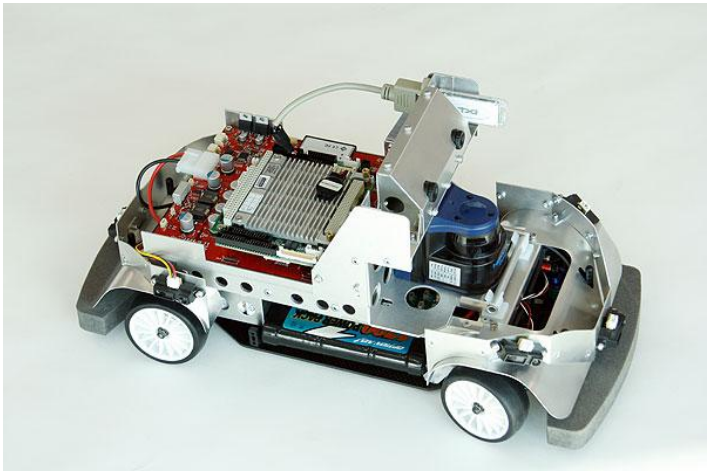
BAB II
LANDASAN TEORI

Berdasarkan jenisnya robot terbagi menjadi enam yaitu :

1. Robot Mobil (Bergerak) yang bisa berpindah tempat.

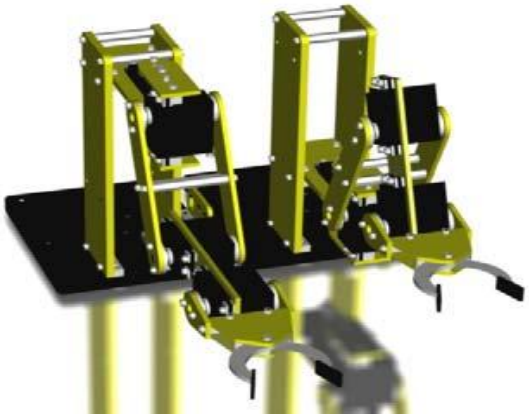
Robot Mobil atau Mobile Robot yang terdapat pada Gambar 2.1 adalah konstruksi robot yang ciri khasnya adalah mempunyai aktuator berupa roda untuk menggerakkan keseluruhan badan robot tersebut, sehingga robot tersebut dapat melakukan perpindahan posisi dari satu titik ke titik yang lain. Robot mobil ini sangat disukai bagi orang yang mulai mempelajari robot. Hal ini karena membuat robot mobil tidak memerlukan kerja fisik yang berat. Untuk dapat membuat sebuah robot mobile minimal diperlukan pengetahuan tentang mikrokontroler dan sensor-sensor elektro [4]. Contoh robot mobil bergerak sebagai berikut :



Gambar 2.1 *Contoh Robot mobil bergerak*

2. Robot Manipulator (tangan)

Pada Gambar 2.2 ini robot hanyak memiliki satu tangan seperti tangan manusia yang fungsinya untuk memegang atau memindahkan barang, contoh robot ini adalah robot las di Industri mobil, robot merakit elektronik dan lain-lain [4]. Contoh Robot Manipulator sebagai berikut :



Gambar 2.2 *Contoh Robot Manipulator Tangan*

3. Robot Humanoid

Pada Gambar 2.3 yaitu robot yang miliki kemampuan menyerupai manusia, baik fungsi maupun cara bertindak, contoh robot ini adalah Ashimo yang dikembangkan oleh Honda [4]. Contoh Robot Humanoid sebagai berikut :



Gambar 2.3 *Contoh Robot Humanoid*

4. Robot Berkaki

Pada Gambar 2.3 robot ini memiliki kaki seperti hewan atau manusia, yang mampu melangkah, seperti robot serangga, robot kepiting dan lain-lain [4]. Contoh Robot Berkaki ini sebagai berikut :



Gambar 2.4 *Contoh Robot Berkaki*

5. Flying Robot (Robot Terbang)

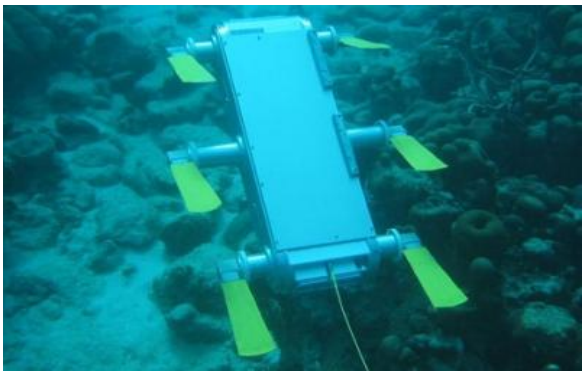
Pada Gambar 2.5 yaitu robot yang mampu terbang, robot ini menyerupai pesawat model yang deprogram khusus untuk memonitor keadaan di tanah dari atas, dan juga untuk meneruskan komunikasi [4]. Contoh Flying Robot (Robot Terbang) sebagai berikut :



Gambar 2.5 *Contoh Flying Robot(Robot Terbang)*

6. Under Water Robot (Robot dalam air)

Pada Gambar 2.6 robot ini digunakan di bawah laut untuk memonitor kondisi bawah laut dan juga untuk mengambil sesuatu di bawah laut [4]. Contoh Water Robot (Robot dalam air) sebagai berikut :



Gambar 2.6 *Contoh Under Water Robot (Robot dalam air)*

Dalam bab ini akan dijelaskan mengenai teori-teori penunjang untuk perancangan robot berkaki baik teori perangkat keras maupun perangkat lunak yang akan digunakan, sebagai berikut:

2.1 PERANGKAT KERAS (*hardware*)

2.1.1. Mikrokontroler *Scenix SX28AC/DP*

Mikrokontroler menjadi sangat penting sebagai pengambil keputusan. Pergerakan robot diatur oleh mikrokontroler berdasarkan hasil pemindahan tempat robot berada yang dilakukan oleh sensor-sensor. Mikrokontroler yang digunakan pada robot ini adalah Scenix SX28AC/DP dengan modul pengendali Basic Stamp.

