

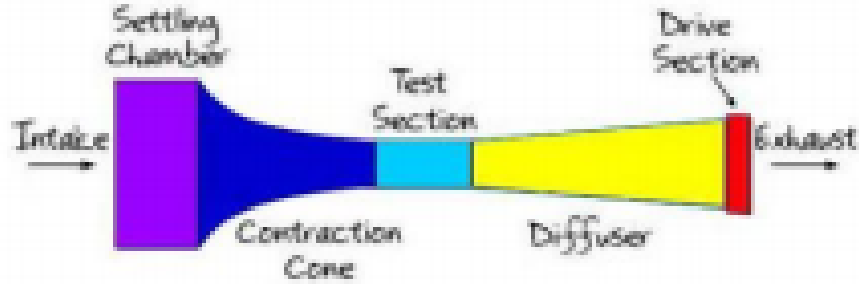
BAB II

TINJAUAN PUSTAKA

2.1 Wind Tunnel

Wind tunnel atau terowongan angin adalah alat riset yang dikembangkan untuk membantu dalam menganalisis efek angin yang bergerak atau di sekitar objek padat. Pada umumnya, perancangan terowongan angin berdasarkan dari data-data hasil eksperimen. *Wind tunnel* sebagai alat uji aerodinamika terdiri dari beberapa bagian penting yaitu bagian seksi uji (*Test Section*), yang transparan, *honeycomb* sebagai penyearah aliran udara yang masuk kedalam seksi uji, kipas penyedot angin berupa motor penggerak dan fan, serta *external balance* sebagai alat ukur gaya yang terjadi pada benda kerja.

Dibagian paling depan terdapat *settling chamber* yang di dalamnya terdapat *honeycomb*. Fungsi *settling chamber* yaitu untuk menyeragamkan aliran udara. Karena aliran turbulen dapat menyebabkan gaya menjadi tidak dapat diperkirakan dan diukur di dalam seksi uji. Di dalam *settling chamber* terdapat *honeycomb*, yang berfungsi untuk mengembangkan atau menghasilkan aliran udara yang halus pada seksi ujinya sehingga aliran udaranya dapat lebih linier. *Honeycomb* ini harus digunakan karena pengaruhnya dalam mengembangkan aliran udara sangat besar. Kemudian udara masuk melalui *contraction cone* yang berfungsi untuk mengambi udara yang memiliki kecepatan tinggi bervolume kecil. Semakin kecil ukuran *contraction cone* semakin tinggi kecepatan udaranya. Setelah itu udara menuju bagian *Test Section* yang berfungsi sebagai tempat untuk menempatkan model sebuah sayap atau pesawat atau benda yang ingin diuji. Saat aliran udara berdasarkan kecepatan yang diinginkan, sensor untuk mengukur gaya dalam hal ini *external balance*, seperti gaya *lift* dan *drag* ditempatkan juga di *test section*. Kemudian oleh *diffuser* laju udara diperlambat sebelum menuju keluaran. Pada *drive section* disediakan gaya yang dapat menyebabkan udara bergerak melewati terowongan. Secara umum bentuk model *wind tunnel* ditunjukkan dalam Gambar 2.1.



Gambar 2.1 Model Wind tunnel Open Circuit

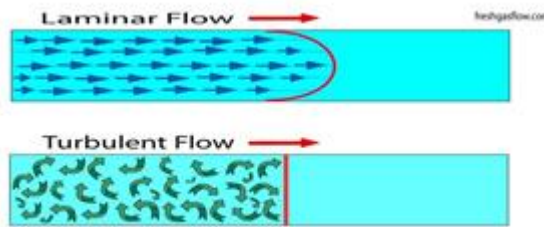
Sumber: Muchammad, 2006

2.2 Tipe Aliran Fluida

Fluida yang bergerak dapat diklasifikasikan ke dalam beberapa kategori. Apakah alirannya steady atau tak steady, apakah fluidanya kompresibel (dapat mampat) atau inkompresibel (tak dapat mampat), apakah fluidanya viskos atau non-viskos, atau apakah aliran fluidanya laminar atau turbulen. Jika fluidanya steady, kecepatan partikel fluida pada setiap titik tetap terhadap waktu. Fluida pada berbagai bagian dapat mengalir dengan laju atau kecepatan yang berbeda, tetapi fluida pada satu lokasi selalu mengalir dengan laju atau kecepatan yang tetap. Aliran fluida dapat dibedakan menjadi aliran laminar dan aliran turbulen, tergantung pada jenis garis alir yang dihasilkan oleh partikel-partikel fluida. Jika aliran dari seluruh partikel fluida bergerak sepanjang garis yang sejajar dengan arah aliran (atau sejajar dengan garis tengah pipa, jika fluida mengalir di dalam pipa), fluida yang seperti ini dikatakan laminar.

Fluida laminar kadang-kadang disebut dengan fluida viskos atau fluida garis alir (streamline). Kata laminar berasal dari bahasa latin lamina, yang berarti lapisan atau plat tipis. Sehingga, aliran laminar berarti aliran yang berlapis-lapis. Lapisan-lapisan fluida akan saling bertindihan satu sama lain tanpa bersilangan seperti dalam Gambar 2.2 (atas).

