus* merupakan perangkat *slave* dan *master*, ditunjukkan pada Gambar 2.21, dengan *master* merupakan perangkat yang berfungsi sebagai pengontrol alamat dan data, juga sebagai sumber *clock* bagi perangkat-perangkat *slave* yang terdapat dalam jalur I2C *bus*.

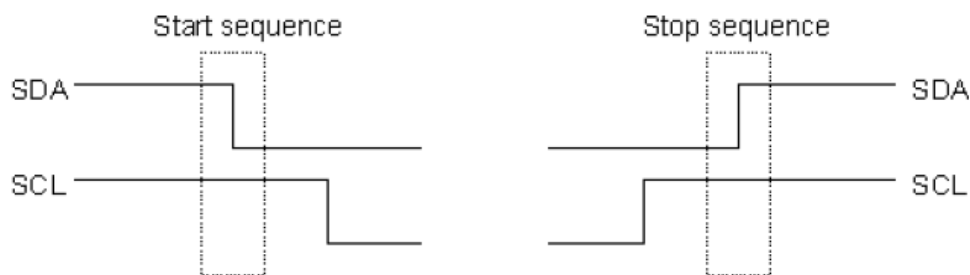


Gambar 2.21 Perangkat pada Jalur I2C Bus

Dalam sebuah jalur I2C bus yang sama memungkinkan penggunaan beberapa perangkat *slave* dan beberapa perangkat *master*. Jika menggunakan *multiple* perangkat *master* dalam sebuah jalur I2C bus maka tiap-tiap perangkat *master* menggunakan jalur secara bergantian.

Kecepatan *transfer* data dari protokol I2C ditentukan oleh besarnya *clock speed* yang digunakan pada jalur SCL. Kecepatan *clock* normal yang diberikan pada jalur SCL sebesar 100kHz. Philips sebagai pembuat protokol I2C membuat standar kecepatan I2C lainnya yaitu *Fast Mode* yang mempunyai kecepatan clock sebesar 400kHz dan *High Speed Mode* yang mempunyai kecepatan hingga 3.4MHz.

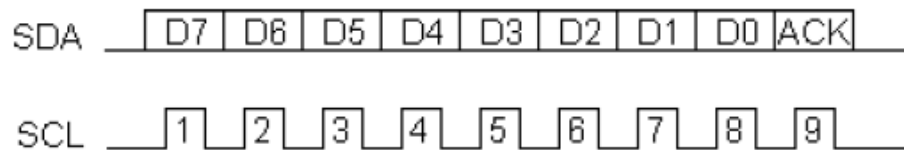
Untuk melakukan transmisi data pada sebuah jalur I2C bus, dimulai dengan mengirimkan sebuah *start sequence* dan diakhiri dengan mengirimkan *stop sequence*. *Start sequence* dan *stop sequence* menandakan awal dan akhir dari proses transmisi data dengan perangkat lainnya dalam sebuah jalur I2C bus, ditunjukkan pada Gambar 2.22.



Gambar 2.22 Start-Stop Sequence pada Transmisi I2C

Transmisi data antar perangkat terjadi setelah *start sequence* dan sebelum *stop sequence*. Data yang ditransmisikan sejumlah 8 bit, MSB (*Most Significant*

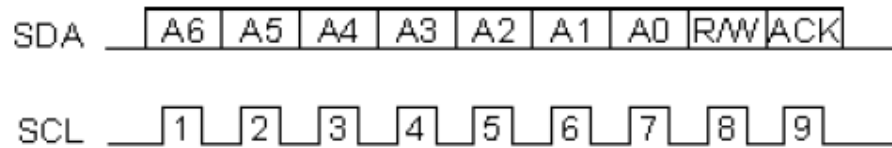
Bit) yang dikirimkan terlebih dahulu hingga LSB (*Least Significant Bit*), kemudian selalu terdapat tambahan satu bit sebagai *Acknowledgement bit* (ACK *bit*). ACK *bit* digunakan untuk mengetahui kondisi transmisi data, jika ACK *bit* berupa kondisi *low* maka perangkat yang ada sudah menerima data dan siap untuk menerima data selanjutnya, sedangkan ACK *bit* berupa kondisi *high* maka perangkat yang ada sudah tidak dapat melakukan transmisi data dan *master* harus mengirimkan *stop sequence* untuk menghentikan komunikasi yang sedang berlangsung. Pada saat berlangsungnya komunikasi antar perangkat dalam sebuah jalur I2C bus, bit data dikirimkan pada saat jalur SCL dalam kondisi *high* dan pergantian bit data terjadi pada saat jalur SCL dalam kondisi *low*, seperti terlihat pada Gambar 2.23.



Gambar 2.23 Kondisi Jalur SDA dan Jalur SCL pada Pengiriman Data

Pada sebuah jalur I2C bus digunakan untuk mengontrol beberapa perangkat *slave* dengan menggunakan sebuah perangkat *master*. Setiap perangkat *slave* pada jalur I2C bus memiliki alamat I2C yang berbeda-beda. Jumlah pengalamatan yang umumnya digunakan pada sebuah protokol I2C sebesar 7 bit, sehingga pada sebuah jalur I2C bus dapat digunakan perangkat *slave* sebanyak 2^7 perangkat dengan alamat anantara 0 sampai 127. Pada saat mengirimkan 7 bit alamat bit R/W (*Read/Write*) digunakan untuk memberitahu alamat perangkat *slave* yang sedang melakukan komunikasi dengan perangkat *master*. *Read* berarti perangkat master akan melakukan pengiriman data pada perangkat *slave* yang dituju, sedangkan *write* berarti perangkat master akan melakukan pengiriman data ke perangkat *slave* yang dituju. Untuk melakukan *read* maka pada bit R/W diberikan kondisi logika *high* dan logika *low* saat melakukan *write*. Pengiriman

alamat perangkat slave pada sebuah *sequence* protokol I2C dapat dilihat pada Gambar 2.24.



