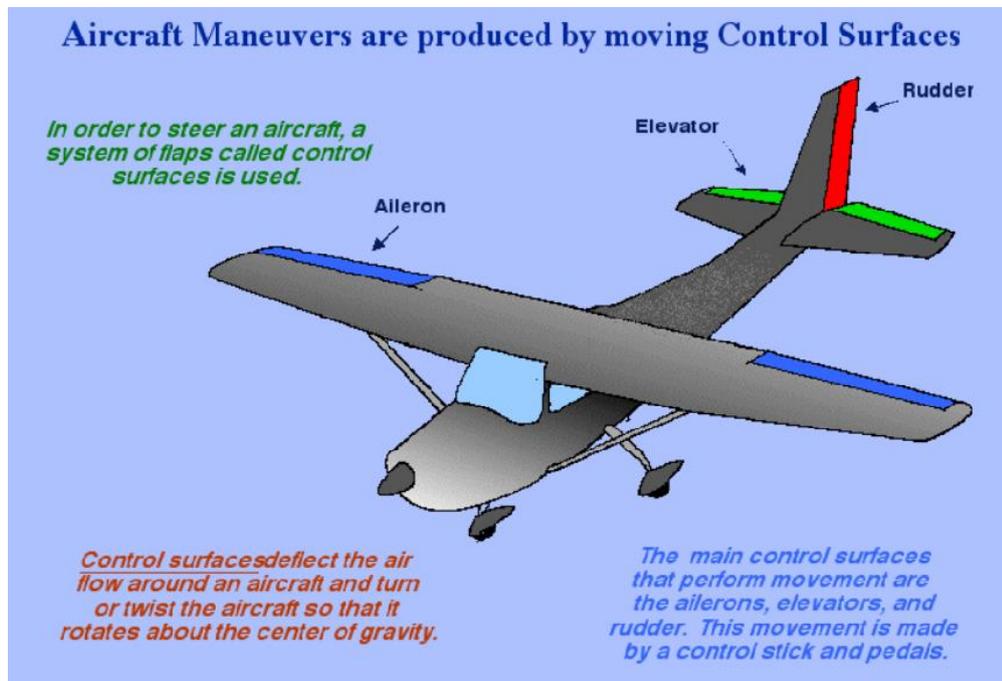


## BAB II

### TINJAUAN PUSTAKA

#### 1.1. *Flight Control System*

*Flight Control System* adalah gabungan dari sistem mekanik dan elektronik yang memungkinkan pesawat untuk bergerak secara baik diudara. *Flight control* dikendalikan oleh pilot dan co-pilot di *cockpit*. *Control system* terdiri dari *control cockpit*, *sensor*, *actuators* (*hydraulic*, *mechanical* atau *electrical*), dan *computer*. Gerakan yang di lakukan *Flight Control* adalah *rolling*, *pitching* dan *yawing*.



**Gambar 2.1** *Flight Control System* (SRMUNIV, 2016)

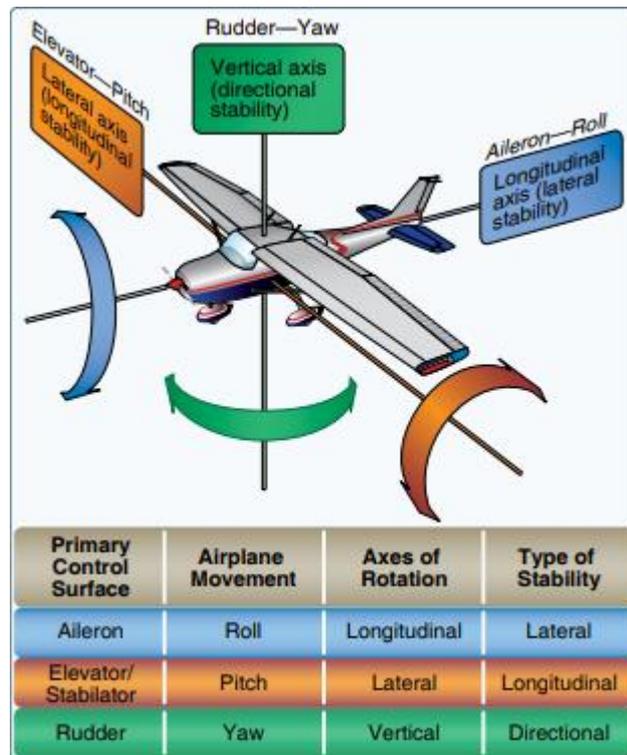
*Aircraft flight control system (AFCS)* erat sekali hubungannya dengan *flight control surface (FCS)* atau bidang kendali pesawat terbang, dimana *Flight Control Surface* merespon setiap pengaturan atau pergerakan yang dilakukan oleh pilot di dalam *cockpit* melalui suatu sistem yang saling berhubungan yang kemudian menggerakkan sistem mekanik untuk melakukan pergerakan pada pesawat yaitu *yawing*, *rolling*, dan *pitching* seperti yang terdapat pada Gambar 2.1.

Jadi secara singkatnya, *AFCS* merupakan suatu sistem yang mengendalikan sikap terbang suatu pesawat dengan menggerakkan *FCS* sebagai bidang kendalinya.

*Flight Control System* terbagi menjadi 2 bagian, yaitu *Primary Control Surface* dan *Secondary Control Surface*.

### 1.2. Primary Flight Control

*Primary control* merupakan kumpulan dari tiga komponen utama yang berperan menggerakkan pesawat ketika pesawat dalam keadaan terbang. Pada Gambar 2.2 dapat dilihat bahwa ketiga komponen ini bekerja pada masing masing sumbunya sendiri atau garis khayal yang membentang lurus (*axis*). Yaitu, *rudder* pada sumbu vertical (*vertical axis*), *elevator* pada sumbu lateral (*lateral axis*), dan *aileron* pada sumbu longitudinal (*longitudinal axis*).



**Gambar 2.2** Primary Flight Control (FAA, 2016)

Ada 3 hal yang bisa dilakukan oleh *primary control surface* diantaranya adalah:

1. Mengendalikan pergerakan pesawat.
2. Mengendalikan pesawat berdasarkan sumbu rotasinya.
3. Mengendalikan kestabilan pesawat.

Bidang kendali utama pada pesawat terbang adalah:

- a. *Aileron*, merupakan bidang kendali yang terletak pada *wing/sayap*.
- b. *Elevator*, merupakan bidang kendali yang terletak pada *horizontal stabilizer*.
- c. *Rudder*, merupakan bidang kendali yang terletak pada *vertical stabilizer*.

### 1.3. Secondary Flight Control

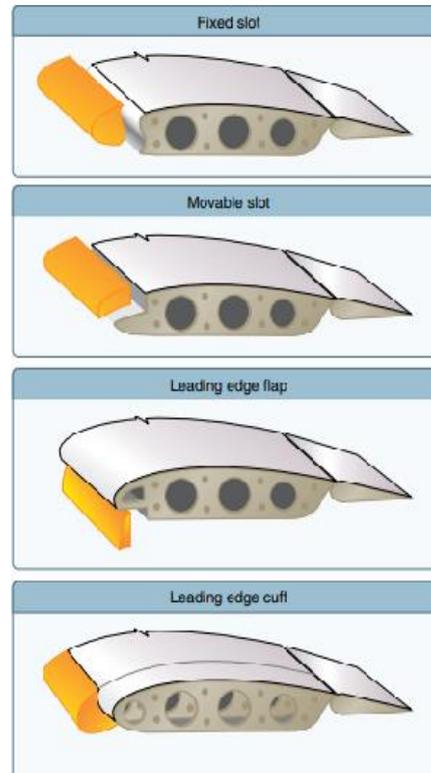
*Secondary flight control surface*, bisa dibilang sebagai bidang kendali tambahan yang bertujuan untuk membantu kinerja dari *primary control surface* dan pergerakan pesawat ketika terbang, *take off* atau pun *landing*.

Yang termasuk dalam *Secondary Flight Control Surface*, yaitu:

#### A. *Slat*

*Slat* adalah permukaan *aerodynamic* yang terletak pada *leading edge* di *wing*. Fungsi *slat* adalah menaikkan nilai *angle of attack* maksimum. Pada saat *approach*, *Angle of Attack* (AoA) pesawat nilainya sangat besar mendekati *AoA maks/AoA stall*. Maka *slat* ini digunakan supaya *Angle of Attack stall* nya meningkat, sehingga dengan *AoA approach* yg cukup tinggi itu tidak terlalu dekat dengan *AoA maks/AoA stall*. *Angle of Attack* sudut yang di bentuk dari *chord line* dan *relative wind*. Koefisien *lift* yang lebih tinggi dihasilkan dari *angle of attack* dan kecepatan, jadi dengan menyebarnya *slat* pada pesawat dapat terbang pada kecepatan yang lebih rendah, atau *take off* dan *landing* dengan jarak yang pendek. Pada penerbangan yang normal *slat* digunakan untuk mengurangi *drag*.

Beberapa jenis *slat* dapat dilihat pada Gambar 2.3 yang diantaranya adalah *fixed slat*, *movable slat*, *leading edge flap*, dan *leading edge cuff*.