Jika gerakan partikel fluida tidak lagi sejajar, mulai saling bersilangan satu sama lain sehingga terbentuk pusaran di dalam fluida, aliran yang seperti ini disebut dengan aliran turbulen, seperti yang ditunjukkan dalam Gambar 2.2 (bawah). (Khamdani, F., 2012)



Gambar 2.2 Aliran laminar (atas) dan aliran turbulen (bawah)

Sumber: Khamdani, F., 2012

2.3 Kontroler

Keberadaan kontroler dalam sebuah sistem kontrol mempunyai kontribusi yang besar terhadap perilaku sistem. Pada prinsipnya hal itu disebabkan oleh tidak dapat diubahnya komponen penyusun sistem tersebut. Artinya, karakteristik *plant* harus diterima sebagaimana adanya, sehingga perubahan perilaku sistem hanya dapat dilakukan melalui penambahan suatu sub sistem, yaitu kontroler.

Salah satu fungsi komponen kontroler adalah mengurangi sinyal kesalahan, yaitu perbedaan antara nilai referensi/nilai yang diinginkan dan nilai aktual. Hal ini sesuai dengan tujuan sistem kontrol dimana mendapat nilai sinyal keluaran sama dengan nilai yang diinginkan referensi. Semakin kecil kesalahan yang terjadi, semakin baik kinerja sistem kontrol yang diterapkan.

Apabila perbedaan antara nilai referensi dengan nilai keluaran relatif besar, maka kontroler yang baik seharusnya mampu mengamati perbedaan ini untuk segera menghasilkan sinyal keluaran untuk mempengaruhi *plant*. Dengan demikian sistem secara cepat mengubah keluaran *plant* sampai diperoleh selisih dengan nilai referensi sekecil mungkin.

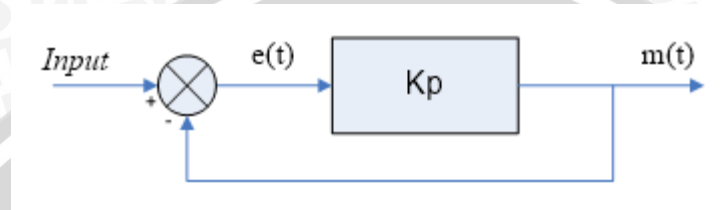
Prinsip kerja kontroler adalah membandingkan nilai aktual keluaran *plant* dengan nilai referensi, kemudian menentukan nilai kesalahan dan akhirnya menghasilkan sinyal kontrol untuk meminimalkan kesalahan (Ogata, K., 1997).

2.3.1 Kontroler Proporsional

Kontroler proporsional memiliki keluaran yang sebanding/proporsional dengan besarnya sinyal kesalahan/*error*. Dapat dikatakan bahwa keluaran kontroler proporsional

merupakan perkalian antara konstanta proporsional dengan masuknya. Perubahan sinyal masukan akan segera menyebabkan sistem secara langsung mengubah keluarannya sebesar konstanta pengalinya.

Untuk lebih jelasnya dapat ditunjukkan pada blok diagram dalam Gambar 2.3.



Gambar 2.3 Diagram blok kontroler proporsional

Sumber: Ogata, K., 1997

Pada Gambar 2.3 menunjukkan blok diagram yang menggambarkan hubungan antara *input* (besaran yang diinginkan), besaran aktual dengan besaran keluaran kontroler proporsional, dan besaran kesalahan (*error*). Sinyal kesalahan (*error*) merupakan selisih antara besaran *setting* dengan besaran aktualnya.

Pada pengendali proporsional hubungan antara keluaran kontroler $m(t)$ dan sinyal kesalahan $e(t)$ adalah sebagai berikut:

$$m(t) = K_p e(t) \quad (2-1)$$

Dengan K_p adalah penguatan proporsional, keluaran $m(t)$ hanya bergantung pada K_p dan *error*, semakin besar *error* maka semakin besar koreksi yang dilakukan. Penambahan K_p akan menaikkan penguatan sistem sehingga dapat digunakan untuk memperbesar kecepatan respon dan mengurangi kesalahan keadaan mantap.

2.3.2 Kontroler Integral

Kontroler integral berfungsi mengurangi kesalahan keadaan mantap yang dihasilkan pada kontroler proporsional sebelumnya. Kalau sebuah *plant* tidak memiliki unsur integrator ($1/s$), kontroler proporsional tidak akan mampu menjamin keluaran sistem dengan kesalahan keadaan mantapnya nol.

Kontroler integral memiliki karakteristik seperti halnya sebuah integral. Keluaran kontroler sangat dipengaruhi oleh perubahan yang sebanding dengan nilai sinyal

kesalahan. Keluaran kontroler ini merupakan jumlahan yang terus menerus dari perubahan masukannya. Kalau sinyal kesalahan tidak mengalami perubahan, keluaran akan menjaga keadaan seperti sebelum terjadinya perubahan masukan. Gambar 2.4 menunjukkan blok diagram kontroler integral.