Gambar 2.24 Pengiriman Alamat *Slave* pada Sebuah *Sequence* Protokol I2C

Untuk melakukan *write* ke sebuah perangkat *slave* maka langkah-langkah yang harus dilakukan oleh sebuah perangkat master adalah sebagai berikut:

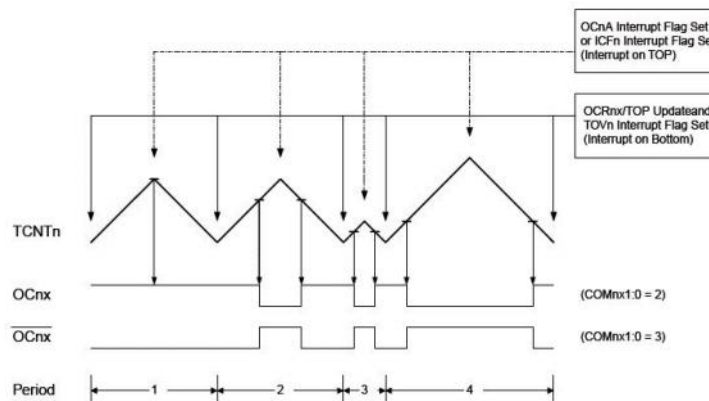
1. Mengirimkan *start sequence*
2. Mengirimkan alamat perangkat *slave* dengan *bit R/W low*
3. Mengirimkan (*write*) *command register* yang diinginkan
4. Mengirimkan (*write*) *data byte* ke perangkat *slave*
5. [Optional, mengirimkan (*write*) *data bytes* lainnya]
6. Mengirimkan *stop sequence*

Untuk melakukan *read* pada sebuah perangkat *slave*, pertama kali perangkat *master* harus memberitahukan *internal address* perangkat *slave* yang ingin dibaca. Jadi untuk melakukan *read* dari sebuah *slave*, sebenarnya dimulai dengan melakukan *write* pada perangkat *slave* tersebut, seperti langkah-langkah berikut:

1. Mengirimkan *start sequence*
2. Mengirimkan alamat perangkat *slave* dengan *bit R/W low*
3. Mengirimkan (*write*) *command register* yang diinginkan
4. Mengirimkan *start sequence* kembali (*repeated start*)
5. Mengirimkan alamat perangkat *slave* dengan *bit R/W high*
6. membaca (*read*) *data byte* dari perangkat *slave*
7. [Optional, membaca (*read*) *data bytes* lainnya]
8. Mengirimkan *stop sequence*

II.5 PWM (*Pulse Width Modulation*) pada AVR ^[12]

ATmega16 dan ATTINY2313 memiliki empat kanal PWM. Dua kanal PWM dibangun dari *counter/timer* 8 bit dan dua kanal PWM dibangun dari *counter/timer* 16 bit. Pada bagian ini hanya dibahas penggunaan PWM dengan *counter/timer* 16 bit dan fungsi PWM sebagai *Phase and Frequency Correct* PWM. Pada Gambar 2.25 ditunjukkan cara kerja mode *Phase and Frequency Correct* PWM. *Phase and Frequency Correct* PWM dibangun dari *dual-slope counter* (pencacah). Nilai pencacah (TCNT) akan bergerak naik dari BOTTOM (0x0000) hingga TOP (ICR1) lalu bergerak turun dari TOP hingga BOTTOM dan begitu seterusnya. Bila fungsi *interrupt on TOP* diaktifkan maka *flag* interupsi akan bernilai logik 1 saat *counter* mencapai nilai TOP, sedangkan bila fungsi *interrupt on BOTTOM* diaktifkan maka *flag* interupsi akan diaktifkan saat *counter* mencapai nilai BOTTOM.



Gambar 2.25 *Phase & Frequency Correct* PWM

Pada pilihan mode *non-inverting*, pulsa PWM digambarkan oleh grafik OCnx. Pulsa keluaran sinyal PWM akan bernilai logika 0 saat pencacah mencapai nilai pembanding (OCRnx) ketika penghitungan bergerak naik. Sebaliknya, keluaran akan bernilai logika 1 saat pencacah mencapai nilai pembanding (OCRnx) ketika penghitungan bergerak turun. Besarnya frekuensi pulsa keluaran

PWM yang digunakan menentukan besarnya nilai TOP (ICR1). Nilai TOP didapat dari Persamaan 2.2.

$$f_{pwm} = \frac{f_{I/O}}{2 \cdot N \cdot TOP} \dots\dots\dots(2.2)$$

Frekuensi I/O adalah frekuensi yang mengendalikan kerja pin *input* dan *output*. Frekuensi ini dapat bersumber dari osilator kristal ataupun rangkaian *clock internal*. Frekuensi PWM adalah frekuensi pulsa keluaran PWM. Nilai N adalah nilai *prescaler* dengan kelipatan 8 yaitu 1, 8, 64, 256 atau 1024. Nilai n pada OCRnx melambangkan *timer/counter* yang digunakan apakah itu *timer/counter0* (8 bit), *timer/counter1* (16 bit), atau *timer/counter2* (8 bit). Nilai x melambangkan kanal *output* A atau kanal *output* B.

II.6 Pengontrol Mikro ^[1]

Pengontrol mikro dapat diartikan sebagai pengontrol dalam ukuran mikro. Secara umum pengontrol mikro dapat diartikan sebagai komputer dalam sebuah chip. Berbagai unsur seperti prosesor, memori, *input*, dan *output* terintegrasi dalam satu kemasan yang berukuran kecil. Pengontrol mikro membutuhkan daya yang rendah, murah, dan mudah didapatkan. Pengontrol mikro beroperasi dengan kecepatan detak (*clocking*) megahertz atau kurang.

II.6.1 Pengenalan ATMEL AVR RISC ^[1]

Salah satu pengontrol mikro yang banyak digunakan saat ini yaitu pengontrol mikro AVR. AVR adalah pengontrol mikro RISC (*Reduce Instruction Set Computing*) 8 bit berdasarkan arsitektur Harvard, yang dibuat oleh Atmel pada tahun 1996. AVR mempunyai kepanjangan *Advanced Versatile RISC* atau *Alf and Vegard's RISC processor* yang berasal dari nama penemunya, dua mahasiswa Norwegian Institute of Technology (NTH), yaitu Alf-Egil Bogen dan Vegard Wollan.

AVR memiliki keunggulan dibandingkan dengan pengontrol mikro lain, keunggulan utama pengontrol mikro AVR adalah memiliki laju putaran eksekusi program yang lebih cepat karena sebagian besar instruksi dieksekusi dalam satu siklus *clock*, lebih cepat dibandingkan dengan pengontrol mikro MCS51 yang memiliki arsitektur CISC (*Complex Instruction Set Computing*) karena pengontrol mikro MCS51 membutuhkan 12 siklus *clock* untuk mengeksekusi 1 instruksi. Selain itu pengontrol mikro AVR memiliki fitur yang lengkap (*ADC Internal, EEPROM Internal, Timer/Counter, Watchdog Timer, PWM, Port I/O, komunikasi serial, komparator, I2C, dll.*), sehingga dengan fasilitas yang lengkap ini, pemrogram dapat menggunakannya untuk berbagai aplikasi sistem elektronika seperti robot, otomasi industri, peralatan telekomunikasi, dan berbagai keperluan lain. Secara umum pengontrol mikro AVR dapat dikelompokkan menjadi tiga kategori, yaitu keluarga AT90Sxx, ATmega, dan ATtiny.