**Gambar 2.3** Jenis-jenis *Slat* (FAA, 2016)

### B. *Flap*

*Flap* pada normalnya terletak di *trailing edge* pada bagian bawah dari *wing* untuk mengurangi kecepatan pada pesawat udara sehingga dapat aman diterbangkan dan untuk menambah sudut pada saat mendarat tanpa meningkatkan kecepatan. *Flap* mempercepat jarak *take off* dan *landing* dengan baik. Pada setiap penambahan defleksi *flap* akan menghasilkan peningkatan koefisien *lift* ( $C_L$ ) dan peningkatan koefisien *drag* ( $C_D$ ). Kondisi ini dimanfaatkan untuk keperluan kinerja pesawat pada saat kecepatan rendah, khususnya pada saat proses *take off* dan *landing*.

#### a. Proses *Landing* (Mendarat).

Dalam proses pendaratan ada beberapa faktor yang diperlukan, yaitu:

- 1) Kecepatan *Approach* Yang Rendah. Pesawat yang sedang dalam proses pendaratan terutama di daerah final *approach*,

mebutuhkan kecepatan *approach* ( $V_{app}$ ) yang rendah, agar pesawat hanya membutuhkan landasan luncur yang relatif pendek. Pada dasarnya kecepatan approach adalah :

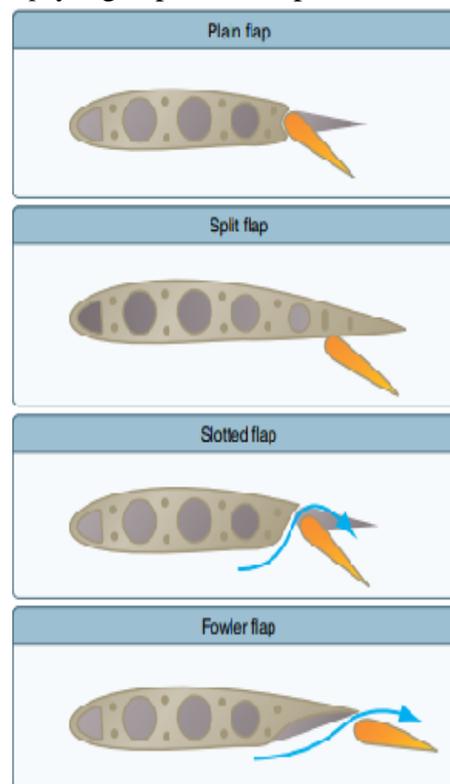
$$V_{app} = 1.2 V_{stall}$$

Agar  $V_{app}$  rendah, maka  $V_{stall}$  dibuat serendah mungkin dengan cara meningkatkan  $C_{Lmax}$  secara maksimal. Dengan demikian guna mendapatkan  $V_{app}$  serendah mungkin, maka diperlukan defleksi *flap* yang maksimal ( $60^\circ$ ).

- 2) Efek Pengereman Yang Tinggi. Selain  $V_{app}$  yang rendah, pada saat proses pendaratan dibutuhkan efek pengereman yang tinggi agar membantu pesawat dapat meluncur dengan menggunakan landasan yang relatif pendek. Dengan demikian dibutuhkan *drag* yang besar, yang diperoleh melalui defleksi *flap* yang maksimal ( $60^\circ$ ).
- 3) Visibilitas Penerbang. Pada proses pendaratan, posisi pesawat harus memberikan jaminan visibilitas yang baik bagi penerbang. Saat pesawat pada posisi *descend* atau *gliding* (posisi pesawat terbang menurun) untuk melakukan *landing approach*, sumbu longitudinal pesawat membentuk sudut negatif terhadap horison. Posisi ini akan mengurangi sudut serang, sehingga pesawat terbang dengan sudut serang relatif kecil. Pesawat yang terbang dengan sudut serang kecil dan kecepatan yang rendah karena melakukan proses pendaratan, hanya menghasilkan *lift* yang rendah. Kondisi ini diatasi dengan defleksi *flap* maksimal ( $60^\circ$ ), sehingga diperoleh *lift* ( $C_L$  positif) meskipun pesawat terbang pada sudut serang kecil atau bahkan sudut serang negatif. Dengan demikian pesawat terbang dengan posisi *descend* atau *gliding* memberikan jaminan visibilitas penerbang yang baik, dan pesawat tetap menghasilkan gaya angkat yang cukup untuk mengimbangi berat pesawat.

b. Proses *Take off*

Proses *take off* dimulai dari *run up* di ujung landasan, kemudian pesawat mulai bergerak dengan memacu kecepatan. Setelah diperoleh kecepatan yang diinginkan, maka pesawat mulai *take off*. Kecepatan *take off* adalah kecepatan minimal pesawat untuk dapat terbang di udara, yang tentu saja harus lebih tinggi dari  $V_{stall}$ . Dengan demikian semakin rendah  $V_{stall}$ , maka kecepatan *take off* segera tercapai, sehingga pesawat dapat *take off* dengan menggunakan landasan yang lebih pendek. Penurunan  $V_{stall}$  dilakukan dengan cara defleksi *flap*, namun penggunaan *flap* harus disertai dengan peningkatan *drag* yang kecil. Dengan demikian penggunaan *flap* pada saat *take off*, menggunakan defleksi *flap* yang kecil antara  $15^\circ$  sampai dengan  $30^\circ$ , yang biasanya sudah diatur pada flap posisi *take off*. Pada umumnya ada empat jenis umum *flap*, yaitu *Plan Flap*, *Split Flap*, *Slotted Flap*, dan *Fowler Flap* yang dapat dilihat pada Gambar 2.4.



**Gambar 2.4** Jenis-jenis *Flap* (FAA, 2016)

*Plan Flap* adalah yang paling sederhana dari empat jenis. Jenis *flap* ini meningkatkan *airfoil* camber dan menghasilkan peningkatan signifikan pada *Coefisien Lift (CL)* pada *AOA* yang diberikan. Pada saat bersamaan, hal tersebut meningkatkan *drag* dan menggerakkan *Center of Pressure (CP)* pada bagian belakang *airfoil* dan menghasilkan gerakan *nose-down*.

*Split Flap* meningkatkan *lift* yang sedikit lebih besar dibandingkan *plan flap*. Lebih banyak *drag* terjadi dikarenakan produksi pola udara yang berputar dibelakang *airfoil*. Ketika *fully extended*, keduanya *plan* dan *split flaps* menghasilkan *drag* tinggi dengan sedikit tambahan *lift*. Salah satu jenis *flap* yang paling populer pada pesawat saat ini adalah *Slotted Flap*. *Slotted Flap* meningkatkan koefisien *lift* secara signifikan lebih dari *plan flap* atau *split flap*.

*Fowler Flap* adalah jenis *slotted flap*. Desain tutup ini tidak hanya mengubah *wing* camber, namun itu juga meningkatkan area *wing* itu sendiri. Pada awal *extension*, *fowler flap* meningkatkan *drag* yang sangat sedikit, tetapi meningkatkan *lift* yang baik sekaligus meningkatkan area dan camber. Pilot harus mengetahui bahwa ketika *flap extend* dapat menyebabkan *nose up* ataupun *nose down*, tergantung pada jenis pesawat. Ketika *fowler flap fully extension*, *flap* mengarah ke bawah hal ini menyebabkan meningkatnya *drag* dengan sedikit tambahan peningkatan pada *lift*.

### C. Spoiler

*Spoiler* adalah *high drag devices* yang terbuka dari *wing* untuk merusak *airflow*, mengurangi *lift* dan menambah *drag*. Pada *gliders*, *spoiler* sangat sering digunakan untuk mengontrol kecepatan menurun atau untuk ketepatan mendarat. Pada pesawat lain, *spoiler* digunakan untuk mengontrol gerakan *rolling* dan mengurangi *yawing* yang kurang baik.

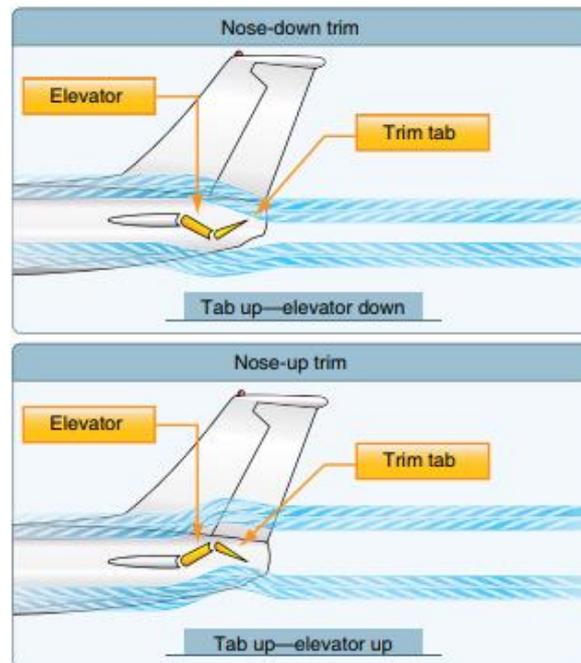


**Gambar 2.5 Spoiler** (FAA, 2016)

Untuk memutar ke kanan, seperti yang terdapat pada Gambar 2.5, *spoiler* pada wing kanan diangkat, mengurangi gaya angkat dan menambah *drag* di kanan. *Wing* kanan turun, dan pesawat *banks* dan *yawing* ke kanan. Adanya *spoiler* pada kedua *wing* pada waktu yang sama membuat pesawat turun tanpa mendapatkan kecepatan. *Spoiler* juga membantu mengurangi *ground roll* setelah landing atau mendarat. Dengan peungurangan *lift* atau gaya angkat, *spoiler* mentransfer berat menuju *wheels*, memberikan pengereman yang efektif. *Spoiler* ada dua macam yaitu, *Ground Spoilers (Lift Dumper)* dan *Flight Spoiler*.