Gambar 2.4 Blok diagram kontroler integral

Sumber: Ogata, K., 1997

Nilai keluaran kontroler $m(t)$ sebanding dengan integral sinyal kesalahan $e(t)$, Sehingga

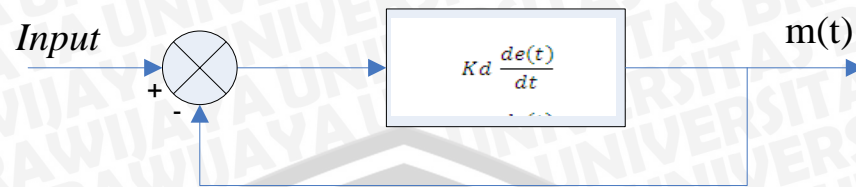
$$\frac{dm(t)}{dt} = Ki \cdot e(t) \quad (2-2)$$

$$m(t) = Ki \int_0^t e(t) dt \quad (2-3)$$

dengan Ki adalah konstanta integral. Jika sinyal kesalahan $e(t)=0$, maka laju perubahan sinyal kendali integral $\frac{dm(t)}{dt} = 0$ atau sinyal keluaran kendali akan tetap berada pada nilai yang dicapai sebelumnya. Aksi kontrol integral digunakan untuk menghilangkan kesalahan posisi dalam keadaan mantap (*error steady state*) tanpa memperhitungkan kecepatan respon.

2.3.3 Kontroler Differensial

Kontroler differensial memiliki sifat seperti halnya suatu operasi differensial. Perubahan yang mendadak pada masukan kontroler, akan mengakibatkan perubahan yang sangat besar dan cepat. Gambar 2.5 berikut menunjukkan blok diagram pada kontroler differensial.



Gambar 2.5 Blok diagram kontroler differensial

Sumber: Ogata, K., 1997

Nilai keluaran kontroler $m(t)$ sebanding laju sinyal kesalahan $\frac{de(t)}{dt}$. Hubungan ini dapat ditulis sebagai:

$$m(t) = K_d \frac{de(t)}{dt} \quad (2-4)$$

Kontroler diferensial akan memberikan sinyal kendali keluaran $m(t) = 0$, untuk sinyal kesalahan $e(t)$ yang konstan sehingga kontroler diferensial tidak mempengaruhi keadaan mantap. Kontroler diferensial digunakan untuk memperbaiki atau mempercepat respon transien sebuah sistem serta dapat meredam osilasi.

Berdasarkan karakteristik kontroler tersebut, kontroler diferensial umumnya dipakai untuk mempercepat respon awal suatu sistem, tetapi tidak memperkecil kesalahan pada keadaan tunaknya. Kerja kontroler diferensial hanyalah efek dari lingkup yang sempit, yaitu pada periode peralihan. Oleh sebab itu kontroler differensial tidak bisa digunakan tanpa ada kontroler lain.

2.3.4 Kontroler Proporsional Integral (PI)

Aksi kontrolnya dinyatakan dalam persamaan:

$$m(t) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e(t) dt \quad (2-5)$$

kontroler ini menghasilkan sinyal kesalahan $\int e(t) dt$ kemudian ditambahkan dengan sinyal kesalahan $e(t)$.

2.3.5 Kontroler Proportional Differensial (PD)

Aksi kontrolnya dinyatakan dalam persamaan:

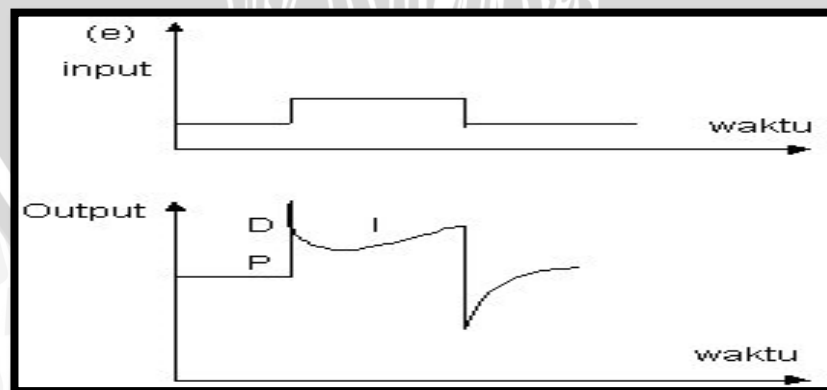
$$m \ddot{t} = K_p \cdot e \dot{t} + K_p \cdot T_d \frac{de \dot{t}}{dt} \quad (2-6)$$

Kontroler PD selalu mengukur kemiringan (*slope*) sinyal kesalahan $\frac{de(t)}{dt}$ dan memperkirakan akan besar *overshoot* yang akan terjadi serta memberikan koreksi sebelum terjadi lewatannya sehingga diperoleh *maximum overshoot* yang kecil.

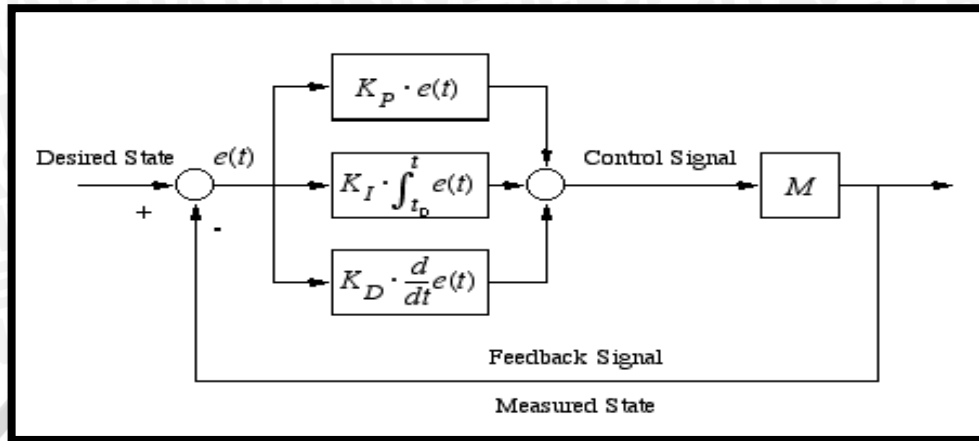
Jika kesalahan keadaan mantap tidak berubah terhadap waktu maka turunannya terhadap waktu sama dengan nol, sehingga kontroler PD tidak mempunyai pengaruh terhadap kesalahan keadaan mantap, tetapi jika terdapat perubahan kesalahan, kontroler PD akan mengurangi besar kesalahan keadaan mantap. Jadi kontrol PD digunakan untuk memperbaiki suatu sistem pengendalian yang tanggapan peralihannya mempunyai *maximum overshoot* yang berlebihan tanpa memperhitungkan kecepatan responnya.