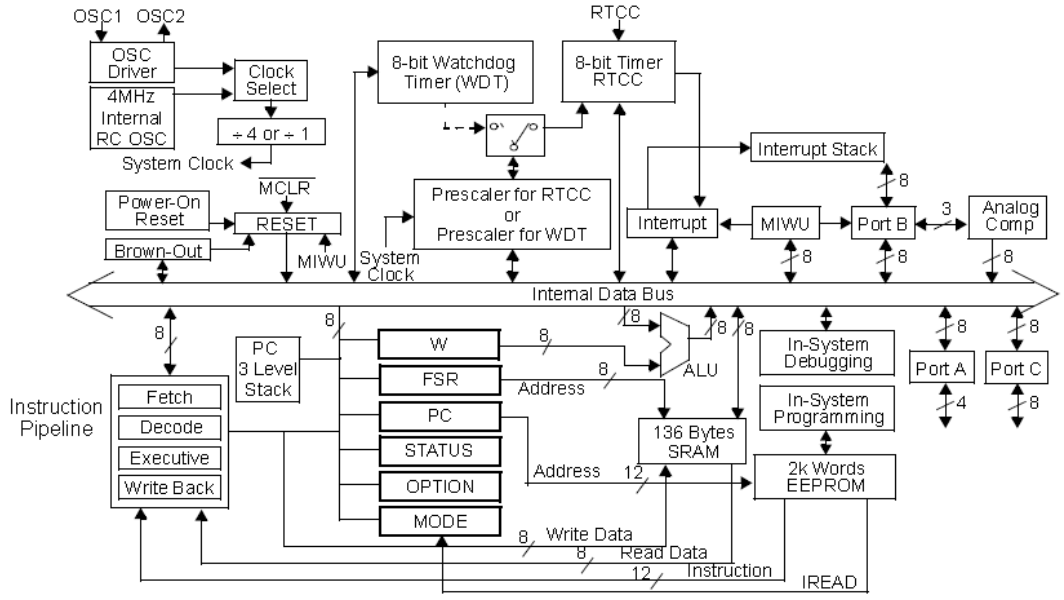
Beberapa alasan utama pemilihan mikrokontroler Scenix SX28AC/DP:

1. Bahasa pemrograman yang sederhana membuat pengembangan menjadi lebih cepat.
2. Kecepatan tinggi dengan frekuensi *clock* 50 MHz.
3. Mampu mengeksekusi 10000 instruksi/detik.
4. Jumlah *port* I/O sebanyak 20 buah, 4 *port* untuk komunikasi serial dan 16 *port* mencukupi kebutuhan sensor-sensor dan aktuator-aktuator.
5. Kapasitas memori program EE/Flash 2048 Word (2k Word).
6. Memori data berukuran 136x8bit SRAM.

7. Rentang tegangan (Vcc) yang digunakan antara 3.0 – 5.5 volt tetapi umumnya menggunakan level tegangan 5 volt.
8. Suplai arus yang melewati Vcc maksimal sebesar 130 mA.

Fitur-fitur lain yang dimiliki Scenix SX28AC/DP adalah:

1. 13.3 ns untuk satu siklus instruksi
2. Merupakan mikrokontroler RISC, sehingga memiliki 43 *single-word* instruksi dasar.
3. 8-bit *Real Time Clock/Counter* (RTCC) dengan 8-bit *prescaler* terprogram.
4. Komparator analog yang terdapat pada *port* B (RB0 out, RB1 IN-, dan RB2 IN+).
5. *Watchdog Timer* dengan isolator internal yang mempunyai frekuensi antara 31,25 kHz sampai 4 MHz.
6. Kemasan fisik 28 pin DIP *(Dual Inline Package).*



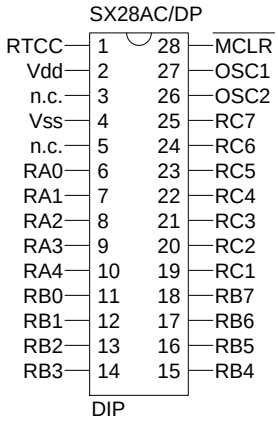
Gambar 2.7 *Diagram blok SX28AC/DP*

Arsitektur keluarga SX menggunakan modifikasi arsitektur Harvard. Arsitektur ini menggunakan dua memori terpisah dengan bus alamat yang terpisah, satu untuk program dan satu untuk data yang mengizinkan transfer data dari memori program ke SRAM. Kemampuan ini mengizinkan

pengaksesan data dari memori program. Keuntungan dari arsitektur ini adalah transfer instruksi *fetch* dan memori dapat di *overlap* dengan sebuah *multi-stage pipeline* (*fetch, decode, execute* dan *write back*), yang berarti instruksi selanjutnya dapat di-*fetch* dari memori program ketika instruksi sekarang sedang dieksekusi menggunakan data dari memori data [2].

2.1.2. Deskripsi pin-pin SX28AC/DP

Pada SX28AC/DP memiliki *port* untuk *input*/*output* sensor-sensor dan aktuator sebanyak 20 pin, yang terdiri dari 4 pin sebagai *port* A, 8 pin sebagai *port* B, dan 8 pin sebagai *port* C. Khusus untuk *port* A (pin 6, 7, 8, dan 9) digunakan sebagai *port* I/O jalur pemrograman mikrokontroler. Pin 6 dan pin 7 secara berturut-turut berfungsi sebagai jalur data serial, dan *input clock* serial. Pin 8 dan pin 9 berfungsi sebagai pin reset atau pin Vpp pada saat mode pemrograman. Komunikasi serial atau *USB* dapat digunakan untuk mengisikan program ke mikrokontroler [8].