#### D. Trim Tab

Bagian *Flight Control Surface* (FCS) yang paling kecil umumnya pada pesawat adalah *trim tabs*. *Trim tabs* biasanya terdapat pada *horizontal stabilizer* dan melekat pada *elevator*. *Trim control* digunakan pilot untuk menyeimbangkan gaya angkat dan gaya hambat yang dihasilkan dari berbagai beban dan kecepatan udara dari atas permukaan sayap. Hal ini dapat mengurangi usaha yang dibutuhkan untuk menyesuaikan dan mempertahankan pergerakan pesawat yang diinginkan. Selain *trim tabs* ada juga *balance tab* dan *control tab*. Fungsinya tidak jauh dari *trim tab* yaitu membantu mengontrol pergerakan *primary control* agar tetap terarah. Prinsip kerja *tab* selalu berlawanan dengan *primary control* seperti yang dapat dilihat pada Gambar 2.6.



**Gambar 2.6** *Trim-tab* (FAA, 2016)

## 1.4. Arduino

### 1.4.1. Pengertian Arduino

Arduino merupakan sebuah *platform hardware open source* elektronik yang mudah digunakan baik dari sisi *hardware* maupun *software* (Arduino). Arduino merupakan salah satu perangkat berbasis mikrokontroler yang bersifat *opensource*, perangkat ini dibekali dengan Prosesor Atmel AVR dengan menggunakan bahasa pemrograman sendiri yang dikembangkan dari bahasa C dan di kemas dalam *libraries* arduino yang menjadikan bahasa pemrograman arduino ini cukup mudah untuk dipelajari (Tresna, 2016).

Arduino merupakan papan rangkaian elektronik *open source* yang dibekali sebuah *chip microcontroller* jenis AVR dari perusahaan Atmel. Mikrokontroler itu sendiri adalah *chip* atau *IC (integrated circuit)* yang bisa diprogram menggunakan komputer. Tujuan menanamkan program pada mikrokontroler adalah agar rangkaian elektronik dapat membaca *input*, memproses *input* tersebut dan kemudian menghasilkan *output* sesuai yang diinginkan. Jadi mikrokontroler bertugas sebagai ‘otak’ yang

mengendalikan *input*, proses dan *output* sebuah rangkaian elektronik. Gambar 2.7 merupakan salah satu jenis Arduino, yaitu Arduino UNO.



**Gambar 2.7** Arduino UNO (Yudha, 2016)

Kegunaan Arduino tergantung kepada kita yang membuat program. Arduino bisa digunakan untuk mengontrol *LED*, bisa juga digunakan untuk mengontrol helikopter. Contoh yang sudah pernah dibuat adalah *MP3 player*, pengontrol motor, mesin *CNC*, monitor kelembaban tanah, pengukur jarak, penggerak servo, balon udara, pengontrol suhu, monitor energi, stasiun cuaca, pembaca *RFID*, drum elektronik, *GPS logger*, *monitoring* bensin dan masih banyak lagi.

#### 1.4.2. Kelebihan Arduino

1. Tidak perlu perangkat *chip programmer* karena di dalamnya sudah ada *bootloader* yang akan menangani *upload program* dari komputer.
2. Sudah memiliki sarana komunikasi *USB*, sehingga pengguna Laptop yang tidak memiliki *port serial/RS323* bisa menggunakannya.
3. Bahasa pemrograman relatif mudah karena *software* Arduino dilengkapi dengan kumpulan *library* yang cukup lengkap.

4. Memiliki modul siap pakai (*shield*) yang bisa ditancapkan pada *board* Arduino. Misalnya *shield* GPS, Ethernet, SD Card, dll.

#### 1.4.3. Bahasa Pemrograman Arduino

Bahasa pemrograman Arduino adalah bahasa C. Tetapi bahasa ini sudah dipermudah menggunakan fungsi-fungsi yang sederhana sehingga pemula pun bisa mempelajarinya dengan cukup mudah. Untuk membuat program Arduino dan mengupload ke dalam *board* Arduino, anda membutuhkan *software* Arduino IDE (*Integrated Development Enviroment*).

#### 1.4.4. Deskripsi Arduino UNO

##### 1.4.4.1. Arduino UNO

Arduino Uno adalah papan sirkuit berbasis mikrokontroler ATmega328. Memiliki 14 *input/output digital* (6 *output* untuk PWM), 6 *analog input*, resonator kristal keramik 16 MHz, Koneksi USB, soket adaptor, *pin header ICSP*, dan tombol *reset*. Hal inilah yang dibutuhkan untuk mensupport mikrokontrol secara mudah terhubung dengan kabel *power USB* atau kabel *power supply* adaptor AC ke DC atau juga *battery*.

Uno berbeda dari semua *board* mikrokontrol diawal-awal yang tidak menggunakan *chip* khusus *driver FTDI USB-to-serial*. Sebagai penggantinya penerapan *USB-to-serial* adalah ATmega16U2 versi R2 (versi sebelumnya ATmega8U2). Versi Arduino Uno Rev.2 dilengkapi resistor ke 8U2 ke garis ground yang lebih mudah diberikan ke mode DFU.

##### 1.4.4.2. Input dan Output Arduino

Masing-masing dari 14 *pin* UNO dapat digunakan sebagai *input* atau *output*, menggunakan perintah fungsi *pinMode()*, *digitalWrite()*, dan *digitalRead()* yang menggunakan tegangan operasi 5V. Setiap *pin* dapat

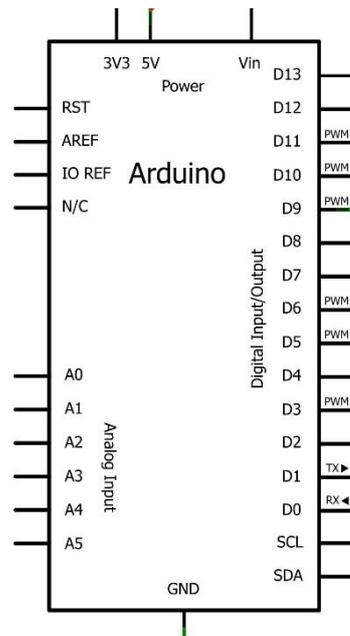
menerima arus maksimal hingga 40mA dan resistor *internal pull-up* antara 20-50kohm, beberapa *pin* memiliki fungsi kekhususan antara lain:

- *Serial*: 0 (RX) dan 1 (TX). Sebagai penerima (RX) dan pemancar (TX) TTL *serial data*. *Pin* ini terkoneksi untuk *pin* korespondensi *chip* ATmega8U2 USB-toTTL *Serial*.
- *External Interrupts*: 2 dan 3. *Pin* ini berfungsi sebagai konfigurasi *trigger* saat interupsi *value low*, naik, dan tepi, atau nilai *value* yang berubah-ubah.
- *PWM*: 3, 5, 6, 9, 10, dan 11. Melayani *output* 8-bit PWM dengan fungsi *analogWrite()*.
- *SPI*: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). *Pin* yang *support* komunikasi *SPI* menggunakan *SPI library*.
- *LED*: 13. Terdapat LED indikator bawaan (*built-in*) dihubungkan ke digital *pin* 13, ketika nilai *value HIGH led* akan *ON*, saat *value LOW led* akan *OFF*.
- Uno memiliki 6 *analog input* tertulis di label A0 hingga A5, masing-masingnya memberikan 10 *bit* resolusi (1024). Secara asal *input analog* tersebut terukur dari 0 (*ground*) sampai 5V, itupun memungkinkan perubahan teratas dari jarak yang digunakan oleh *pin* AREF dengan fungsi *analogReference()*.

Sebagai tambahan, beberapa *pin* ini juga memiliki kekhususan fungsi antara lain:

- *TWI*: *pin* A4 atau *pin* SDA dan and A5 atau *pin* SCL. *Support* *TWI communication* menggunakan *Wire library*. Inilah *pin* sepasang lainnya di *board* UNO:
- AREF. Tegangan referensi untuk *input analog*. digunakan fungsi *analogReference()*.
- *Reset*. Menekan jalur *LOW* untuk mereset mikrokontroler, terdapat tambahan tombol *reset* untuk melindungi salah satu blok.