II.6.2 Pengontrol Mikro ATMEGA16^[12]

Pengontrol mikro ATMEGA16 adalah salah satu pengontrol mikro yang dibuat oleh Atmel Corporation, industri yang bergerak di bidang manufaktur semikonduktor. Pengontrol mikro ini termasuk dalam keluarga AVR 8-bit RISC yang dikategorikan dalam kelas Atmega AVR. Angka 16 dalam ATMEGA16 menandakan bahwa pengontrol mikro ini memiliki kapasitas memori flash sebesar 16 KiloByte.

II.6.2.1 Fitur ATMEGA16^[12]

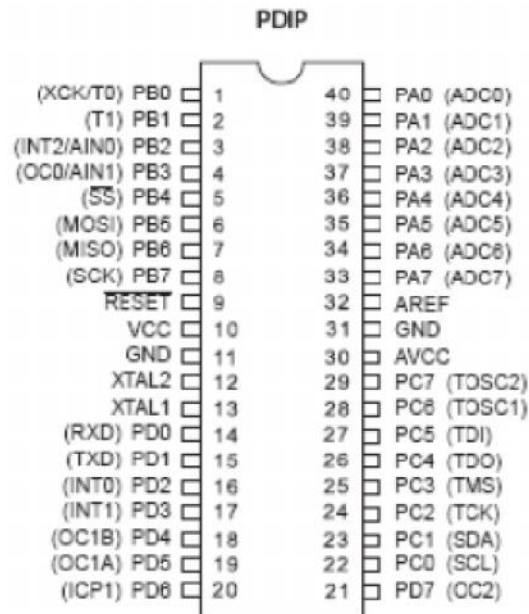
Fitur – fitur yang dimiliki ATMEGA16 adalah sebagai berikut :

1. Pengontrol mikro AVR 8 bit yang memiliki kemampuan tinggi dengan daya rendah.
2. Arsitektur RISC dengan *throughput* mencapai 16 MIPS (*Millions Of Instruction per Second*) pada frekuensi 16 MHz.
3. Memiliki kapasitas *flash* memori 16 kByte, EEPROM (*Electrically Erasable Programmable Read Only Memory*) 512 Byte, dan SRAM (*Static Random Access Memory*) 1kByte.

4. Saluran I/O sebanyak 32 buah, yaitu Port A, Port B, Port C, dan Port D.
5. CPU yang terdiri atas 32 buah register.
6. Unit interupsi internal dan eksternal.
7. Port USART (*Universal Synchronous Asynchronous Receiver Transmitter*) untuk komunikasi serial.
8. Fitur *Peripheral*, yaitu :
 - c) Tiga buah *Timer/Counter* dengan kemampuan, yaitu :
 - o Dua buah *Timer/Counter* 8 bit dengan *prescaler* terpisah dan *mode compare*.
 - o Satu buah *Timer/Counter* 16 bit dengan *prescaler* terpisah, *mode compare*, dan *mode capture*.
 - d) *Real Time Counter* dengan *oscillator* tersendiri.
 - e) Empat kanal PWM.
 - f) Delapan kanal ADC (*Analog To Digital Converter*) 10 bit.
 - g) *Byte-oriented Two-wire Serial Interface*.
 - h) Serial USART yang dapat diprogram.
 - i) Antarmuka SPI (*Serial Peripheral Interface*).
 - j) *Watchdog Timer* dengan *oscillator* internal.
 - k) *On-chip Analog Comparator*.

II.6.2.2 Konfigurasi Pin ATMEGA16 ^[12]

Konfigurasi pin ATMEGA16 dengan kemasan 40 PIN DIP (*Dual In – line Package*) ditunjukkan pada Gambar 2.26.



Gambar 2.26 Konfigurasi Pin ATMEGA16

Dari Gambar 2.26 dapat dijelaskan fungsi dari masing-masing pin ATmega16 sebagai berikut :

1. VCC merupakan pin yang berfungsi sebagai masukan catu daya.
2. GND merupakan pin *ground*.
3. Port A (PA0..PA7) merupakan pin *input / output* dua arah dan pin masukan ADC.

Port B (PB0..PB7) merupakan pin *input / output* dua arah dan pin fungsi khusus, seperti yang ditunjukkan pada Tabel 2.2.

Tabel 2.2 Fungsi Khusus Port B

| Pin | Fungsi Khusus |
|-----|---|
| PB7 | SCK (SPI Bus Serial Clock) |
| PB6 | MISO (SPI Bus Master Input / Slave Output) |
| PB5 | MOSI (SPI Bus Master Output / Slave Input) |
| PB4 | SS (SPI Slave Select Input) |
| PB3 | AIN1 (Analog Comparator Negative Input) OC0 (Timer/Counter0 Output Compare Match Output) |
| PB2 | AIN2 (Analog Comparator Positive Input) INT2 (External Interrupt 2 Input) |
| PB1 | T1 (Timer/Counter1 External Counter Input) |
| PB0 | T0 (Timer/Counter0 External Counter Input) XCK (USART External Clock Input / Output) |

4. Port C (PC0..PC7) merupakan pin *input / output* dua arah dan pin fungsi khusus, seperti yang ditunjukkan pada Tabel 2.3.

Tabel 2.3 Fungsi Khusus Port C

| Pin | Fungsi Khusus |
|-----|--|
| PC7 | TOSC2 (Timer Oscillator Pin 2) |
| PC6 | TOSC1 (Timer Oscillator Pin 1) |
| PC5 | TD1 (JTAG Test Data In) |
| PC4 | TD0 (JTAG Test Data Out) |
| PC3 | TMS (JTAG Test Mode Select) |
| PC2 | TCK (JTAG Test Clock) |
| PC1 | SDA (Two-wire Serial Bus Data Input / Output Line) |
| PC0 | SCL (Two-wire Serial Bus Clock Line) |

5. Port D (PD0..PD7) merupakan pin *input / output* dua arah dan pin fungsi khusus, seperti yang ditunjukkan pada Tabel 2.4.