Gambar 2.8 *Konfigurasi pin SX28AC/DP*

Berikut adalah daftar pin-pin pada mikrokontroler SX28AC/DP beserta fungsinya:

Tabel 2.1. *Deskripsi pin SX28AC/DP*

No. Pin	Nama	Tipe Pin	Level *Input*	Fungsi
1	RTCC	I	ST	*Input to Real-Time Clock/Counter*
2	Vdd	P	-	Tegangan sumber 5 volt
3	n.c.			*No connection*
4	Vss	P	-	*Ground*
5	.c.			*No connection*
6	RA0	I/O	TTL/CMOS	Pin I/O dua arah; *Symmetrical source/sink capability*
7	RA1	I/O	TTL/CMOS	Pin I/O dua arah; *Symmetrical source/sink capability*
8	RA2	I/O	TTL/CMOS	Pin I/O dua arah; *Symmetrical source/sink capability*
9	RA3	I/O	TTL/CMOS	Pin I/O dua arah; *Symmetrical source/sink capability*
10	RB0	I/O	TTL/CMOS/ST	Pin I/O dua arah; *comparator output; MIWU/Interrupt input*
11	RB1	I/O	TTL/CMOS/ST	Pin I/O dua arah; *comparator negative input; MIWU/Interrupt input*
12	RB2	I/O	TTL/CMOS/ST	Pin I/O dua arah; *comparator positive input; MIWU/Interrupt input*
13	RB3	I/O	TTL/CMOS/ST	Pin I/O dua arah; MIWU/*Interrupt input*
14	RB4	I/O	TTL/CMOS/ST	Pin I/O dua arah; MIWU/*Interrupt input*
15	RB5	I/O	TTL/CMOS/ST	Pin I/O dua arah; MIWU/*Interrupt input*
16	RB6	I/O	TTL/CMOS/ST	Pin I/O dua arah; MIWU/*Interrupt input*
17	RB7	I/O	TTL/CMOS/ST	Pin I/O dua arah; MIWU/*Interrupt input*
18	RC0	I/O	TTL/CMOS/ST	Pin I/O dua arah
19	RC1	I/O	TTL/CMOS/ST	Pin I/O dua arah

20	RC2	I/O	TTL/CMOS/ST	Pin I/O dua arah
21	RC3	I/O	TTL/CMOS/ST	Pin I/O dua arah
22	RC4	I/O	TTL/CMOS/ST	Pin I/O dua arah
23	RC5	I/O	TTL/CMOS/ST	Pin I/O dua arah
24	RC6	I/O	TTL/CMOS/ST	Pin I/O dua arah
25	RC7	I/O	TTL/CMOS/ST	Pin I/O dua arah
26	OSC2	O	CMOS	*Output* osilator kristal – dalam mode R/C
27	OSC1	I	ST	*Input* osilator kristal – *input* sumber *clock* eksternal
28	$\overline{MCLR}$	I	ST	*Master Clear reset input – active low*
Catatan: I = *Input*, O = *Output*, I/O = *Input/Output*, P = *Power*, TTL = TTL *input*, CMOS = CMOS *input*, ST = *Schmitt Trigger input*, MIWU = *Multi-Input Wakeup input*				

2.1.3. Organisasi memori

Memori pada SX28AC/DP terbagi menjadi dua blok memori, yaitu memori program dan memori data. Ukuran memori program adalah 2k word dengan lebar 12-bit word yang merupakan memori bersifat *Erasable Electrical Programmable Read Only Memory* (EEPROM). Sedangkan ukuran memori data sebesar 136 bytes pada RAM, dikelompokkan menjadi 8 *bank* dari 16 register dan 8 register *not bank*. Keduanya bisa dialamati baik secara langsung maupun tidak langsung menggunakan *file select register* (FSR). *Special-function register* dipetakan ke dalam memori data [8].

2.1.4 Modul Penggerak

Dalam modul penggerak akan bergerak sesuai dengan signal pulsa yang dikeluarkan mikrokontroler.

2.1.4.1. Motor Servo Standard

Sebuah motor servo standard adalah alat yang dapat mengendalikan posisi, dapat membelokkan dan menjaga suatu posisi berdasar penerimaan pada suatu signal elektronik itu. Karena motor DC servo merupakan alat untuk mengubah energi listrik menjadi energi mekanik, maka magnit permanent motor DC servolah yang mengubah energi listrik ke dalam energi mekanik melalui interaksi dari dua medan magnit. Salah satu medan dihasilkan oleh magnit permanent dan yang satunya dihasilkan oleh arus yang mengalir dalam kumparan motor. Resultan dari dua medan magnit tersebut menghasilkan torsi yang membangkitkan putaran motor tersebut. Saat motor berputar, arus pada kumparan motor menghasilkan torsi yang nilainya konstan [9].

Motor servo merupakan sebuah motor dc kecil yang diberi sistem gear, rangkaian kontrol dan potensiometer sehingga dia dapat menempatkan ”horn” yang dikehendaki bisa dipertanahankan. Fungsi potensiometer adalah 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.

Gambar 2.9 menunjukkan bagian-bagian dari sebuah motor servo standard. Adapun bagian-bagian tersebut adalah sebagai berikut:

1) Konektor yang digunakan untuk menghubungkan motor servo kepada suatu sumber (Vdd dan Vss) dan suatu sumber signal I/O yang diletakkan pada BASIC Stamp.
2) Kabel menghubungkan Vdd, Vss dan garis signal dari konektor ke dalam motor servo itu.
3) Tuas menjadi bagian dari motor servo yang kelihatan seperti suatu bintang *four-pointed*. Ketika motor servo sedang sedang berputar, tuas motor servo akan bergerak ke bagian yang dikendalikan dalam program di BASIC Stamp.

4) *Cassing* berisi bagian untuk mengendalikan kerja motor servo yang pada dasarnya berupa motor DC dan *gear*. Bagian ini bekerja untuk menerima instruksi dari BASIC Stamp dan mengkonversinya ke dalam sebuah pulsa untuk menentukan arah/posisi servo.