#### 1.4.4.3. Bagian-bagian Arduino UNO



**Gambar 2.8** Bagian-bagian Arduino UNO (Andrew, 2018)

Gambar 2.8 menunjukkan bagian-bagian dari Arduino UNO dan berikut penjelasan mengenai bagian-bagian dari Arduino UNO:

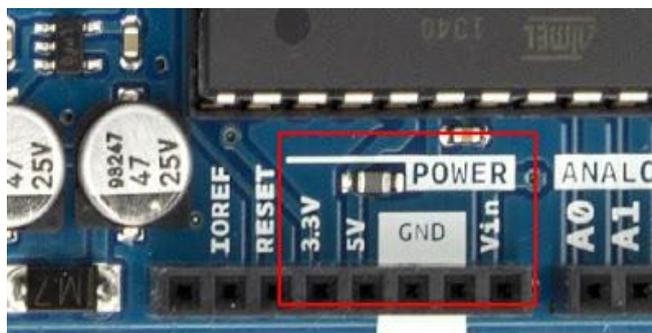
**Tabel 2.1** Penjelasan bagian-bagian Arduino UNO

No	Nama Bagian	Deskripsi
1	USB <i>Female Type-B</i>	Sebagai sumber DC 5V sekaligus untuk jalur pemrograman antara PC dan Arduino
2	<i>BarrelJack</i>	Sebagai <i>input</i> sumber antara 7-12 V
3	<i>PinGND</i>	Sebagai sumber pentanahan ( <i>ground</i> )
4	<i>Pin 5V</i>	Sebagai sumber tegangan 5V
5	<i>Pin 3.3 V</i>	Sebagai sumber tegangan 3.3V
6	A0–A5	Sebagai <i>AnalogInput</i>
7	2-13	Sebagai <i>I/O digital</i>
8	0-1	Sebagai <i>I/O digital</i> sekaligus bisa juga sebagai Rx Tx

No	Nama Bagian	Deskripsi
9	AREF	Sebagai <i>Analog</i> Referensi untuk fungsi ADC
10	Tombol <i>RESET</i>	Sebagai perintah <i>ResetArduino</i>
11	LED	Sebagai indikator Daya
12	LED Rx Tx	Sebagai indikator Rx Tx saat pengisian program
13	Mikrokontroler	Sebagai otak arduino dengan menggunakan mikrokontroler AVR Atmega328

#### 1.4.4.4. Power Arduino

Arduino Uno dapat disuplai langsung ke *USB* atau *power supply* tambahan yang pilihan *power* secara otomatis berfungsi tanpa saklar. Kabel *external (non-USB)* seperti menggunakan adaptor *AC* ke *DC* atau baterai dengan konektor *plug* ukuran 2,1mm polaritas positif di tengah ke *jack power* di *board*. Jika menggunakan baterai dapat disematkan pada *pin GND* dan *Vin* di bagian *Power connector*.



**Gambar 2.9** Pin Power Arduino (Jameco Electronics, 2018)

Papan Arduino pada Gambar 2.9 dapat disuplai tegangan kerja antara 6 - 20 V, jika catu daya di bawah tegan standar 5V *board* akan tidak stabil, jika dipaksakan ke tegangan regulator 12V mungkin *board* arduino cepat panas (*overheat*) dan merusak *board*. Sangat direkomendasikan tegangannya 7-12V.

Penjelasan *power PIN*:

- VIN – *Input voltase board* saat anda menggunakan sumber catu daya luar (adaptor USB 5 V atau adaptor yang lainnya 7-12V), Anda bisa menghubungkannya dengan *pin* Vin ini atau langsung ke *jack power* 5V. DC *power jack* (7-12V), Kabel konektor *USB* (5V) atau catu daya lainnya (7-12V). Menghubungkan secara langsung *power supply* luar (7-12V) ke *pin* 5V atau *pin* 3.3V dapat merusak rangkaian Arduino ini.
- 3V3 – *Pin* tegangan 3.3V catu daya umum langsung ke *board*. Maksimal arus yang diperbolehkan adalah 50 mA.
- GND – *Pin Ground*.
- IOREF – *Pin* ini penyedia referensi tegangan agar mikrokontrol beroperasi dengan baik. Memilih sumber daya yang tepat atau mengaktifkan penerjemah tegangan pada *output* untuk bekerja dengan 5V atau 3.3V.

#### 1.4.4.5. Komunikasi

Arduino Uno memiliki nomor untuk berkomunikasi dengan komputer, Arduino lainnya, ataupun dengan mikrokontroler. Pada ATmega328 menerjemahkan serial komunikasi UART TTL (5V) pada *pin* 0 (RX) dan 1 (TX). Pada ATmega16U2 serial komunikasinya dengan *USB* dan *port virtual* pada *software* di komputer. Perangkat lunak (*firmware*) 16U2 menggunakan *driver* standar *USB COM* dan tidak membutuhkan *driver* luar lainnya. Bagaimanapun pada *OS Windows file* ekstensi .inf sangat diperlukan. *Software* Arduino bawaan telah menyertakan serial monitor yang sangat mudah membaca dan mengirim data dari dan ke Arduino. *LED* indikator TX dan RX akan kedip ketika data telah terkirim via koneksi *USB-to-serial* dengan *USB* pada komputer (tetapi tidak pada serial *com* di *pin* 0 dan *pin* 1).

*SoftwareSerial library* membolehkan banyak *pin serial communication* pada UNO. ATmega328 juga *support I2C (TWI)* dan *SPI communication*. *Software Arduino* terbenam di dalamnya *Wire library* untuk memudahkan penggunaan *bus I2C*.

#### 1.4.4.6. Mikrokontroler ATmega 328

Mikrokontroler adalah sebuah komputer kecil (*“special purpose computers”*) di dalam satu IC yang berisi CPU, memori, *timer*, saluran komunikasi serial dan parallel, *Port input/output*, ADC. Mikrokontroler digunakan untuk suatu tugas dan menjalankan suatu program. Saat ini keluarga mikrokontroler yang ada di pasaran yaitu Intel 8048 dan 8051(MCS51), Motorola 68HC11, Microchip PIC, Hitachi H8, dan Atmel AVR.

ATmega328 adalah mikrokontroler keluaran Atmel yang merupakan anggota dari keluarga AVR 32-bit. Mikrokontroler ini memiliki kapasitas *flash (memory program)* sebesar 32 Kb (32.768 *bytes*), memori (*static RAM*) 2 Kb (2.048 *bytes*), dan EEPROM (*non-volatile memory*) sebesar 1024 *bytes*. Kecepatan *clock* yang dapat dicapai adalah 16 MHz.

ATmega328 adalah prosesor yang kaya fitur. Dalam chip yang dipaketkan dalam bentuk DIP-28 ini terdapat 20 pin Input/Output (21 pin bila pin reset tidak digunakan, 23 pin bila tidak menggunakan osilator eksternal), dengan 6 di antaranya dapat berfungsi sebagai pin ADC (*analog-to-digital converter*), dan 6 lainnya memiliki fungsi PWM (*pulse width modulation*). Berikut adalah bentuk fisik dari ATmega328 dapat dilihat pada Gambar 2.10 :



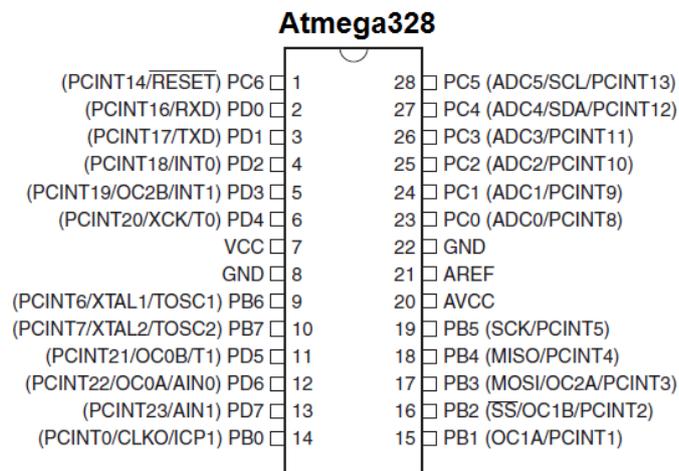
**Gambar 2.10** Bentuk Fisik ATMega328 (Microchip, 2016)

Mikrokontroler ini diproduksi oleh atmel dari seri AVR. Untuk seri AVR ini banyak jenisnya, yaitu Atmega 328, Atmega 8535, Mega 8515, Mega 16, dan lain-lain. Mikrokontroler ATmega328 digunakan pada Arduino UNO sebagai otak untuk mengendalikan perangkat elektronik yang akan dirancang. ATmega328 itu sendiri diproduksi oleh ATMEL yang mempunyai arsitektur *RISC (Reduced Instruction Set Computing)* dimana arsitektur RISC ini adalah suatu arsitektur yang memiliki instruksi yang sederhana namun memiliki banyak fasilitas tambahan.

Fitur-fitur yang terdapat pada mikrokontroler ATmega328 antara lain :

- Memiliki EEPROM (*Electrically Erasable Programmable Read Only Memory*) sebesar 1KB sebagai tempat penyimpanan data semi permanen karena EEPROM tetap dapat menyimpan data meskipun catu daya dimatikan.
- Memiliki SRAM (*Static Random Access Memory*) sebesar 2KB.
- Memiliki pin I/O digital sebanyak 14 pin 6 diantaranya PWM (*Pulse Width Modulation*) output.
- 32 x 8-bit register serba guna.
- Dengan clock 16 MHz kecepatan mencapai 16 MIPS.
- 32 KB *Flashmemory* dan pada arduino memiliki *bootloader* yang menggunakan 2 KB dari *flashmemory* sebagai *bootloader*.
- 130 macam instruksi yang hampir semuanya dieksekusi dalam satu siklus clock.