2.3.6 Kontroler Proportional Integral Differensial (PID)

Setiap kekurangan dan kelebihan masing-masing kontroler P, I dan D dapat saling menutupi dengan menggabungkan ketiganya secara paralel menjadi kontroler proporsional integral differensial (PID). Elemen-elemen P, I, dan D masing-masing secara keseluruhan bertujuan untuk mempercepat reaksi sebuah sistem, menghilangkan *offset* dan menghasilkan perubahan awal yang besar (Gunterus, 1994, 8-10). Kontroler PID memiliki diagram kendali seperti yang ditunjukkan dalam Gambar 2.6.



Gambar 2.6 Hubungan fungsi waktu antara sinyal keluaran dan masukan kontroler PID



Gambar 2.7 Diagram Blok Kontroler PID

Sumber: Ogata, K., 1997

Aksi kontrolnya dapat dinyatakan sebagai berikut:

$$m \ddot{t} = K_p e \dot{t} + \frac{K_p}{T_i} \int_0^t e \dot{t} dt + K_p T_d \frac{de(t)}{dt} \quad (2-7)$$

Jenis kontroler ini digunakan untuk memperbaiki kecepatan respon dan mencegah terjadinya kesalahan keadaan mantap serta mempertahankan kestabilan.

Keluaran kontroler PID merupakan penjumlahan dari keluaran kontroler proporsional, integral dan differensial. Gambar 2.7 menunjukkan hubungan tersebut. Karakteristik kontroler PID sangat dipengaruhi oleh kontribusi besar dari ketiga parameter P, I, dan D. Penyetelan konstanta K_p , K_i , dan K_d akan mengakibatkan penonjolan sifat dari masing-masing elemen. Satu atau dua dari ketiga konstanta tersebut dapat disetel lebih menonjol dibandingkan yang lain. Konstanta yang menonjol itulah yang akan memberikan kontribusi pengaruh pada respon sistem secara keseluruhan (Gunterus, 1994, 8-10).

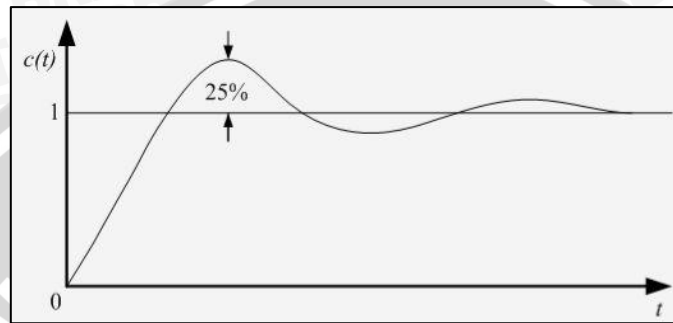
2.4 Metode Tuning PID

Metode Ziegler-Nichols.

Ziegler dan Nichols mengemukakan aturan-aturan untuk menentukan nilai dari gain proporsional K_p , waktu integral T_i , dan waktu derivatif T_d berdasarkan karakteristik respon transien dari *plant* yang diberikan. Penentuan parameter kontroler PID atau

penalaan kontroler PID tersebut dapat dilakukan dengan bereksperimen dengan plan. (Ogata, K., 1997)

Terdapat dua metode yang disebut dengan aturan penalaan Ziegler-Nichols, pada kedua metode tersebut memiliki tujuan yang sama yaitu untuk mencapai 25% *maximum overshoot* pada respon unit step, seperti ditunjukkan dalam Gambar 2.8.

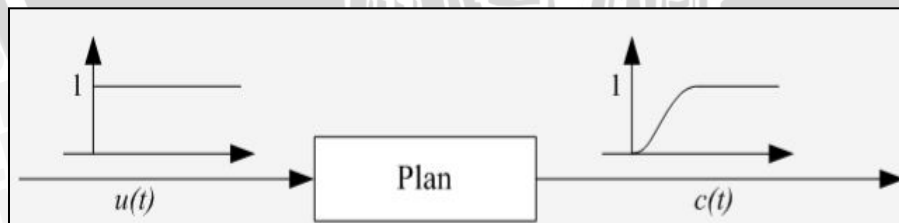


Gambar 2.8 Kurva Respon Unit Step yang Menunjukkan 25% Maximum Overshoot

Sumber: Ogata, K., 1997

a) Metode Pertama

Metode pertama atau sering disebut metode kurva reaksi, respon dari *plan* dapat dapat diperoleh secara eksperimental dengan masukan berupa unit step, seperti yang ditunjukkan dalam Gambar 2.9.