(1) Konektor

(2) Kabel

(3) Tuas

(4) *Cassing*



Gambar 2.9 *Servo Standar*

Pada umumnya standard servo mempunyai 3 warna kabel, yaitu:

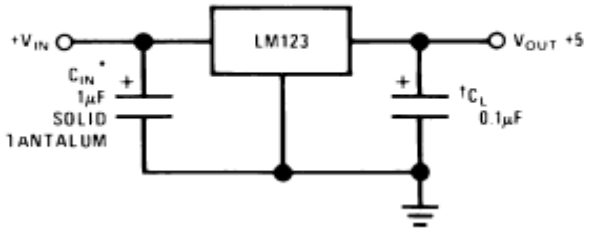
1. warna merah menandakan positif
2. warna coklat menandakan negatif
3. warna orange menandakan inputan data

2.1.5. Catu daya

Setiap rangkaian elektronik didesain untuk beroperasi pada tegangan tertentu dalam keadaan konstan. Regulator tegangan menyediakan *output* tegangan dc yang konstan dan secara terus menerus menahan tegangan *output* pada nilai yang diinginkan. Regulator ini hanya dapat bekerja jika tegangan *input* (V_{in}) lebih besar daripada tegangan *output* (V_{out}). Biasanya perbedaan tegangan *input* dengan *output* yang direkomendasikan tertera pada *datasheet* komponen tersebut.

Dalam hal ini sumber tegangan yang diperoleh berasal dari luar yang terhubung dengan dua buah regulator yang dirangkai secara paralel, yaitu regulator untuk mikrokontroler dan regulator untuk motor servo. Regulator

ini dibuat terpisah agar pada saat motor servo mengambil arus yang besar, sehingga sumber arus yang dipergunakan oleh mikrokontroler tidak berkurang [5].



Gambar 2.10 *Rangkaian Regulator*

2.2 PERANGKAT LUNAK (*Software*)

Perangkat lunak merupakan faktor penentu paling akhir dalam tahap perancangan robot. Perangkat lunak ini berupa algoritma gerak dan tugas robot dalam bentuk listing program yang ditanamkan ke dalam mikrokontroler. Program dapat bermacam-macam bentuk versi dan bahasa pemrogramannya, sesuai dengan spesifikasi dari mikrokontroler yang digunakan [8].

2.2.1 Mikrokontroler SX28AC/DP

Mikrokontroler SX28AC/DP adalah jenis mikrokontroler yang didalamnya sudah terdapat *interpreter chip*. Di dalam *interpreter chip* sudah terdapat program kecil yang berfungsi untuk menginterpretasikan program yang ditulis di dalam *Basic Stamp Editor*. Bahasa pemrograman yang digunakan untuk mengisikan program ke dalam mikrokontroler ini adalah Bahasa PBasic, yang bahasa pemrogramannya hampir sama dengan bahasa Basic [8].

2.2.2. Ringkasan Instruksi

Instruksi yang dapat digunakan pada editor basic stamp relatif cukup banyak dan bergantung dari type dan jenis *basic stamp* yang digunakan.

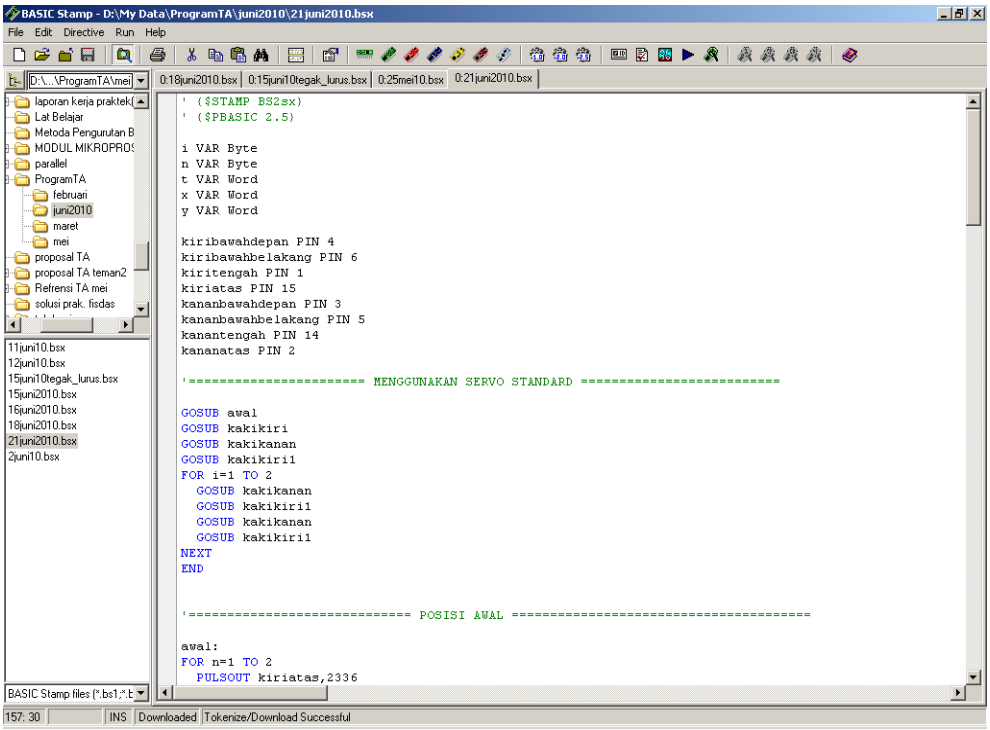
Berikut ini beberapa instruksi-instruksi dasar yang dapat digunakan pada mikrokontroler *basic stamp* dengan type BS2SX [8].

Tabel 2.2. *Beberapa instruksi dasar BS2SX*

Instruksi	**Keterangan**
DO…LOOP	Perulangan
GOSUB	Memanggil prosedur
IF…THEN	Percabangan
SELECT…CASE	Pencabangan
FOR..NEXT	Perulangan
HIGH	Menset pin I/O menjadi 1
LOW	Menset pin I/O menjadi 0
PAUSE	Delay atau waktu tunda
PWM	Konversi suatu nilai digital ke keluaran analog lewat pulse-width modulasi
COUNT	Menghitung jumlah pulsa (0-1-0 atau 1-0-1)
PULSOUT	Membangkitkan pulsa
PULSIN	Menerima/membaca pulsa yang diterima
GOTO	Menuju/loncat ke alamat memori tertentu

2.2.3. Editor Basic Stamp

Editor Basic Stamp merupakan sebuah software yang dapat dijalankan pada dua jenis versi sistem operasi, yakni DOS dan Windows. Software ini dapat berjalan pada komputer dengan sistem minimum, tanpa harus membutuhkan spesifikasi komputer yang sangat canggih. Berikut ini tampilan jendela program editor Basic Stamp yang berjalan pada sistem operasi windows [8].