## 1.4.4.7. Konfigurasi ATmega 328



**Gambar 2.11** Pin ATmega 328 (Zain, 2018)

Penjelasan konfigurasi *pin* ATmega 328 berdasarkan Gambar 2.11 yaitu :

- VCC  
Merupakan *supply* tegangan digital.
- GND  
Merupakan ground untuk semua komponen yang membutuhkan *grounding*.
- Port B  
Merupakan jalur data 8 bit yang dapat difungsikan sebagai *input/output*. Selain itu PORT B juga dapat memiliki fungsi seperti di bawah ini :
  - ✓ ICP1 (PB0), berfungsi sebagai *Timer Counter 1 input capture pin*.
  - ✓ OC1A (PB1), OC1B (PB2) dan OC2 (PB3) dapat difungsikan sebagai keluaran PWM (*Pulse Width Modulation*).
  - ✓ MOSI (PB3), MISO (PB4), SCK (PB5), SS (PB2) merupakan jalur komunikasi SPI.
  - ✓ Selain itu pin ini juga berfungsi sebagai jalur pemrograman serial (ISP).

- ✓ TOSC1 (PB6) dan TOSC2 (PB7) dapat difungsikan sebagai sumber *clock external* untuk *timer*.
- ✓ XTAL1 (PB6) dan XTAL2 (PB7) merupakan sumber *clock* utama mikrokontroler.

Tabel 2.2 Port B

Port Pin	Alternate Function
PB7	XTAL2 ( <i>chip clock oscillator pin 1 or external clock input</i> ) TOSC2 ( <i>timer oscillator pin 1</i> ) PCINT7 ( <i>pin change interrupt 6</i> )
PB6	SCK ( <i>SPI bus master clock input</i> ) PCINT5 ( <i>pin change interrupt 5</i> )
PB5	MISO ( <i>SPI bus master input/slave output</i> ) PCINT4 ( <i>pin change interrupt 4</i> )
PB4	MOSI ( <i>SPI bus master output/slave input</i> ) OC2A ( <i>timer/counter 2, output compare match B output</i> ) PCINT3 ( <i>pin change interrupt 3</i> )
PB3	SS ( <i>SPI bus master slave select</i> ) OCIB ( <i>timer/counter 1, output compare match B output</i> ) PCINT2 ( <i>pin change interrupt 2</i> )
PB2	OCIA ( <i>timer/counter 1, output compare match A output</i> ) PCINT1 ( <i>pin change interrupt 1</i> )
PB1	ICPI ( <i>timer/counter 1 input capture input</i> ) CLK0 ( <i>divided system clock output</i> ) PCINT0 ( <i>pin change interrupt 0</i> )
PB0	XTAL2 ( <i>chip clock oscillator pin 1 or external clock input</i> ) TOSC2 ( <i>timer oscillator pin 1</i> ) PCINT7 ( <i>pin change interrupt 6</i> )

- Port C

Merupakan jalur data 7 bit yang dapat difungsikan sebagai *input/output* digital. Fungsi PORTC antara lain sebagai berikut :

- ✓ ADC6 *channel* (PC0,PC1,PC2,PC3,PC4,PC5) dengan resolusi sebesar 10 bit. ADC dapat kita gunakan untuk mengubah *input* yang berupa tegangan analog menjadi data digital.

- ✓ I<sup>2</sup>C (SDA dan SDL) merupakan salah satu fitur yang terdapat pada *PORTC*. I<sup>2</sup>C digunakan untuk komunikasi dengan sensor atau *device* lain yang memiliki komunikasi data tipe I<sup>2</sup>C seperti sensor kompas dan *accelerometer*.
- ✓ *RESET/PC6*  
Jika *RSTDISBL Fuse* diprogram, maka PC6 akan berfungsi sebagai *pin I/O*. *Pin* ini memiliki karakteristik yang berbeda dengan *pin-pin* yang terdapat pada *port C* lainnya. Namun jika *RSTDISBL Fuse* tidak diprogram, maka *pin* ini akan berfungsi sebagai *inputreset*. Dan jika level tegangan yang masuk ke *pin* ini rendah dan pulsa yang ada lebih pendek dari *pulsaminimum*, maka akan menghasilkan suatu kondisi *reset* meskipun *clock*-nya tidak bekerja.
- ✓ *AVcc*  
*Pin* ini berfungsi sebagai *supply* tegangan untuk ADC. Untuk *pin* ini harus dihubungkan secara terpisah dengan VCC karena *pin* ini digunakan untuk *analog* saja. Bahkan jika ADC pada AVR tidak digunakan tetap saja disarankan untuk menghubungkannya secara terpisah dengan VCC. Jika ADC digunakan, maka *AVcc* harus dihubungkan ke VCC melalui *LOWpassfilter*.
- ✓ *AREF* Merupakan *pin* referensi jika menggunakan ADC.

Tabel 2.2 Port C

Port Pin	Alternate Function
PC6	RESET ( <i>reset pin</i> ) PCINT14 ( <i>pin change interrupt 14</i> )
PC5	XTAL2 ( <i>ADC input channel 5</i> ) SCL ( <i>2-wire serial bus clock line</i> ) PCINT13 ( <i>pin change interrupt 13</i> )
PC4	ADC4 ( <i>ADC input channel 4</i> ) SDA ( <i>2-wire serial bus data input/output line</i> ) PCINT12 ( <i>pin change interrupt 12</i> )

Port Pin	Alternate Function
PC3	ADC3 (ADC input channel 3) PCINT11 (pin change interrupt 11)
PC2	ADC2 (ADC input channel 2) PCINT10 (pin change interrupt 10)
PC1	ADC1 (ADC input channel 1) PCINT9 (pin change interrupt 9)
PC0	ADC0 (ADC input channel 0) PCINT8 (pin change interrupt 8)

- Port D

Merupakan jalur data 8 bit yang masing-masing *pin*-nya juga dapat difungsikan sebagai *input/output*. Sama seperti Port B dan Port C, Port D juga memiliki fungsi dibawah ini :

- ✓ USART (TXD dan RXD) merupakan jalur data komunikasi serial dengan level sinyal TTL. Pin TXD berfungsi untuk mengirimkan data serial, sedangkan RXD kebalikannya yaitu sebagai *pin* yang berfungsi untuk menerima data serial.
- ✓ Interrupt (INT0 dan INT1) merupakan *pin* dengan fungsi khusus sebagai interupsi hardware. Interupsi biasanya digunakan sebagai selaan dari program, misalkan pada saat program berjalan kemudian terjadi interupsi *hardware/software* maka program utama akan berhenti dan akan menjalankan program interupsi.

Tabel 2.3 Port D

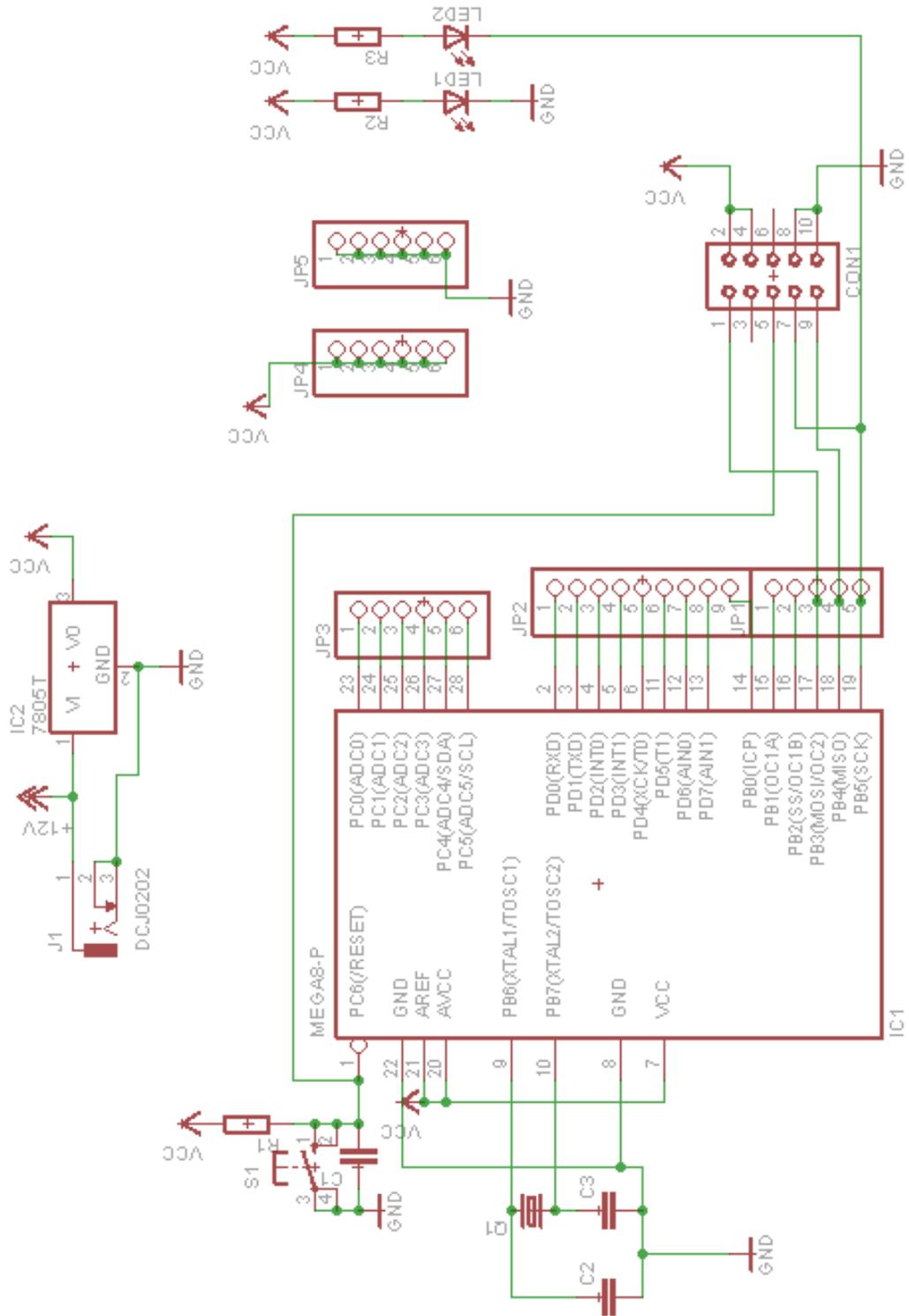
Port Pin	Alternate Function
PD7	AIN1 (analog comparator negative input) PCINT23 (pin change interrupt 23)
PD6	AIN0 (analog comparator positive input) OC0A (timer/counter0 output compare match A output) PCINT22 (pin change interrupt 22)