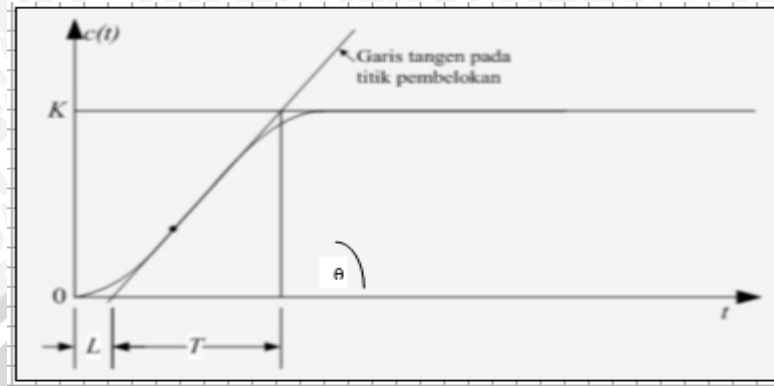


Gambar 2.9 Respons Plan Terhadap Masukan Berupa Unit Step

Sumber: Ogata, K., 1997

Jika dalam plan tersebut terdapat integrator atau *dominan complex-conjugate poles*, maka kurva respons unit *step* berbentuk seperti huruf S, seperti dalam Gambar 2.10.

Jika respons tidak memberikan bentuk kurva S, maka metode ini tidak berlaku. (Ogata, K., 1997).



Gambar 2.10 Respons Plan berbentuk S

Sumber: Ogata, K., 1997

Kurva berbentuk S tersebut dapat dikarakteristikan menjadi dua konstanta yaitu waktu tunda L dan konstanta waktu T . Waktu tunda dan konstanta waktu ditentukan dengan menggambar sebuah garis tangen pada titik pembelokan dari kurva S, dan menentukan perpotongan antara garis tangen dengan sumbu waktu t dan sumbu $c(t) = K$, Fungsi alih $C(s)/U(s)$ dapat dilakukan pendekatan dengan sistem orde satu dengan persamaan sebagai berikut:

$$\frac{C(s)}{U(s)} = \frac{Ke^{-Ls}}{Ts + 1} \quad (2-8)$$

Ziegler dan Nichols menyarankan untuk menentukan nilai-nilai dari K_p , T_i dan T_d berdasarkan pada formula yang ditunjukkan dalam Tabel 2.1 (Ogata, K., 1997)

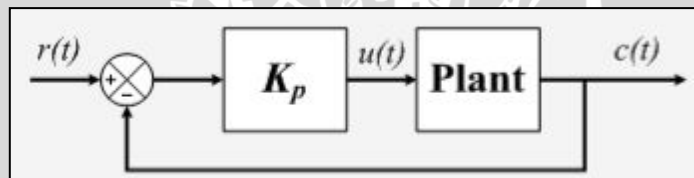
Tabel 2.1 Aturan Penalaran Ziegler-Nichols Berdasarkan Respons Unit Step Dari Plan

Tipe Kontroler	K_p	T_i	T_d
P	$\frac{T}{L}$	∞	0
PI	$0,9\frac{T}{L}$	$\frac{L}{0,3}$	0
PID	$1,2\frac{T}{L}$	$2L$	$0,5 L$

Sumber: Ogata, K., 1997

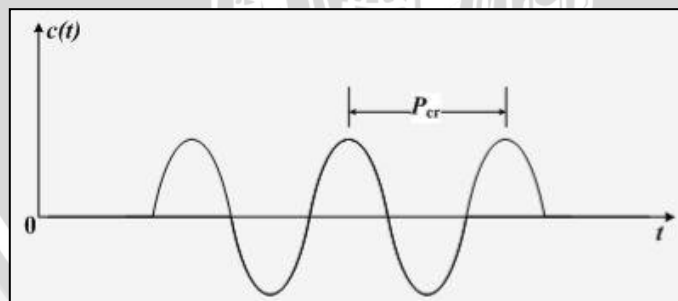
b) Metode Kedua

Dalam metode kedua *Ziegler-Nichols*, mula-mula yang dilakukan adalah membuat $T_i = 0$ dan $T_d = 0$. Kemudian hanya dengan menggunakan tindakan kontrol proporsional, harga ditingkatkan dari nol ke suatu nilai kritis K_{cr} disini mula-mula keluaran memiliki osilasi yang berkesinambungan (Jika keluaran tidak memiliki osilasi berkesinambungan untuk nilai K_p manapun yang telah diambil, maka metode ini tidak berlaku). Dari keluaran yang berosilasi secara berkesinambungan, penguatan kritis K_{cr} dan periode P_{cr} dapat ditentukan. Diagram blok sistem loop tertutup dengan kontroler proporsional dapat dilihat dalam Gambar 2.11 dan untuk osilasi berkesinambungan dengan periode P_{cr} dapat dilihat dalam Gambar 2.12 Ziegler dan Nichols menyarankan penyetelan nilai parameter K_p, T_i, T_d dan berdasarkan rumus yang diperlihatkan dalam Tabel 2.2. (Ogata, K., 1997)



Gambar 2.11 Sistem Loop Tertutup dengan Kontroler Proporsional

Sumber: Ogata, K., 1997



Gambar 2.12 Osilasi Berkesinambungan dengan periode P_{cr}

Sumber: Ogata, K., 1997

Tabel 2.2 Aturan Dasar Ziegler-Nichols Berdasarkan *Critical Gain* K_{cr} dan *Critical Period* P_{cr}