Gambar 2.11 *Tampilan Program Basic Stamp Editor*

Dari jendela editor basic stamp di atas terdapat beberapa bagian, diantaranya menu dan toolbar yang digunakan untuk melakukan operasi file seperti *New*, *Open*, *Save*, serta pengeditan *listing* program yang sedang dirancang. Gambar 2.12 merupakan bagian menu dan toolbar pada editor basic stamp.



Gambar 2.12 *Tampilan Menu/ToolBar Editor Basic Stamp*

Setiap file yang dibuka pada program editor Basic Stamp akan dibuatkan tab editor seperti yang terlihat pada gambar 2.13, sehingga memudahkan pengguna software untuk berpindah-pindah dari satu file ke file yang lain.



Gambar 2.13 *Contoh tampilan tab editor dengan 4 file terbuka*

Gambar 2.14 berikut memperlihatkan editor/lembar kerja dari editor Basic Stamp. Disinilah tempat penulisan *listing* program yang akan dibuat. Adapun warna-warna tulisan seperti yang terlihat dibawah tidak diwarnai secara manual, melainkan secara otomatis akan dilakukan oleh software sesuai dengan ketentuan penulisan *listing* program.

```
' {$STAMP BS2sx}
' {$PBASIC 2.5}

i VAR Byte
n VAR Byte

FOR n=1 TO 2
  PULSOUT kiriatas,2336
  PAUSE 5
  PULSOUT kiriatas,2336
  PAUSE 500
  PULSOUT kiritengah,1580
  PAUSE 500
  PULSOUT kiribawahbelakang,1600
  PAUSE 500
  PULSOUT kiribawahdepan,2050
  PAUSE 500
  PULSOUT kananatas,1435
  PAUSE 500
  FOR i=1 TO 2
    PULSOUT kanantengah,1889
    PULSOUT kanantengah,0
    PAUSE 5
  NEXT
  PAUSE 500
    FOR i=1 TO 2
      PULSOUT kanantengah,1500
      PULSOUT kanantengah,0
      PAUSE 5
    NEXT
  PAUSE 500
  PULSOUT kananbawahbelakang,1700
  PULSOUT kananbawahdepan,1408
  PAUSE 500
NEXT
```

Gambar 2.14 *Contoh editor Basic Stamp*

2.2.4. Cara Pembuatan program

Pembuatan *listing* program dengan menggunakan editor basic stamp diawali dengan menentukan jenis/tipe mikrokontroler basic stamp dan versi bahasa PBASIC yang digunakan. Hal ini sangatlah mudah karena cukup memilih/meng-klik icon jenis/tipe basic stamp dan versi bahasa PBASIC

yang akan kita gunakan pada toolbar, seperti yang terlihat pada gambar 2. berikut :