Port Pin	Alternate Function
PD5	T1 ( <i>timer/counter 1 external counter input</i> ) OC0B ( <i>timer/counter0 output compare match B output</i> ) PCINT21 ( <i>pin change interrupt 21</i> )
PD4	XCK ( <i>USART external clock input/output</i> ) T0 ( <i>timer/counter 0 external counter input</i> ) PCINT20 ( <i>pin change interrupt 20</i> )
PD3	INT1 ( <i>external interrupt 1 input</i> ) OC2B ( <i>timer/counter 2, output compare match B output</i> ) PCINT19 ( <i>pin change interrupt 19</i> )
PD2	INT0 ( <i>external interrupt 0 input</i> ) PCINT18 ( <i>pin change interrupt 18</i> )
PD1	TXD ( <i>USART output pin</i> ) PCINT17 ( <i>pin change interrupt 17</i> )
PD0	RXD ( <i>USART input pin</i> ) PCINT16 ( <i>pin change interrupt 16</i> )

#### 1.4.4.8. Sistem Minimum ATmega 328

Mikrokontroler ATmega 328 memiliki sistem minimum untuk dapat beroperasi sesuai yang kita inginkan. Dalam membuat sistem minimum suatu rangkaian mikrokontroler ATmega 328 dibutuhkan beberapa komponen pendukung seperti pada Gambar 2.12 yang merupakan salah satu sistem minimum dari:

1. Resistor 1k dan 470 berjumlah masing-masing 1
2. Kapacitor 4,7uF dan 22pF berjumlah masing-masing 1
3. Krystal 16Mhz jumlah 1
4. *Pin Male/Female Header* jumlah 2
5. DC *jack* jumlah 1
6. Regulator Tegangan 7805 jumlah 1
7. LED 3mm jumlah 2 serta Push Button kaki 4 jumlah 1

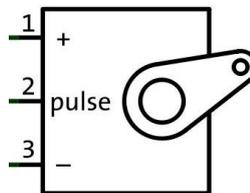


Gambar 2.12 Skema Rangkaian Sistem Minimum ATmega 328 (Chaerul, 2018)

## 1.5. Motor Servo

### 1.5.1. Deskripsi Motor Servo

Motor servo adalah sebuah motor dengan sistem *closed feedback* di mana posisi dari motor akan diinformasikan kembali ke rangkaian kontrol yang ada di dalam motor servo. Motor ini terdiri dari sebuah motor, serangkaian *gear*, potensiometer dan rangkaian kontrol. Potensiometer berfungsi untuk menentukan batas sudut dari putaran servo. Sedangkan sudut dari sumbu motor servo diatur berdasarkan lebar pulsa yang dikirim melalui kaki sinyal dari kabel motor. Pada Gambar 2.13 motor servo memiliki 3 buah kaki yaitu, Vcc, Gnd, dan Pulse.



**Gambar 2.13** Motor Servo (Sumozade, 2016)

Keunggulan dari penggunaan motor servo adalah :

- Tidak bergetar, tidak berisik dan tidak ber-resonansi saat beroperasi dengan kecepatan tinggi.
- Daya yang dihasilkan sebanding dengan ukuran dan berat motor.
- Penggunaan arus listrik sebanding dengan beban yang diberikan.
- Resolusi dan akurasi dapat diubah dengan hanya mengganti *encoder* yang dipakai.

Kelemahan dari penggunaan motor servo adalah :

- Harga dari motor servo relatif lebih mahal daripada motor DC lainnya.
- Bentuknya lebih besar karena satu paket.

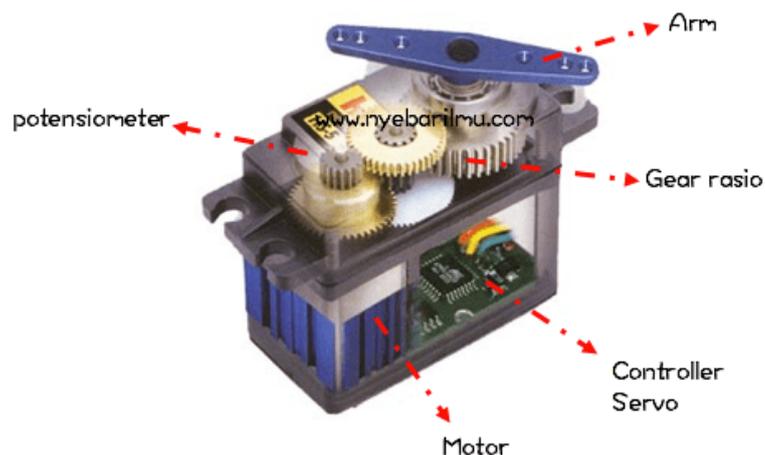
### 1.5.2. Aplikasi Motor Servo

Motor servo dapat dimanfaatkan pada pembuatan robot, salah satunya sebagai penggerak kaki robot. Motor servo dipilih sebagai penggerak pada kaki robot karena motor servo memiliki tenaga atau torsi yang besar,

sehingga dapat menggerakkan kaki robot dengan beban yang cukup berat. Pada umumnya motor servo yang digunakan sebagai penggerak pada robot adalah motor servo 180°.

### 1.5.3. Komponen Penyusun Motor Servo

Motor servo pada dasarnya dibuat menggunakan motor DC yang dilengkapi dengan *controler* dan sensor posisi sehingga dapat memiliki gerakan 0°, 90°, 180° atau 360°. Berikut adalah komponen *internal* sebuah motor servo 180°. Tiap komponen pada motor servo diatas masing-masing memiliki fungsi sebagai *controler*, *driver*, *sensor*, *gearbox* dan aktuator. Pada Gambar 2.14 terlihat beberapa bagian komponen motor servo. Motor pada sebuah motor servo adalah motor *DC* yang dikendalikan oleh bagian *controler*, kemudian komponen yang berfungsi sebagai sensor adalah potensiometer yang terhubung pada sistem *gearbox* pada motor servo.

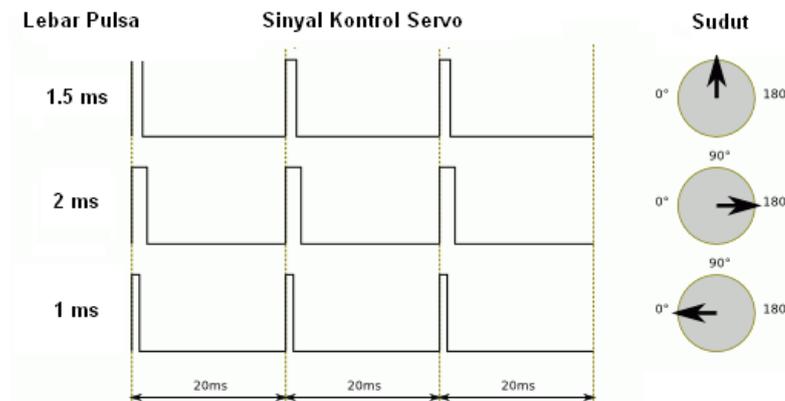


**Gambar 2.14** Konstruksi Motor Servo (Dave, 2019)

### 1.5.4. Prinsip Kerja Motor Servo

Motor servo dikendalikan dengan memberikan sinyal modulasi lebar pulsa *PWM* (*Pulse Wide Modulation*) melalui kabel control. Besarnya sumber tegangan tergantung dari spesifikasi motor servo yang digunakan. Lebar pulsa sinyal *control* yang diberikan akan menentukan posisi sudut putaran dari poros motor yang dapat dilihat pada Gambar 2.15. Hal tersebut

dapat dilakukan dengan mengirimkan pulsa kontrol dengan frekuensi 50 Hz dengan periode 20ms dan *duty cycle* yang berbeda. Dimana untuk menggerakkan motor servo sebesar  $90^\circ$  diperlukan pulsa dengan *ton duty cycle* pulsa positif 1,5ms dan untuk bergerak sebesar  $180^\circ$  diperlukan lebar pulsa 2ms.