Tipe Kontroler	K_p	T_i	T_d
P	$0.5 K_{cr}$	∞	0
PI	$0.45 K_{cr}$	$\frac{1}{1,2} P_{cr}$	0
PID	$0.60 K_{cr}$	$0.5 P_{cr}$	$0.125 P_{cr}$

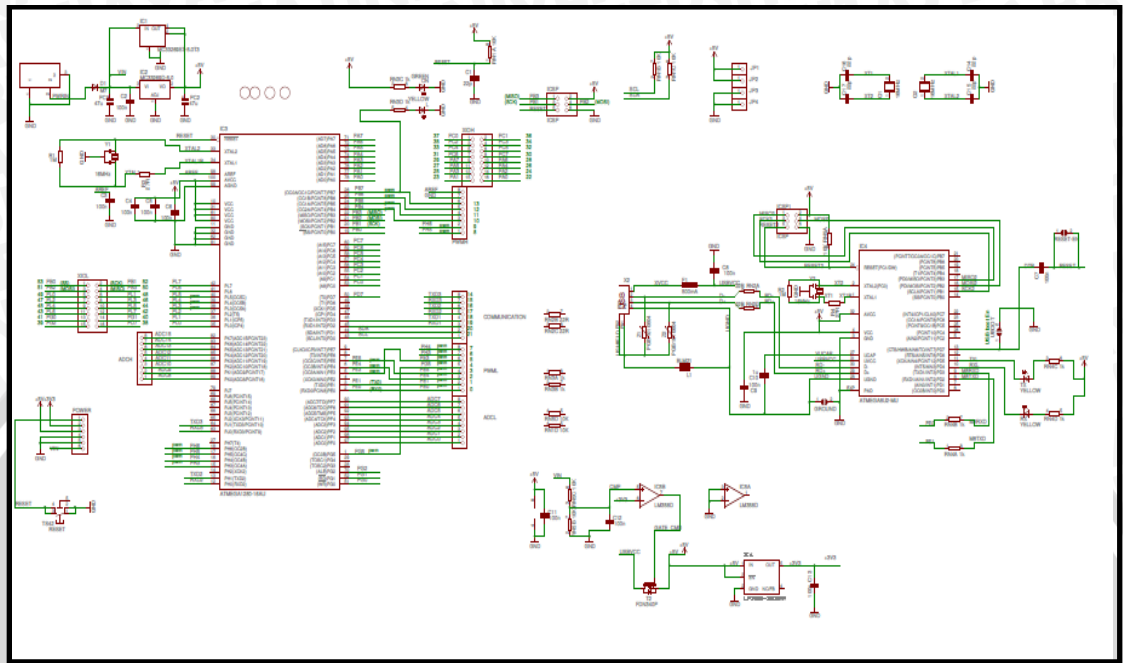
Sumber: Ogata, K., 1997

2.5 Mikrokontroler Arduino mega 2560

Arduino Mega2560 adalah *board* mikrokontroler berbasis ATmega2560. Arduino Mega memiliki 54 pin Input/Output digital (dimana 15 pin diantaranya dapat digunakan sebagai output PWM), 16 input analog, 4 UARTs (port serial perangkat keras), 16 MHz osilator Kristal, koneksi USB, *power jack*, ICSP *header*, dan tombol reset. Untuk mendukung mikrokontroler agar dapat digunakan, cukup hanya menghubungkan Arduino Mega2560 ke komputer dengan menggunakan kabel USB atau dengan memberi daya dengan adaptor AC-to-DC atau baterai agar dapat bekerja. Arduino Mega2560 juga cocok menggunakan sebagian besar *shield* yang didesain untuk Arduino Duemilanove.

Arduino Mega2560 berbeda dengan *board* Arduino sebelumnya dimana *board* sebelumnya menggunakan FTDI USB-to-serial *driver chip*. Pada Arduino Mega2560 terdapat ATmega16U2 (sebagai ganti dari ATmega8U2 pada board sebelumnya) yang deprogram sebagai USB-to-serial *converter*. Arduino Mega2560 juga memiliki RESET sirkuit yang lebih kuat dibandingkan *board* sebelumnya. Fitur baru yang terdapat pada Arduino Mega2560 yaitu penambahan pin SDA dan SCL yang berdekatan dengan pin AREF dan dua pin baru lainnya diletakkan dekat pin RESET, pin pertama yaitu IOREF yang memperbolehkan *shield* untuk beradaptasi dengan tegangan yang disediakan oleh *board*. Kedepannya, *shield* akan bekerja sangat baik dengan *board* yang menggunakan AVR yang dapat beroperasi dengan 5V dan dengan 3,3V (seperti pada Arduino Duemilanove). Pin kedua adalah pin yang tidak terhubung yang disediakan untuk tujuan

yang akan datang. Gambar 2.13 menunjukkan konfigurasi kaki I/O dari Arduino Mega 2560.