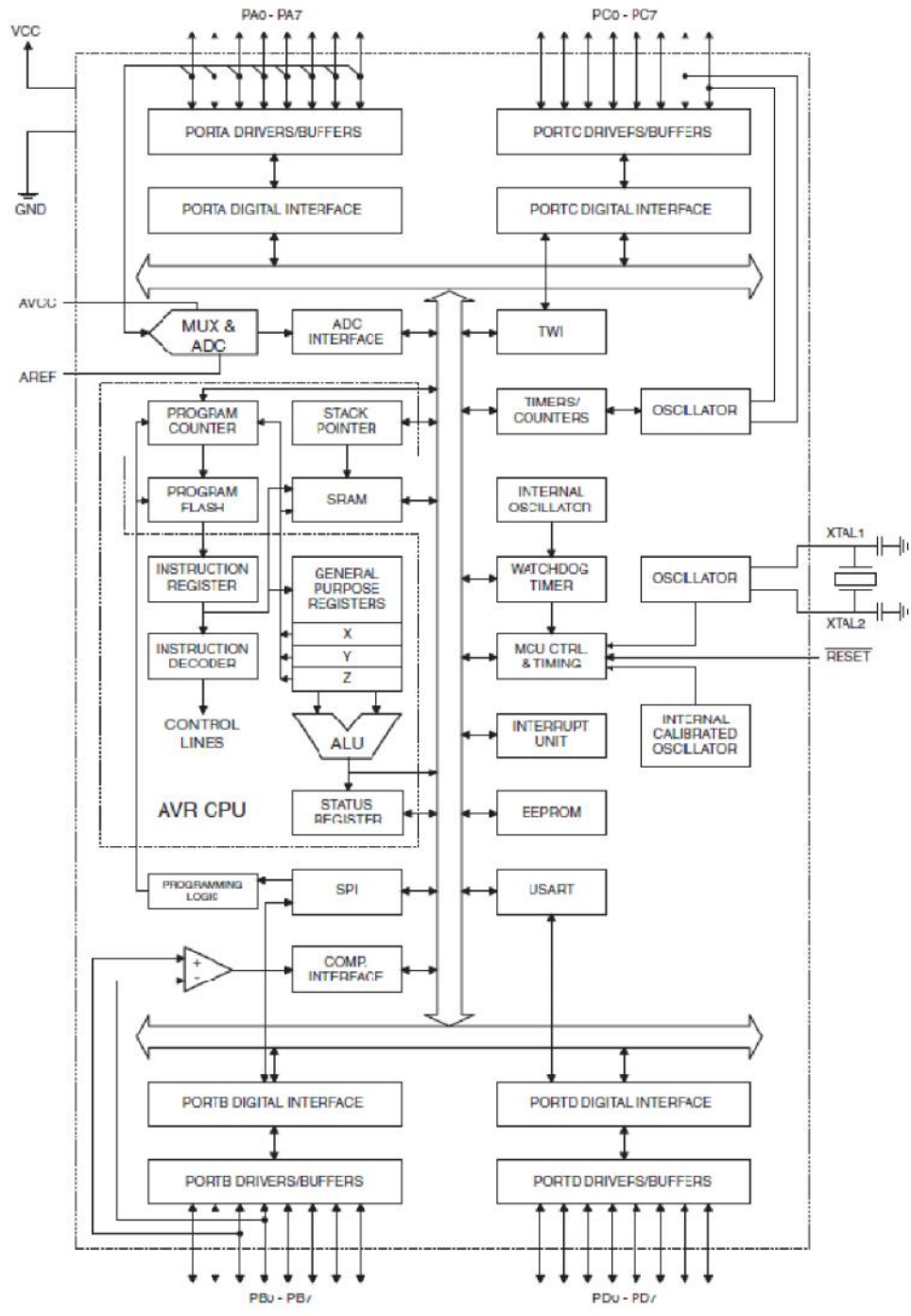
Tabel 2.4 Fungsi Khusus Port D

| Pin | Fungsi Khusus |
|-----|---|
| PD7 | OC2 (Timer/Counter2 Output Compare Match Output) |
| PD6 | ICP (Timer/Counter1 Input Capture Pin) |
| PD5 | OC1A (Timer/Counter1 Output Compare A Match Output) |
| PD4 | OC1B (Timer/Counter1 Output Compare B Match Output) |
| PD3 | INT1 (External Interrupt 1 Input) |
| PD2 | INT0 (External Interrupt 0 Input) |
| PD1 | TXD (USART Output Pin) |
| PD0 | RXD (USART Input Pin) |

6. RESET merupakan pin yang digunakan untuk mereset pengontrol mikro.
7. XTAL1 dan XTAL2 merupakan pin masukan *clock* eksternal.
8. AVCC merupakan pin masukan tegangan untuk ADC.
9. AREF merupakan pin masukan tegangan referensi ADC.

II.6.2.3 Blok Diagram ATmega16 ^[12]

Struktur ATmega16 digambarkan dalam blok diagram yang ditunjukkan pada Gambar 2.27. ATmega16 disusun dari beberapa bagian seperti AVR *central processing unit*, empat *port input / output*, *analog to digital converter*, *analog comparator*, SPI, TWI, rangkaian pewaktu (*timer*), rangkaian *clocking*, kontrol interupsi, EEPROM, dan USART.



Gambar 2.27 Blok Diagram ATMEGA16

II.6.2.4 General Purpose Register ATmega16^[12]

Gambar 2.28 menunjukkan struktur 32 *general purpose register* yang terdapat dalam pengontrol mikro ATmega16, masing–masing *register* ditentukan juga dalam alamat memori data, dipetakan ke dalam 32 lokasi pertama data *user*. Walaupun tidak secara fisik diimplementasikan sebagai lokasi SRAM, namun pengaturan ini memberikan fleksibilitas dalam mengakses *register*, seperti *register pointer* X, Y, dan Z dapat diset menuju *index* dari semua *register file*.

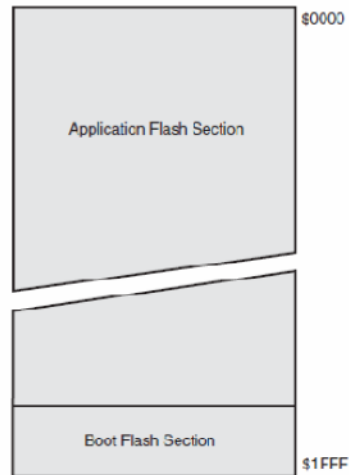
| | 7 | 0 | Addr. | |
|--|-----|---|-------|----------------------|
| General Purpose Working Registers | R0 | | \$00 | |
| | R1 | | \$01 | |
| | R2 | | \$02 | |
| | ... | | | |
| | R13 | | \$0D | |
| | R14 | | \$0E | |
| | R15 | | \$0F | |
| | R16 | | \$10 | |
| | R17 | | \$11 | |
| | ... | | | |
| | R26 | | \$1A | X-register Low Byte |
| | R27 | | \$1B | X-register High Byte |
| | R28 | | \$1C | Y-register Low Byte |
| | R29 | | \$1D | Y-register High Byte |
| | R30 | | \$1E | Z-register Low Byte |
| | R31 | | \$1F | Z-register High Byte |