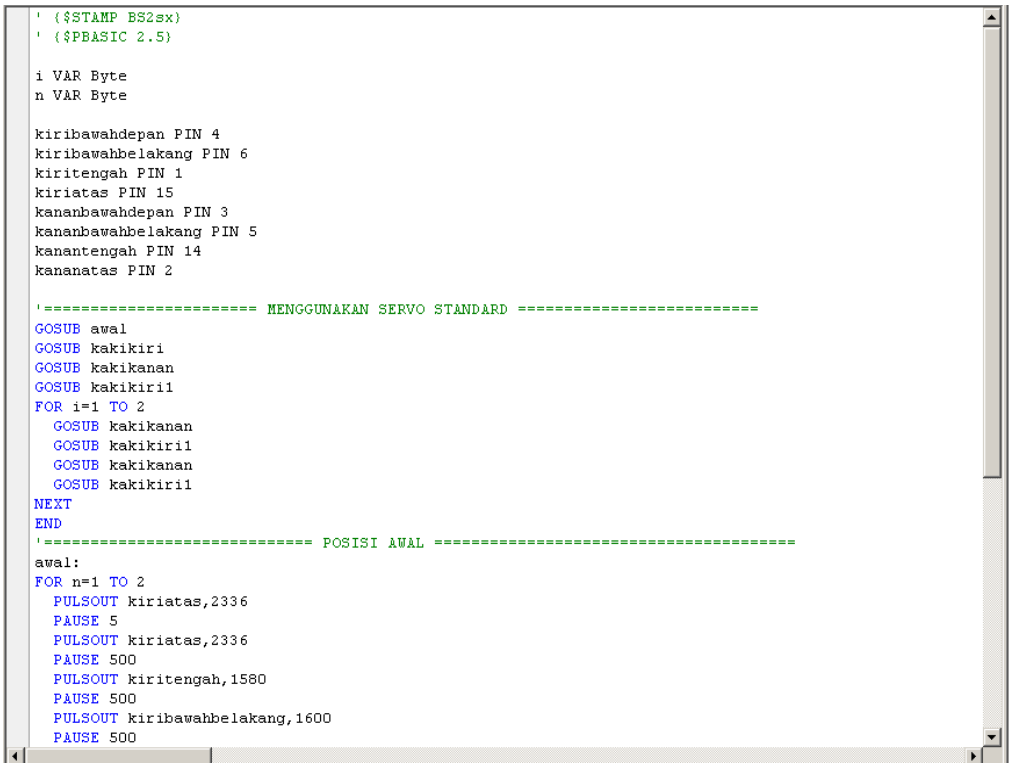


Gambar 2.15 *Icon type Basic Stamp dan versi bahasa PBASIC*



Gambar 2.16 *Tampilan jenis Basic Stamp dan versi bahasa PBASIC pada editor*

Setelah menentukan tipe/jenis Basic Stamp dan versi bahasa PBASIC yang akan digunakan, dilanjutkan dengan penulisan listing program. Ketentuan penulisan Listing program tergolong cukup mudah dan dapat diselesaikan dalam waktu yang relatif singkat, karena menggunakan bahasa pemrograman tingkat tinggi yakni PBASIC yang bahasa pemrogramannya hampir sama dengan BASIC. Dalam penulisan Listing program dapat ditambahkan komentar yang dapat membantu proses pengeditan jika terjadi kesalahan dengan algoritma program yang dibuat. Gambar 2.17 berikut ini contoh tampilan editor basic stamp yang berisi potongan program lengkap dengan deklarasi variabel dan komentar [8].



```
' {$STAMP BS2sx}
' {$PBASIC 2.5}

i VAR Byte
n VAR Byte

kiribawahdepan PIN 4
kiribawahbelakang PIN 6
kiritengah PIN 1
kiriatas PIN 15
kananbawahdepan PIN 3
kananbawahbelakang PIN 5
kanantengah PIN 14
kananatas PIN 2

'====================== MENGGUNAKAN SERVO STANDARD ========================
GOSUB awal
GOSUB kakikiri
GOSUB kakikanan
GOSUB kakikiri1
FOR i=1 TO 2
  GOSUB kakikanan
  GOSUB kakikiri1
  GOSUB kakikanan
  GOSUB kakikiri1
NEXT
END
'============================ POSISI AWAL ======================================
awal:
FOR n=1 TO 2
  PULSOUT kiriatas,2336
  PAUSE 5
  PULSOUT kiriatas,2336
  PAUSE 500
  PULSOUT kiritengah,1580
  PAUSE 500
  PULSOUT kiribawahbelakang,1600
  PAUSE 500
```

Gambar 2.17 *Contoh tampilan editor Basic Stamp yang berisi potongan program*

Dalam perancangan perangkat lunak, ada beberapa instruksi-instruksi umum yang sering digunakan, seperti perulangan, percabangan dan rutin/prosedur. Berikut ini ditunjukkan cara-cara penggunaan instruksi-instruksi umum pada mikrokontroler *basic stamp*.

1. Perulangan

Ada beberapa cara perulangan yang dapat digunakan pada mikrokontroler Basic Stamp, diantaranya :

- FOR…NEXT (perulangan yang dibatasi)

  Contoh :

  FOR i = 1 to 150 → awal perulangan
  Pulsout 1, 1000
  Pulsout 2, 2000 } isi perulangan
  Pause 20 → menunggu 20µs
  NEXT → akhir perulangan

- DO…LOOP (perulangan secara terus menerus)

Contoh :

```
DO                  →   awal perulangan
Pulsout 1, 1000     }
Pulsout 2, 2000     }   isi perulangan
Pause 20            }
LOOP                →   akhir perulangan
```

Perulangan DO…LOOP ini selain dapat digunakan untuk perulangan tanpa batas dapat juga digunakan pada perulangan yang dibatasi. Untuk jumlah perulangan yang dibatasi tinggal menambahkan UNTIL dengan syarat perulangan pada instruksi DO…LOOP. Contohnya :

```
DO                      →   awal perulangan
Pulsout 1, 1000         }
Pulsout 1, 700          }   isi perulangan
Pause 20                }
n = n + 1               }
LOOP UNTIL n = 5        →   akhir perulangan
```

2. Percabangan

Instruksi percabangan yang dapat digunakan pada mikrokontroler *Basic Stamp* ada beberapa jenis, diantaranya :

- IF…THEN

Syntax : **IF** *Condition* **THEN**
Statement(s)
ELSEIF *Condition(s)* **THEN**
Statement(s)
ELSE
Statement(s)
ENDIF

Contoh : IF IRkanan=1 AND IRkiri=0 THEN
DO

```
        GOSUB belok2
      LOOP UNTIL IRkanan=0
    ELSEIF IRkiri=1 AND IRkanan=0 THEN
      DO
        GOSUB belok1
      LOOP UNTIL IRkiri=i0
    ELSE IRkiri=1 AND IRkanan=1 THEN
     DO
        GOSUB mundur
      LOOP UNTIL IRkiri=0 OR IRkanan=0
    ENDIF
```

- SELECT…CASE

Syntax : SELECT *Expression*
CASE *Condition(s)*
Statement(s)
CASE ELSE
Statement(s)
ENDSELECT

```
Contoh :  SELECT ruang
            CASE 1
              derajat=15
              derajatmax=35
           CASE ELSE
              derajat=12
              derajatmax=33
          ENDSELECT
```

Dalam pemilihan instruksi percabangan yang akan digunakan disesuaikan dengan kebutuhan dan bentuk percabangan yang akan di buat.

3. Rutin/Prosedur

Prosedur merupakan sekumpulan instruksi yang karena berbagai pertimbangan dipisahkan dari program utama. Bagian-bagian di program utama akan memanggil prosedur, jadi mikrokontroler sementara akan meninggalkan aliran program utama untuk mengerjakan instruksi-instruksi dalam prosedur. Setelah selesai mengerjakan prosedur, maka mikrokontroler akan kembali ke aliran program utama.

Contoh :

Prosedur :

maju: ⟶ nama prosedur

pulsout 3,700
pulsout 4,1000 } isi prosedur
pause 20

return ⟶ keluar dari prosedur dan kembali ke baris instruksi setelah pemanggilan prosedur

Pemanggilan prosedur

…….. ⟶ baris instruksi lainnya

……..

GOSUB maju ⟶ baris instruksi pemanggilan prosedur

…….

4. Pengaksesan port I/O

Jumlah port I/O yang dapat digunakan pada mikrokontroler *Basic Stamp* berjumlah 16 buah, dimulai dari port 0 sampai port 15 yang masing-masing port dapat difungsikan sebagai input maupun output. Dalam pengaksesan (mengeset atau membaca) port I/O dapat langsung disebutkan/dituliskan nomor port I/O yang akan di akses.