Gambar 2.13 Desain sistem Arduino Mega2560-R3(arduino.cc)

2.5.1 Daya

Arduino Mega2560 dapat diaktifkan melalui koneksi USB atau dengan menggunakan catu daya eksternal (otomatis). Daya eksternal (non-USB) dapat berasal dari adaptor AC-to-DC atau baterai. Adaptor dapat dihubungkan dengan menancapkan plug jack dengan pusat-positif sebesar 2,1mm pada *Power jack* pada *board*. Untuk baterai, ujung kepala baterai dapat dimasukkan pada pin Gnd dan Vin pada header di bagian konektor POWER. Board Arduino Mega2560 ini dapat beroperasi dengan catu daya eksternal antara 6 sampai dengan 20 volt. Jika diberi daya kurang dari 7V, kemungkinan pin 5V tetap dapat beroperasi tetapi tidak stabil. Jika menggunakan daya lebih dari 12V, maka regulator tegangan akan panas dan dapat merusak *board* Arduino Mega2560. Range daya yang direkomendasikan antara 7 sampai dengan 20 volt.

Pin POWER pada *board* Arduino Mega2560 diantaranya adalah:

- **VIN.** Tegangan masukan untuk *board* Arduino ketika menggunakan catu daya eksternal (berbeda dengan 5V yang berasal dari konektor USB atau sumber tegangan yang telah disesuaikan).
- **5V.** Pin output ini mengeluarkan output sebesar 5V yang telah disesuaikan menggunakan regulator yang berasal dari *board* Arduino. *Board* Arduino dapat dicatu dengan daya yang berasal dari *power jack* DC (7-12V), konektor USB (5V), atau pin VIN yang terdapat pada *board* (7-12V). Mencatu daya pada pin 5V dan 3,3V akan merusak regulator dan *board* Arduino.
- **3,3V.** Merupakan catu daya sebesar 3,3V yang dihasilkan oleh regulator pada *board* Arduino.
- **GND.** Merupakan pin *ground*.
- **IREF.** Pada *board Arduino*, pin ini menyediakan tegangan referensi yang dioperasikan oleh mikrokontroler. *Shield* yang telah dikonfigurasi dengan baik dapat membaca tegangan pin IREF dan dapat memilih catu daya yang sesuai atau dapat mengaktifkan tegangan translasi pada output yang bekerja pada 5V atau 3,3V.

2.5.2 Memori

ATmega2560 memiliki *flash memory* sebesar 256KB untuk penyimpanan kode (dimana 8KB digunakan sebagai bootloader), 8KB untuk SRAM dan 4KB untuk EEPROM (yang dapat dibaca dan ditulis dengan *library* EEPROM).

2.5.3 Input dan Output

Setiap pin dari 54 pin digital pada Arduino Mega2560 dapat digunakan sebagai input dan output menggunakan fungsi *pinMode()*, *digitalWrite()*, dan *digitalRead()*. Setiap pin beroperasi pada tegangan 5V. Masing-masing pin dapat menyediakan atau menerima arus maksimal sebesar 40 mA dan memiliki resistor *pull-up* internal dengan 20-50k Ω . Selain itu, beberapa pin memiliki fungsi special seperti:

- **Serial 0: 0 (RX) dan 1 (TX); Serial 1: 19 (RX) dan 18 (TX); Serial 2: 17 (RX) dan 16 (TX); Serial 3: 15 (RX) dan 14 (TX).** Pin (RX) digunakan untuk menerima

dan pin (TX) untuk mentransmisikan data TTL serial. Pin 0 dan 1 juga terhubung dengan pin koresponden dari ATmega16U2 USB-to-TTL serial chip.

- **Eksternal Interrupt: 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), dan 21 (interrupt 2).** Pin tersebut dapat dikonfigurasi untuk memicu interrupt pada kondisi *LOW*, tepi naik atau turun, dan pada kondisi *HIGH*. Untuk lebih jelasnya dapat dilihat pada fungsi *attachInterrupt()*.

- **PWM: 2-13 dan 44-46.** Menyediakan *output* PWM 8-bit dengan fungsi *analogWrite()*.

- **SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS).** Pin tersebut mendukung komunikasi SPI dengan menggunakan *library* SPI. Pin SPI juga terdapat pada ICSP header yang secara fisik dapat bekerja dengan baik untuk Uno, Duemilanove, Diecimila.

- **LED: 13.** Merupakan LED yang terpasang pada board dan terhubung dengan pin *digital* 13. LED akan menyala saat nilai *HIGH* dan akan mati ketika pin bernilai *LOW*.

- **TWI: 20 (SDA) dan 21 (SCL).** Mendukung komunikasi TWI menggunakan *Wire library*. Pin tersebut tidak terletak pada posisi yang sama dengan pin TWI pada Duemilanove atau Diecimila.

Arduino Mega2560 memiliki 16 analog input yang masing-masing menyediakan resolusi 10bit (1024 yang memiliki nilai berbeda). Pada dasarnya nilai yang terukur dari *ground* hingga 5V, yang titik tertingginya dapat diubah menggunakan pin AREF dan fungsi *analogReference()*.

Berikut adalah beberapa pin *analog* yang terdapat pada board:

- **AREF.** Merupakan tegangan referensi untuk *input analog*. digunakan dengan *analogReference()*.

- **Reset.** Membawa garis *LOW* untuk me-reset mikrokontroler. Pada umumnya digunakan untuk menambahkan tombol reset untuk membatasi tombol reset yang berada pada *board*.