Gambar 2.28 General Purpose Register ATmega16

II.6.2.5 Peta Memori ATmega16^[12]

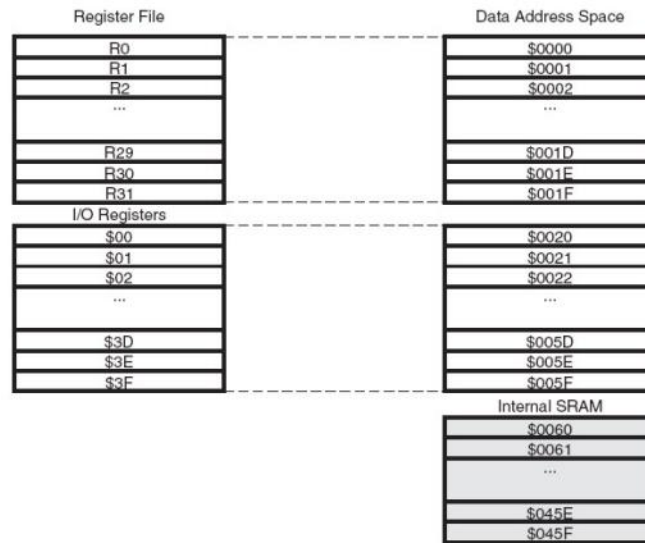
Arsitektur AVR mempunyai dua memori utama yaitu memori data dan memori program. Selain itu, ATmega16 juga memiliki memori EEPROM untuk menyimpan data.

ATmega16 memiliki 16 KByte *On-Chip In-System Reprogrammable Flash memory* untuk menyimpan program. Semua instruksi AVR memiliki format 16 atau 32 bit, *flash* diatur dalam 8 K *word* x 16 bit/*word*. Untuk keamanan program, memori program flash dibagi ke dalam 2 bagian, yaitu bagian program *boot* dan aplikasi. *Bootloader* adalah program kecil yang bekerja pada saat *start up time* yang dapat memasukan seluruh program aplikasi ke dalam memori prosesor. Pada Gambar 2.29 ditunjukkan peta memori program ATmega16.



Gambar 2.29 Pemetaan Memori ATMEGA16

Memori data ATmega16 terbagi menjadi tiga bagian, yaitu 32 *register* umum, 64 buah *register* I/O dan 1 KByte SRAM *internal*. *General purpose register* menempati alamat data terbawah, yaitu \$00 sampai \$1F, sedangkan memori I/O menempati 64 alamat berikutnya mulai dari \$20 hingga \$5F. Memori I/O merupakan *register* yang khusus digunakan untuk mengatur fungsi terhadap berbagai *peripheral* pengontrol mikro seperti contoh *register*, *timer/counter*, fungsi-fungsi I/O, dan sebagainya. 1024 alamat memori berikutnya mulai alamat \$60 hingga \$45F digunakan untuk SRAM *internal*. Pada Gambar 2.30 ditunjukkan peta memori data ATmega16.



Gambar 2.30 Pemetaan Memori Data ATMEGA16

ATmega16 terdiri dari 512 Byte memori data EEPROM, data dapat tulis atau baca dari memori ini, ketika catu daya dimatikan, data terakhir yang ditulis pada memori EEPROM masih tersimpan pada memori ini, atau dengan kata lain memori EEPROM bersifat *non-volatile*.

II.6.2.6 Pin Input/Output ATMEGA16 ^[12]

ATMEGA16 memiliki 32 *pin input/output* yang terbagi menjadi empat *port*. Setiap *pin* dapat menjadi *input* ataupun *output* tergantung inisialisasi yang diberikan. Ada tiga variabel yang berperan penting dalam inisialisasi dan penggunaan *port* sebagai *input/output*, yaitu *PORTxn*, *DDRxn* dan *PINxn*. Nilai x mewakili masing-masing *port* (A-D) dan n mewakili nilai *pin* (0-7).

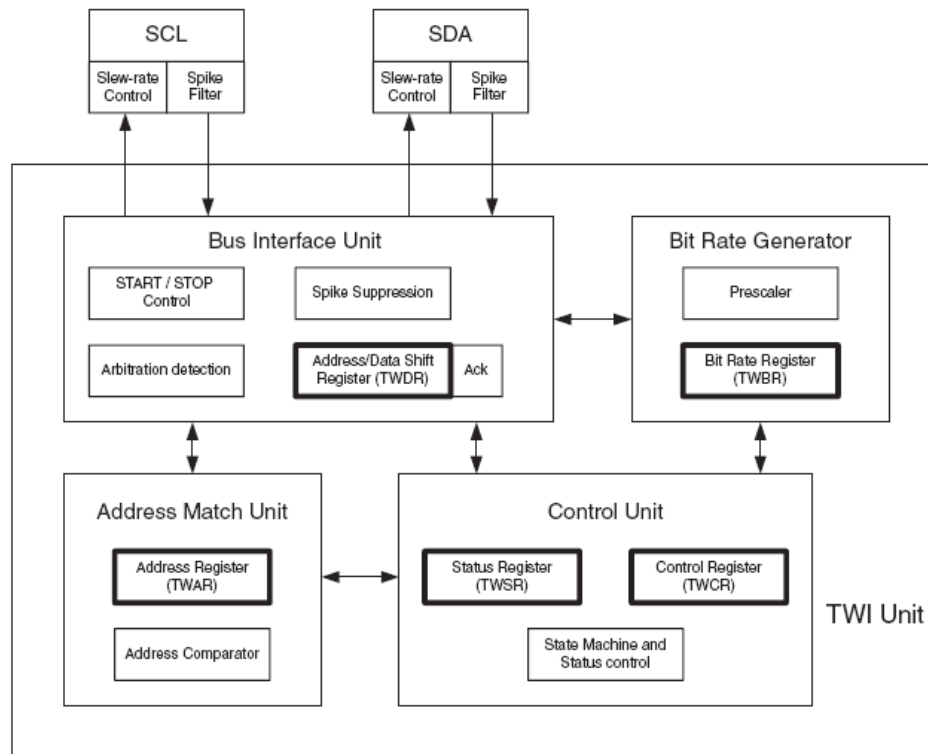
Tabel 2.5 Konfigurasi *Port* ATMEGA16

| DDRxn | PORTxn | I/O | Pull-Up |
|-------|--------|---------------|---------|
| 0 | 0 | <i>Input</i> | Tidak |
| 0 | 1 | <i>Input</i> | Ya |
| 1 | 0 | <i>Output</i> | Tidak |
| 1 | 1 | <i>Output</i> | Tidak |