**Gambar 2.15** Pulsa Kendali Motor Servo (Saripudin, 2011)

## 1.6. Rotary Encoder KY-040

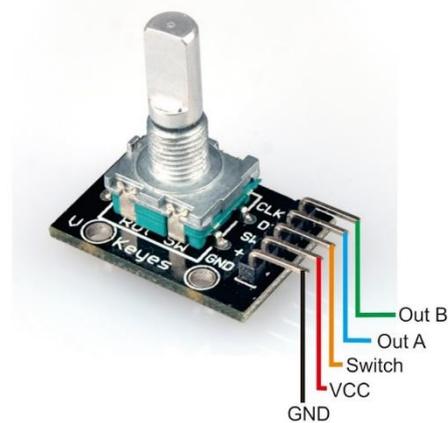
### 1.6.1. Pengertian Rotary Encoder

*Rotary Encoder* KY-040 adalah perangkat *input* putar (seperti pada knop) yang memberikan indikasi seberapa besar knop diputar dan ke arah mana ia berputar. Perangkat ini dapat digunakan untuk mengontrol motor stepper dan servo serta dapat digunakan untuk mengontrol perangkat seperti potensiometer digital. *Rotary encoder* memiliki sejumlah posisi per revolusi. Posisi-posisi ini mudah dirasakan sebagai "klik" kecil pada saat memutar *encoder*.

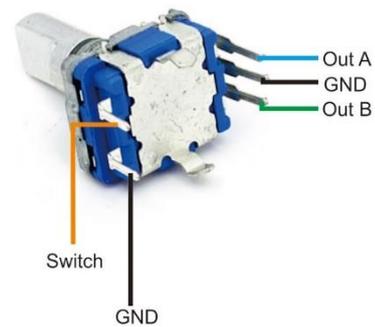
Berdasarkan pada Gambar 2.16 di satu sisi bagian dari *rotary encoder* ada tiga *pin*. Mereka biasanya disebut sebagai A, B dan C. Dalam kasus KY-040, mereka berorientasi seperti yang ditunjukkan. Di dalam *encoder* ada dua *switch*. Sekali saklar menghubungkan pin A ke pin C dan saklar lainnya menghubungkan pin B ke C. Di setiap posisi *encoder*, kedua sakelar dibuka atau ditutup. Setiap klik menyebabkan sakelar ini mengubah status sebagai berikut:

- Jika kedua sakelar ditutup, memutar *encoder* searah jarum jam atau berlawanan arah jarum jam satu posisi akan menyebabkan kedua sakelar terbuka
- Jika kedua sakelar terbuka, memutar *encoder* searah jarum jam atau berlawanan arah jarum jam satu posisi akan menyebabkan kedua sakelar tertutup.

With Breakout Board

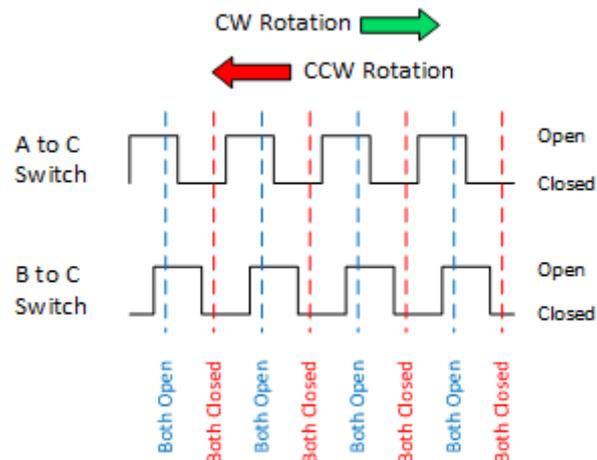


Without Breakout



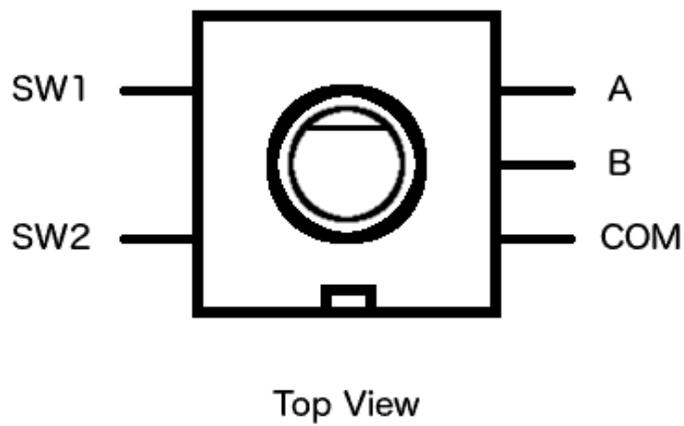
**Gambar 2.16** Ilustrasi Pin pada *Rotary Encoder* (Henry, 2018)

Fase pada *Rotary Encoder* secara mendasar, menentukan sakelar mana yang berubah terlebih dahulu adalah bagaimana arah rotasi ditentukan. Jika A berubah keadaan terlebih dahulu, sakelar berputar searah jarum jam. Jika B mengubah keadaan terlebih dahulu, sakelar berputar berlawanan arah jarum jam seperti yang terlihat pada Gambar 2.17.



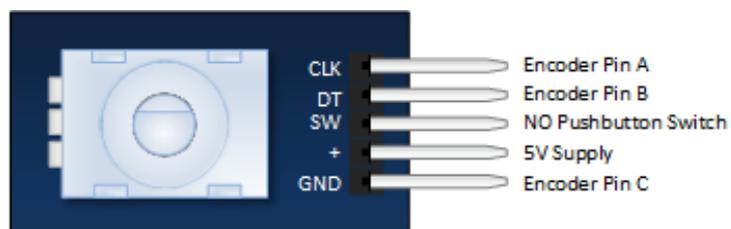
**Gambar 2.17** Gelombang Pembukaan dan Penutupan dari *Rotary Encoder* (Henry, 2018)

1.6.2. Skema Rangkain *Rotary Encoder*



**Gambar 2.18** Skema Rangkaian *Rotary Encoder* (Henry, 2018)

1.6.3. *Rotary Encoder Pin Out*



**Gambar 2.19** Penjelasan Pin pada *Rotary Encoder* (Henry, 2018)

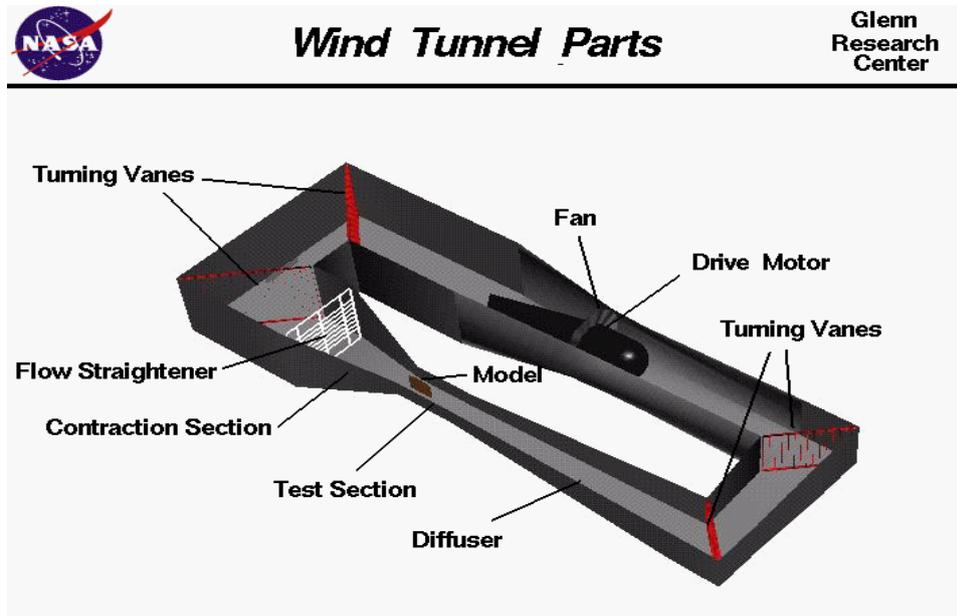
Seperti yang dapat dilihat pada Gambar 2.18 dan Gambar 2.19 modul ini dirancang sedemikian rupa sehingga *output* dengan *input* rendah ketika sakelar ditutup dan *input* tinggi saat sakelar terbuka. *Input* rendah dihasilkan dengan menempatkan *ground* pada *Pin C* dan meneruskannya ke pin CLK dan DT ketika sakelar ditutup. *Input* tinggi dihasilkan dengan *input* pasokan 5V dan resistor *pullup*, sehingga CLK dan DT keduanya tinggi ketika sakelar terbuka. Tidak disebutkan sebelumnya adalah adanya saklar tombol yang merupakan bagian integral dari *encoder*. Pada saat poros ditekan, saklar yang biasanya terbuka akan menutup. Fitur ini berguna jika ingin mengubah fungsi sakelar.