Contoh :

HIGH 4 ⟶ mengeset port 4 menjadi high

LOW 5 ⟶ mengeset port 5 menjadi low

IF 6 = 1 THEN ⟶ membaca port 6 apakah high

IF 7 = 0 THEN ⟶ membaca port 7 apakah low

2.3 Cara Memprogram Mengendalikan Servo

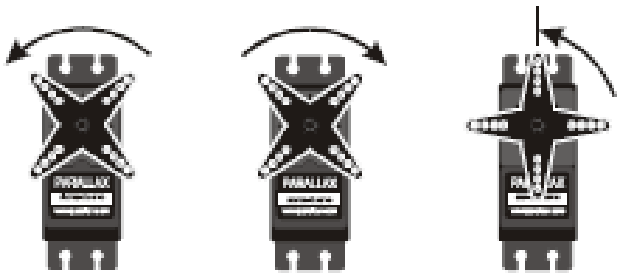
Sebuah motor servo dikendalikan oleh signal yang sangat tinggi. Signal pulsa tunda dikirimkan pada tiap-tiap 20 ms. Signal berlangsung selama 1 ms sampai 2 ms. Perintah PULSOUT dapat digunakan untuk mengirimkan sebuah pulsa dengan menggunakan BASIC Stamp pada pin I/O. Di bawah ini sintaksis perintah untuk PULSOUT:

PULSOUT *Pin, Duration* (PULSOUT Pin, Jangka waktu)

Mikrokontroler mengirimkan data berupa pulsa, yaitu data berupa perubahan HIGH dan LOW dengan jeda waktu pada sebuah pin, pin merupakan sesuatu yang dinamakan oleh BASIC Stamp sebagai I/O untuk mengirimkan dan menerima signal. Dengan mengirimkan pulsa antara 1ms ke 2ms 1/1000 sampai 2/1000 satuan detik untuk mengendalikan motor servo dengan manual.

Contoh program menggunakan perintah PULSOUT untuk mengirim pulsa yang menginstruksikan posisi motor servo. FOR…NEXT pengulangan digunakan untuk mengirim sebuah pulsa, sehingga menyebabkan motor servo untuk tetap menjaga suatu posisi dalam waktu tertentu.

Servotest.Bs akan membuat servo mulai pada jam 10:00 dan menjaga posisi itu sekitar tiga detik. Kemudian, program akan pindah menggerakkan motor servo searah jarum jam menuju jam 14:00 dan menjaga posisi itu sekitar tiga detik. Setelah itu, motor servo akan menjaga posisi pusat pada jam 12:00 sekitar tiga detik.



Gambar 2.18 *Tuas servo (kiri) menunjuk jam 10:00 ,(tengah) jam 14:00, (kanan) jam 12:00*

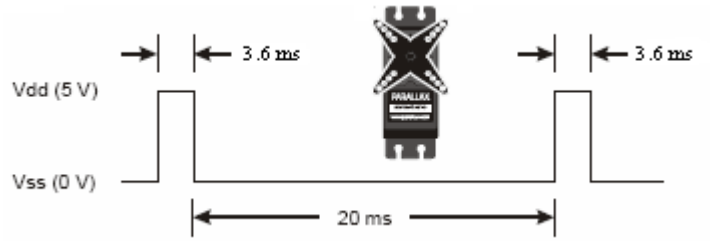
Untuk melakukan hal tersebut dapat menggunakan sebuah Servo Standard. Merek motor servo yang memiliki kekuatan berputar hanya pada jam 11:00 kemudian jam 13:00, atau mungkin untuk jam 9:00 dan kemudian 15:00. Ataupun bisa berputar kebalikan arah dan mulai searah jarum jam dengan sebelumnya berlawanan arah jarum jam. Tetapi jika servo dinilai untuk penggunaan dengan sebuah baterei 9 V, gerakan tuas motor servo mudah untuk diamati dan konsisten.

```
' {$STAMP BS2}
' {$PBASIC 2.5}
counter VAR Word
Jam 10:00
FOR counter = 1 TO 150
PULSOUT 2, 1800
PAUSE 20
NEXT
Jam 14:00
FOR counter = 1 TO 150
PULSOUT 2, 1000
PAUSE 20
NEXT
Jam 12:00
FOR counter = 1 TO 150
PULSOUT 2, 1585
PAUSE 20
NEXT
END
```

FOR…NEXT pengulangan yang pertama mengirim 150 pulsa, masing-masing yang mana di akhir 2.0 ms. Pulsa ini instruksikan motor servo untuk melanjutkan ke posisi jam 10:00 jika diamati di dalam terminologi suatu muka jam.

```
FOR counter = 1 TO 150
PULSOUT 2, 1000
PAUSE 20
NEXT
```

PULSOUT 2, 1800 (mengirimkan suatu pulsa yang berlangsung 1800 x 2µs. Menjadi 3600 µs atau 3,6 ms)



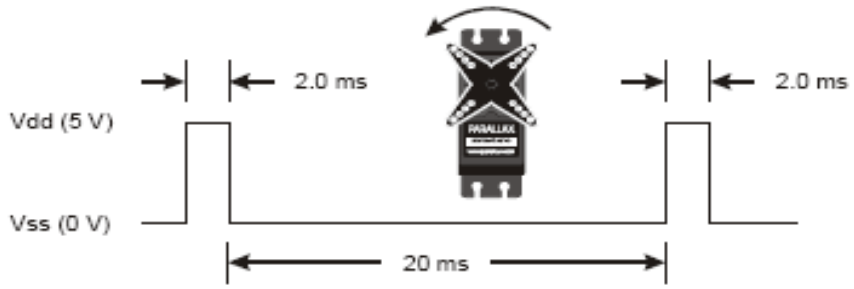
Gambar 2.19 *Pemilihan diagram waktu untuk 3.6 ms pulsa. Tiap-Tiap 20 ms servo memasuk posisi jam 10:00*

Gambar 2.19 disebut sebuah diagram pemilihan waktu. Itu menunjukkan suatu gambaran pulsa yang tinggi dan rendah dan berapa lama servo berlangsung. Pemilihan waktu pada diagram tidak menunjukkan banyaknya pulsa yang dikirimkan, tetapi itu memberi informasi tentang berapa lama pulsa yang tinggi dan rendah berlangsung. Masing-masing pulsa cukup untuk 2.0 ms dan dipisahkan oleh 20 ms waktu tunda jika sedang signal rendah.