Tabel 2.5 adalah Konfigurasi *port* dari ATMEGA16. DDR_{xn} digunakan untuk menentukan penggunaan *pin* ATMEGA16 yaitu sebagai *input* atau *output*. Bila DDR_{xn} bernilai logika 1 maka *pin* tersebut berlaku sebagai *output*, juga sebaliknya bila bernilai 0 maka *pin* akan bertindak sebagai *input*. Jika *pin* dibuat sebagai *output* dan nilai logika PORT_{xn}-nya sama dengan 1, maka PIN_{xn}-nya akan memberikan logika 1. Sebaliknya, jika nilai logika PORT_{xn}-nya sama dengan 0, maka PIN_{xn}-nya akan memberikan logika 0. Jika *pin* dibuat sebagai *input* dan nilai logika PORT_{xn}-nya sama dengan 1, maka resistor *pull-up pin* akan diaktifkan. Tapi untuk menon-aktifkan resistor *pull-up pin* maka nilai logika PORT_{xn}-nya harus dibuat 0 atau *pin* dijadikan sebagai *output* (DDR_{xn} = 1).

II.6.2.7 I2C (Inter-Integrated Circuit) ATMEGA16^[12]

I2C pada pengontrol mikro sering disebut juga Two-Wire Serial Interface (TWI). Pada ATMEGA16, perangkat yang digunakan maksimal 128 berbeda dengan menggunakan two bi-directional bus lines (SDA dan SCL). Modul TWI merupakan kompresi dari beberapa modul, ditunjukkan pada Gambar 2.31.



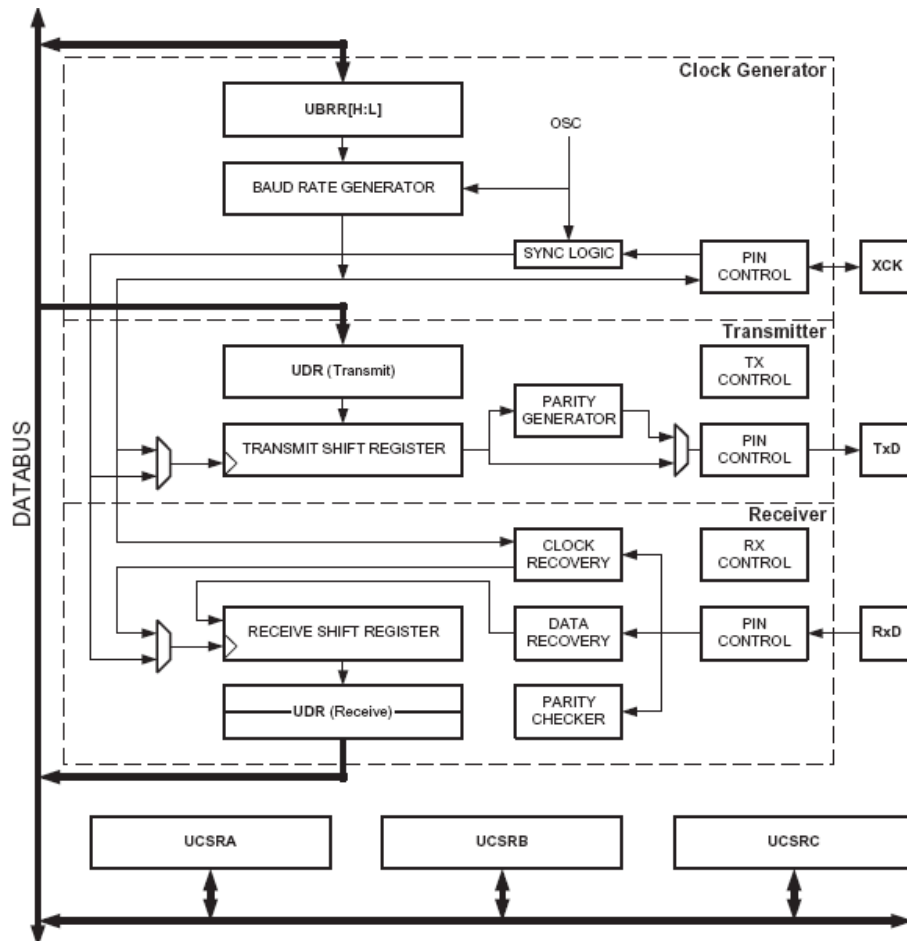
Gambar 2.31 Gambaran Modul TWI keseluruhan

I2C (SDA dan SCL) pada ATmega16 dapat ditempatkan pada semua pin tetapi dalam Port yang sama. Normalnya Port C yang digunakan, pin C.1 sebagai SCL dan pin C.0 sebagai SDA.

II.6.2.8 USART (The Universal Synchronous and Asynchronous Serial Receiver and Transmitter) ATMEGA16^[12]

USART dapat difungsikan sebagai transmisi data sinkron dan asinkron. Sinkron berarti *clock* yang digunakan antara *transmitter* dan *receiver* mempunyai satu sumber *clock*. Sedangkan asinkron berarti *transmitter* dan *receiver* mempunyai *clock* sendiri-sendiri.

USART terbagi dalam tiga blok yaitu *clock generator*, *transmitter*, dan *receiver*. Ketiga blok tersebut dipetakan seperti Gambar 2.32.



Gambar 2.32 Blok USART

Clock Generator pada Gambar 2.32 berfungsi sebagai pengatur kecepatan data (*baud rate*), register yang bertugas menentukan *baud rate* adalah register pasangan UBRR.

Table 2.6 Baud Rate

| Operating Mode | Equation for Calculating Baud Rate ⁽¹⁾ | Equation for Calculating UBRR Value |
|--|---|-------------------------------------|
| Asynchronous Normal Mode (U2X = 0) | $BAUD = \frac{f_{OSC}}{16(UBRR + 1)}$ | $UBRR = \frac{f_{OSC}}{16BAUD} - 1$ |
| Asynchronous Double Speed Mode (U2X = 1) | $BAUD = \frac{f_{OSC}}{8(UBRR + 1)}$ | $UBRR = \frac{f_{OSC}}{8BAUD} - 1$ |
| Synchronous Master Mode | $BAUD = \frac{f_{OSC}}{2(UBRR + 1)}$ | $UBRR = \frac{f_{OSC}}{2BAUD} - 1$ |

Table 2.6 menunjukkan perhitungan untuk menentukan besar *baud rate* yang akan digunakan. F_{osc} menunjukkan frekuensi osilasi yang digunakan, dan *baud* sendiri berarti kemampuan transfer *bit* per detik.