2.5.4 Komunikasi

Arduino Mega2560 memiliki beberapa fasilitas untuk berkomunikasi dengan komputer, Arduino yang lain, atau mikrokontroler lain. ATmega2560 menyediakan empat perangkat UART untuk komunikasi *serial* TTL (5V). ATmega16U2 pada *board* merupakan saluran untuk USB dan menyediakan *port* com virtual untuk *software* pada komputer. Pada Windows diperlukan sebuah file dengan tipe .inf; tetapi tidak pada OSX atau Linux, kedua *Operating System* ini akan mengidentifikasi atau mengenali secara otomatis *board* Arduino sebagai *port* COM. *Software* Arduino menyediakan serial monitor yang memperbolehkan data tekstual untuk dikirimkan baik ke *board* ataupun dari *board*. LED RX dan TX pada *board* akan berkedip saat data dikirimkan melalui chip ATmega8U2/ ATmega16U2 dan koneksi USB ke konektor (namun tidak untuk komunikasi serial pada pin 0 dan 1).

Software Serial library memperbolehkan komunikasi serial antara beberapa pin *digital* pada Arduino Mega2560. ATmega2560 juga mendukung komunikasi TWI dan SPI. *Software* Arduino menyediakan *Wire library* untuk memudahkan bus TWI dan komunikasi SPI.

2.6 Sensor Air Flow

Pengukuran aliran mulai dikenal sejak tahun 1732 ketika Henry Pitot mengatur jumlah fluida yang mengalir. Dalam pengukuran fluida perlu ditentukan besaran dan vektor kecepatan aliran pada suatu titik dalam fluida dan bagaimana fluida tersebut berubah dari titik ke titik.

Jenis pengukur aliran yang paling luas digunakan adalah pengukuran tekanan diferensial. Pada prinsipnya beda luas penampang melintang dari aliran dikurangi dengan yang mengakibatkan naiknya kecepatan, sehingga menaikkan pula energi gerakan atau energi kinetis. Karena energi tidak bisa diciptakan atau dihilangkan (Hukum perpindahan energi), maka kenaikan energi kinetis ini diperoleh dari energi tekanan yang berubah. Lebih jelasnya, apabila fluida bergerak melewati penghantar (pipa) yang seragam dengan kecepatan rendah, maka gerakan partikel masing-masing umumnya sejajar disepanjang garis dinding pipa. Kalau laju aliran meningkat, titik puncak dicapai apabila gerakan partikel menjadi lebih acak dan kompleks. Kecepatan kira-kira di mana perubahan ini

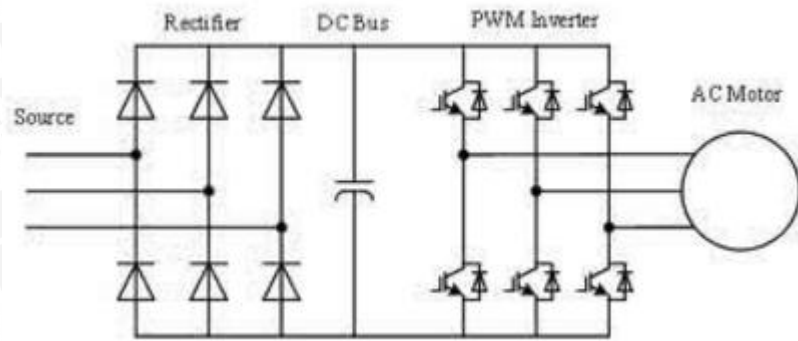
terjadi dinamakan kecepatan kritis dan aliran pada tingkat kelajuan yang lebih tinggi dinamakan turbulen dan pada tingkat kelajuan lebih rendah dinamakan laminer.

Pengukuran aliran metoda ini dapat dilakukan dengan banyak cara misalnya: menggunakan pipa *venturi*, pipa pitot, *orifice* plat (lubang sempit), *turbine flow meter*, rotameter, cara thermal, menggunakan bahan radio aktif, elektromagnetik, *ultra sonic* dan *flowmeter gyro*. Cara lain dapat dikembangkan sendiri sesuai dengan kebutuhan proses.

2.7 Variable Frequency Drive

Inverter adalah suatu peralatan elektronika daya yang berfungsi untuk mengubah listrik DC menjadi AC. *Inverter* seringkali disebut sebagai *Variable Frequency Drive* (VFD).

Prinsip kerja inverter adalah mengubah input motor (listrik AC) menjadi DC dan kemudian dijadikan AC lagi dengan frekuensi yang dikehendaki sehingga motor dapat dikontrol sesuai dengan kecepatan yang diinginkan. Untuk mengubah tegangan AC menjadi DC dibutuhkan penyearah (*converter* AC-DC) dan biasanya menggunakan penyearah tidak terkendali (*rectifier dioda*) namun juga ada yang menggunakan penyearah terkendali (*thyristor rectifier*). Setelah tegangan sudah diubah menjadi DC maka diperlukan perbaikan kualitas tegangan DC dengan menggunakan tandon kapasitor sebagai perata tegangan. Kemudian tegangan DC diubah menjadi tegangan AC kembali oleh inverter dengan teknik PWM (*Pulse Width Modulation*). Dengan teknik PWM ini bisa didapatkan amplitudo dan frekuensi keluaran yang diinginkan. Selain itu teknik PWM juga menghasilkan harmonisa yang jauh lebih kecil dari pada teknik yang lain serta menghasilkan gelombang sinusoidal, dimana kita tahu kalau harmonisa ini akan menimbulkan rugi-rugi pada motor yaitu cepat panas. Maka dari itu teknik PWM inilah yang biasanya dipakai dalam mengubah tegangan DC menjadi AC (*Inverter*). Skema rangkaian VFD ditunjukkan dalam gambar 2.14.



Gambar 2.14 Skema rangkaian *Variable Frequency Drive*