## 2.7. Wind Tunnel

*Wind Tunnel* (Terowongan Angin) adalah alat penghasil angin buatan yang digunakan untuk mengetahui aliran udara di sekitar benda yang ingin diuji serta untuk mengetahui gaya gaya yang ditimbulkan. Ahli aerodinamika menggunakan *wind tunnel* untuk menguji model pesawat yang diusulkan. Di dalam *wind tunnel*, insinyur dapat dengan hati-hati mengontrol kondisi aliran yang mempengaruhi gaya pada pesawat. Dengan melakukan pengukuran yang cermat terhadap gaya pada model, insinyur dapat memprediksi kekuatan pada pesawat skala penuh. Dan dengan menggunakan teknik diagnostik khusus, insinyur dapat lebih memahami dan meningkatkan kinerja pesawat. *Wind tunnel* biasanya dirancang untuk tujuan dan rentang kecepatan tertentu.

Terdapat dua jenis *wind tunnel* yaitu sistem terbuka (*open jet type*) dan sistem tertutup (*close jet type*). Perbedaan dari kedua jenis *tunnel* ini hanya terdapat pada ruang percobaannya seperti yang terlihat pada Gambar 2.20 dan Gambar 2.21. Pada sistem terbuka penampang ruang percobaannya lebih lebar, sedangkan pada sistem tertutup ruang percobaan merupakan suatu ruangan yang sama sekali tertutup.

## 1. Sistem Tertutup



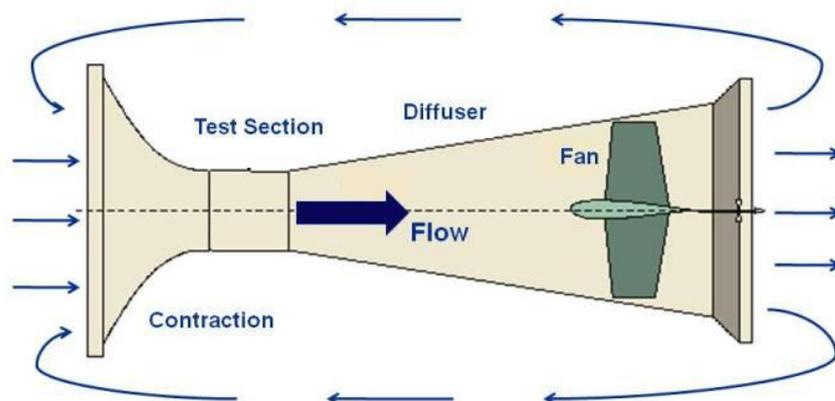
Gambar 2.20 Close jet type Wind Tunnel (NASA, 2015)

## 2. Sistem Tertutup

National Aeronautics and Space Administration



## Open Return Wind Tunnel



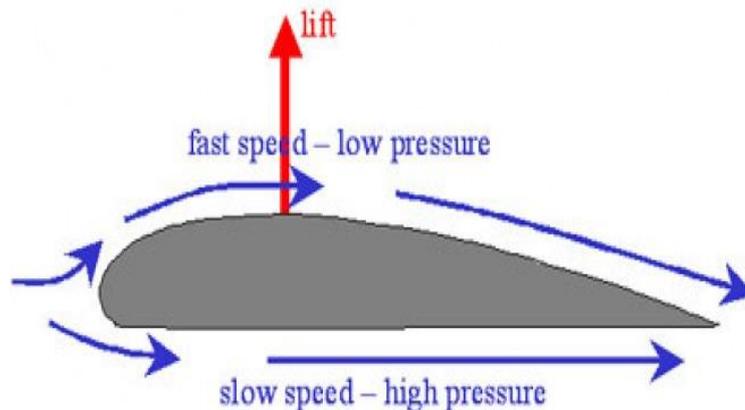
www.nasa.gov

Gambar 2.21 Open jet type Wind Tunnel (NASA, 2015)

## 2.8. Lift (Gaya Angkat) dan Drag (Gaya Hambat)

### 2.8.1. Lift (Gaya Angkat)

*Lift* merupakan gaya yang dapat membuat pesawat terangkat ke udara. Dengan memanfaatkan *drag* yang dihasilkan oleh sayap sehingga udara mengalir kebagian bawah sayap akan menghasilkan gaya angkat dan menerbangkan pesawat tersebut. Dengan bentuk Airfoil sayap yang telah dirancang sedemikian rupa membuat kecepatan udara diatas sayap lebih tinggi daripada kecepatan udara dibagian bawah sayap sehingga tekanan udara di bagian atas sayap lebih rendah dibandingkan dengan dibagian bawah sayap seperti yang dapat dilihat pada Gambar 2.22, hal tersebut mengakibatkan udara akan mengangkat pesawat keatas, sesuai dengan hukum bernoulli yang memang menjadi dasar acuan gaya angkat pesawat.

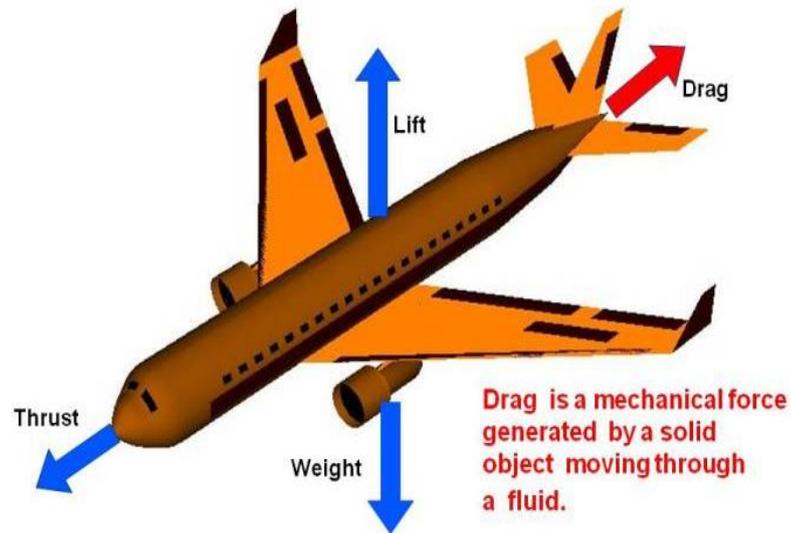


**Gambar 2.22** Lift pada Airfoil (Jack, 2015)

### 2.8.2. Drag (Gaya Hambat)

Gaya yang bekerja berlawanan dengan *lift* atau menghambat gerakan pesawat dengan menghasilkan gaya gesek sehingga menahan laju pesawat disebut *drag*. Hal yang mempengaruhi *drag* pada pesawat adalah *fuselage* dari pesawat itu sendiri, tetapi *drag* juga dapat diperoleh dari *secondary flight control* seperti *spoiler*, *flap*, dan *slat*. *Drag* dinilai sangat merugikan karena dapat menghambat laju pesawat sehingga konsumsi bahan bakar menjadi tinggi, tetapi juga dapat sangat bermanfaat apabila pesawat sedang melakukan proses *landing* dan *brake*. Untuk meminimalisir gaya ini, pesawat

dirancang agar jalannya udara tidak terlalu terhambat dengan *body* pesawat itu sendiri,



**Gambar 2.23** Drag pada Pesawat (Rawi, 2012)

### 2.8.3. Rumus *Lift* dan *Drag*

*Lift* adalah angka yang digunakan oleh para aerodinamika untuk memodelkan semua dependensi bentuk, kemiringan, dan beberapa kondisi aliran pada pengangkatan yang rumit. *Lift* ( $L$ ) dapat dihitung dengan menggunakan rumus:

$$L = \frac{C_L \rho V^2 A}{2} \dots\dots\dots (2.1)$$

$C_L$  = Koefisien *Lift*

$L$  = Gaya Angkat

$\rho$  = Kerapatan Udara (Density)

$V$  = Kecepatan Udara (m/s)

$A$  = Luas Penampang ( $m^2$ )

*Drag* adalah angka yang digunakan oleh para aerodinamik untuk memodelkan semua dependensi bentuk, kemiringan, dan kondisi aliran yang rumit pada hambatan pesawat. *Drag* ( $D$ ) dapat dihitung dengan menggunakan rumus:

$$D = \frac{C_D \rho V^2 A}{2} \dots\dots\dots (2.2)$$

$C_d$  = Koefisien *Drag*

D = Gaya Hambat

$\rho$  = Kerapatan Udara (Density)

V = Kecepatan Udara (m/s)

A = Luas Penampang (m<sup>2</sup>)

Besarnya koefisien *lift* dan koefisien *drag* ditentukan oleh jenis *airfoil* dan sudut serang.