USART *transmitter* berhubungan dengan pengiriman data pada *pin* TX yang berada pada *port* D.1 atau *pin* 15. Register UDR berfungsi sebagai tempat penampungan data yang akan ditransmisikan, sedangkan *flag* TX Control (TXC) memberikan tanda bila data yang ditransmisikan telah sukses. USART *receiver* berhubungan dengan penerimaan data dari *pin* RX yang berada pada *port* D.0 atau *pin* 14.

II.6.3 Pengontrol Mikro ATTINY2313 ^[13]

Sama halnya dengan pengontrol mikro ATMEGA16, pengontrol mikro ATTINY2313 juga merupakan salah satu pengontrol mikro yang dibuat oleh Atmel Corporation, industri yang bergerak di bidang manufaktur semikonduktor. Pengontrol mikro ini termasuk dalam keluarga AVR 8-bit RISC yang dikategorikan dalam kelas ATmegaAVR. Pengontrol mikro ini memiliki kapasitas memori flash sebesar 2 KiloByte.

II.6.3.1 Fitur ATTINY2313 ^[13]

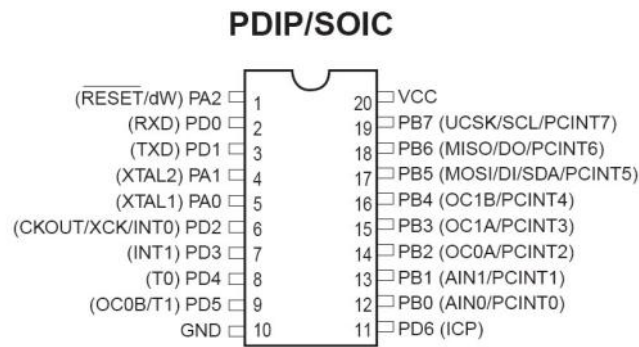
Fitur – fitur yang dimiliki ATTINY2313 adalah sebagai berikut :

1. Pengontrol mikro AVR 8 bit yang memiliki kemampuan tinggi dengan daya rendah.

2. Arsitektur RISC dengan *throughput* mencapai 20 MIPS (*Millions Of Instruction per Second*) pada frekuensi 20 MHz.
3. Memiliki kapasitas *flash* memori 2 KByte, EEPROM (*Electrically Erasable Programmable Read Only Memory*) 128 Byte, dan SRAM (*Static Random Access Memory*) 128 Byte.
4. Saluran I/O sebanyak 18 buah, yaitu Port A, Port B, dan Port D.
5. CPU yang terdiri atas 32 buah register.
6. Unit interupsi internal dan eksternal.
7. Port USART (*Universal Synchronous Asynchronous Receiver Transmitter*) untuk komunikasi serial.
8. Fitur *Peripheral*, yaitu :
 - a) Dua buah *Timer/Counter* dengan kemampuan perbandingan, yaitu :
 - o Satu buah *Timer/Counter* 8 bit dengan *prescaler* terpisah dan *mode compare*.
 - o Satu buah *Timer/Counter* 16 bit dengan *prescaler* terpisah, *mode compare*, dan *mode capture*.
 - b) Empat kanal PWM.
 - c) Antarmuka USI (*Universal Serial Interface*).
 - d) *Watchdog Timer* dengan *oscillator* internal.
 - e) *On-chip Analog Comparator*.
 - f) *Full Duplex* USART (*Universal Synchronous Asynchronous Receiver Transmitter*).

II.6.3.2 Konfigurasi Pin ATTINY2313 ^[13]

ATTINY2313 hanya memiliki dua puluh *pin* secara keseluruhan baik pada kemasan PDIP maupun MLF. *Port* A hanya memiliki tiga *pin*, *port* B memiliki delapan *pin* dan *port* D memiliki tujuh *pin*. Semua *pin* dapat difungsikan sebagai masukan atau keluaran kecuali *pin* VCC dan ground.



Gambar 2.33 Konfigurasi Pin ATTINY2313

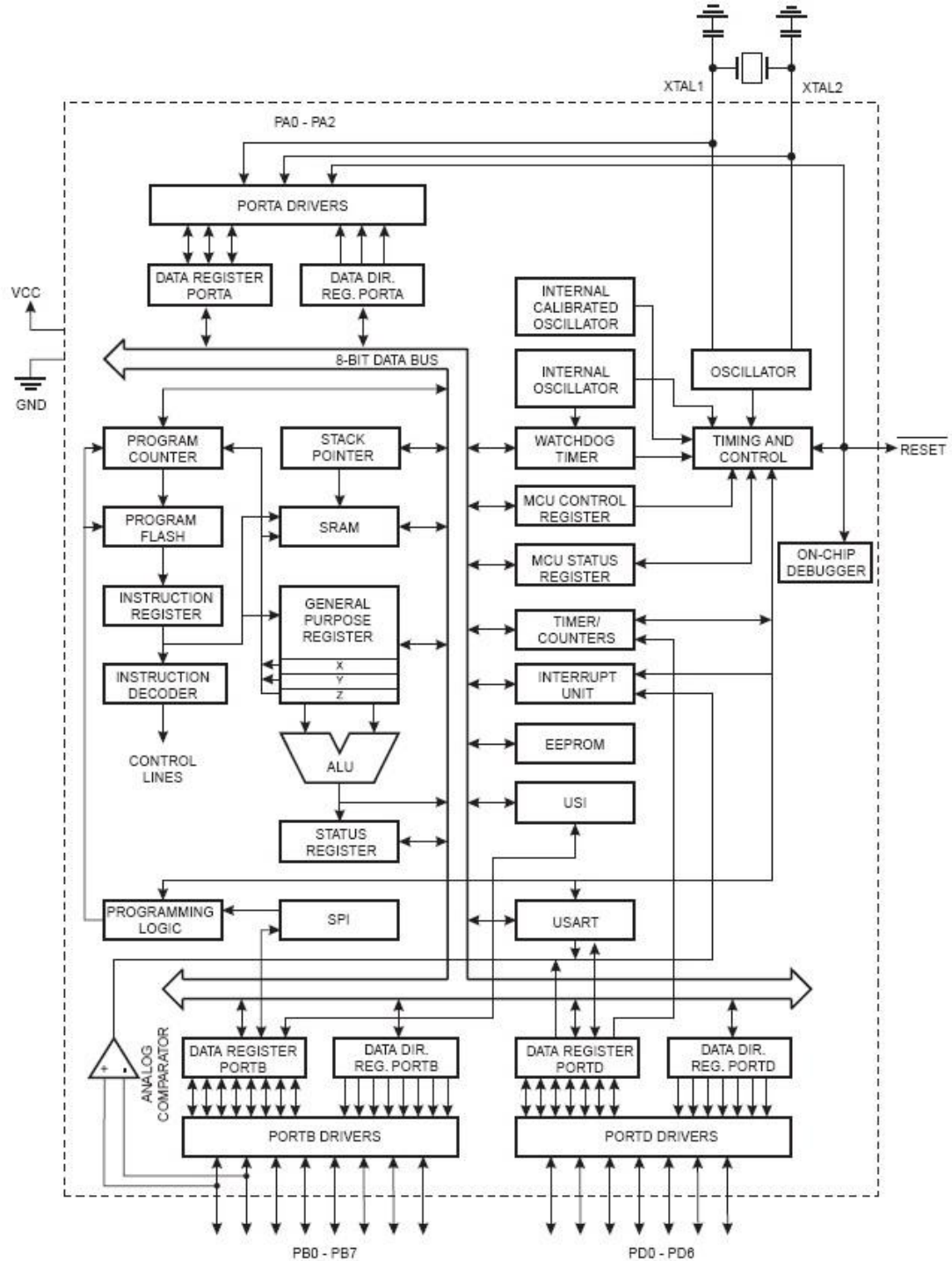
Gambar 2.33 adalah konfigurasi *pin* dari ATTINY2313. Setiap *pin* diberi nama sesuai dengan fungsinya seperti penamaan *pin* mikrokontroler AVR secara umum. Fungsi alternatif setiap *pin* dituliskan dalam tanda kurung.

Keterangan *pin* :

- 1,4,5 : *Port A*, hanya terdiri dari 3 *pin* dengan alternatif fungsi pada *pin* 1 sebagai *reset*, *pin* 4 sebagai *output* inverting osclilator dan *pin* 5 sebagai *input* inverting osclilator serta *input* internal *clocking*.
- 2,3,6-9,11 : *Port D*, terdiri dari 7 *pin*, sebagai *port input* atau *output*.
- 10 : *Input* catu daya negatif dari sumber tegangan.
- 20 : *Input* catu daya positif dari sumber tegangan.
- 12-19 : *Port B*, sebagai *port input* atau *output* dengan fungsi alternatif seperti *pin* 15 dan 16 untuk *output timer* 1A dan 1B.

Struktur ATTINY2313 mirip dengan pengontrol mikro ATMEGA16. Perbedaan yang mencolok yaitu hilangnya *port C* dan *analog to digital converter*. Fitur lainnya seperti *port input/output*, *analog comparator*, SPI, TWI, rangkaian pewaktu (*timer*), rangkaian *clocking*, kontrol interupsi, EEPROM dan USART masih diadopsi oleh ATTINY2313. Arsitektur ATTINY2313 ditunjukkan pada Gambar 2.34. Diagram blok arsitektur ini mirip dengan diagram blok

ATMEGA16. Salah satu fitur yang tidak ditemukan dalam ATTINY2313 adalah *analog to digital converter*.



Gambar 2.34 Blok Diagram ATTINY2313

II.6.3.3 Register dan Memori ATTINY2313 ^[13]

ATTINY2313 memiliki 32 register yang digambarkan pada Gambar 2.35.

| | 7 | 0 | Addr. | |
|--|-----|---|-------|----------------------|
| General Purpose Working Registers | R0 | | \$00 | |
| | R1 | | \$01 | |
| | R2 | | \$02 | |
| | ... | | | |
| | R13 | | \$0D | |
| | R14 | | \$0E | |
| | R15 | | \$0F | |
| | R16 | | \$10 | |
| | R17 | | \$11 | |
| | ... | | | |
| | R26 | | \$1A | X-register Low Byte |
| | R27 | | \$1B | X-register High Byte |
| | R28 | | \$1C | Y-register Low Byte |
| | R29 | | \$1D | Y-register High Byte |
| | R30 | | \$1E | Z-register Low Byte |
| | R31 | | \$1F | Z-register High Byte |

Gambar 2.35 Register ATTINY2313

Gambar 2.35 menunjukkan ATTINY2313 memiliki register 16 bit juga. Register ini juga dikenal dengan nama Register X, Register Y dan Register Z. Selain register terdapat memori yang menyokong kinerja penyimpanan data seperti 2 kilobyte memori flash, 128 byte memori EEPROM, dan 128 byte memori SRAM yang menempati alamat \$0060 hingga \$00DF.

II.6.3.4 Port Input/Output ATTINY2313 ^[13]

ATTINY2313 memiliki maksimum 18 *pin input/output* yang terbagi menjadi tiga *port*. Setiap *pin* dapat menjadi *input* ataupun *output* tergantung inisialisasi yang diberikan. Ada tiga variabel yang berperan penting dalam inisialisasi dan penggunaan *port* sebagai *input/output*, yaitu PORT_{xn}, DDR_{xn} dan PIN_{xn}, dengan x mewakili masing-masing *port* (A, B dan D) dan n mewakili nilai *pin* masing-masing *port*.

Tabel 2.7 Konfigurasi *Port* ATTINY2313

| DDRxn | PORTxn | I/O | Pull-Up |
|-------|--------|---------------|---------|
| 0 | 0 | <i>Input</i> | Tidak |
| 0 | 1 | <i>Input</i> | Ya |
| 1 | 0 | <i>Output</i> | Tidak |
| 1 | 1 | <i>Output</i> | tidak |

Tabel 2.7 adalah Konfigurasi *port* dari ATTINY2313. DDRxn digunakan untuk menentukan penggunaan *pin* yaitu sebagai *input* atau *output*. Bila DDRxn bernilai logika 1 maka *pin* tersebut berlaku sebagai *output*, juga sebaliknya bila bernilai 0 maka *pin* akan bertindak sebagai *input*. Jika *pin* dibuat sebagai *output* dan nilai logika PORTxn-nya sama dengan 1, maka PINxn-nya akan memberikan logika 1. Sebaliknya, jika nilai logika PORTxn-nya sama dengan 0, maka PINxn-nya akan memberikan logika 0. Jika *pin* dibuat sebagai *input* dan nilai logika PORTxn-nya sama dengan 1, maka resistor pull-up *pin* akan diaktifkan. Tapi untuk menon-aktifkan resistor pull-up *pin* maka nilai logika PORTxn-nya harus dibuat 0 atau *pin* dijadikan sebagai *output* (DDRxn = 1).