FOR…NEXT pengulangan yang kedua mengirim 150 pulsa, tetapi waktu ini, masing-masing pulsa hanya berlangsung 1.0 ms. Ini akan menginstruksikan motor servo untuk membelokkan ke jam 14:00 dan akan menjaga posisi sekitar 2.40 detik.

```
FOR COUNTER = 1 TO 150
PULSOUT 2, 1000
PAUSE 20
NEXT
```

PULSOUT 2, 1000 (PULSOUT 2, 1000 mengirimkan suatu pulsa yang berlangsung 1000 x 2µs. Menjadi 2000 µs atau 2 ms)



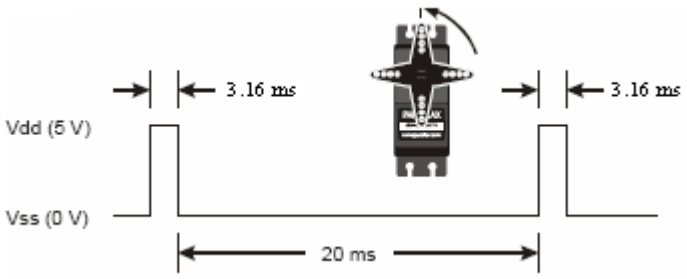
Gambar 2.20 *Pemilihan diagram waktu untuk 2.0 m pulsa. Tiap-Tiap 20 ms Servo masuk posisi jam 14:00*

FOR…NEXT pengulangan terakhir mengirim 150 pulsa, masing-masing yang terakhir 1.5 ms. Ini instruksikan servo untuk pergi ke posisi pusatnya (jam 12:00) untuk sekitar 3.23 detik.

```
FOR counter = 1 TO 150
PULSOUT 2, 1580
PAUSE 20
NEXT
```

PULSOUT 2, 1580 (PULSOUT 2, 1580 mengirimkan suatu pulsa yang berlangsung 1580 x 2µs. Menjadi 3160 µs atau 3.16 ms)

Gambar 2.21 menunjukkan pemilihan diagram waktu untuk pulsa ini. Dengan waktu tunda/waktu yang rendah masih 20 ms dan lama pulsa dikirim sekarang 1.5 ms.



Gambar 2.21 *Pemilihan diagram waktu untuk 3.16 m pulsa Tiap-Tiap 20 ms tuas servo ada pada posisi jam 12:00*

Pemilihan waktu dapat dilakukan juga dengan cara Matematis. Jika ingin mengkonversi waktu dari seperseribu detik untuk sebuah jangka waktu yang didapat menggunakan untuk PULSOUT, menggunakan persamaan ini:

Duration = number of ms x 1800 …….(2.1)

(Jangka waktu = jumlah m x 1800)

Sebagai contoh, jika tidak diketahui bahwa PULSOUT untuk 1.5 ms adalah 1000, dapat dikalkulasi sebagai berikut:

Duration = 1.5 x 1000

= 1500

Dapat juga menghitung jangka waktu dari perintah PULSOUT dengan menggunakan persamaan ini:

$$\text{number of ms} = \frac{duration}{500} ms \quad \text{.....(2.2)}$$

Sebagai contoh, lihat perintah PULSOUT 2, 1580, untuk mengerjakan pulsa itu yang terakhir adalah sebagai berikut:

$$\text{number of ms} = \frac{1580}{500} ms$$

= 3.16 ms

Putaran akan menyesuaikan posisi dan pegangan waktu masing-masing pengulangan FOR…NEXT. Pengulangan adalah apa yang dikendalikan dan berapa lama tuas motor servo tinggal di dalam posisi yang ditentukan. Nilai PULSOUT jangka waktu mengendalikan kapan tuas motor servo akan membelok. Itu penting untuk mengadakan percobaan dengan mengubah ini dapat menilai untuk memastikan bagaimana tuas motor servo bekerja sebelum bergerak ke dalam percobaan yang berikutnya. Menggunakan **Servotest.Bs2**, dapat

modifikasi semua FOR...NEXT pengulangan sedemikian sehingga servo melaksanakan sebanyak program yang asli:

FOR counter = 1 to 500

2.4. Matrik

2.4.1. Pengenalan Matrik

Notasi suatu matrik berukuran n x m ditulis dengan huruf besar dan dicetak tebal,misalnya **A**n×m. Huruf n menyatakan jumlah baris, dan huruf m jumlah kolom. Matrik terdiri dari elemen-elemen matrik yang dinyatakan dengan huruf kecil diikuti angka-angka indeks, misalnya a_{ij} , dimana i menunjukan posisi baris ke-i dan j menentukan posisi kolom ke-j.

$$\mathbf{A} = (a_{ij}) = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1m} \\ a_{21} & a_{22} & \dots & a_{2m} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nm} \end{bmatrix} \quad \text{.....(2.3)}$$

Contoh 1: Matrik $\mathbf{A}_{2\times3}$

$$\mathbf{A} = \begin{bmatrix} 3 & 8 & 5 \\ 6 & 4 & 7 \end{bmatrix}$$

dimana masing-masing elemennya adalah $a_{11} = 3$, $a_{12} = 8$, $a_{13} = 5$, $a_{21} = 6$,
$a_{22} = 4$, dan $a_{23} = 7$.

Contoh 2: Matrik $\mathbf{B}_{3\times2}$

$$\mathbf{B} = \begin{bmatrix} 1 & 3 \\ 5 & 9 \\ 2 & 4 \end{bmatrix}$$

dimana masing-masing elemennya adalah $b_{11} = 1$, $b_{12} = 3$, $b_{21} = 5$, $b_{22} = 9$, $b_{31} = 2$, dan $b_{32} = 4$.

2.4.2. Jenis-jenis matrik

Berdasarkan ordonya terdapat matrik sebagai berikut:

1) Matrik Bujursangkar

Matrik bujursangkar adalah matrik yang jumlah baris dan jumlah kolomnya sama.

Contoh 4: Matrik bujursangkar berukuran 3x3 atau sering juga disebut matrik bujursangkar orde 3

$$\mathbf{A} = \begin{bmatrix} 1 & 3 & 8 \\ 5 & 9 & 7 \\ 2 & 4 & 6 \end{bmatrix}$$

2) Matrik Simetrik

Matrik simetrik adalah matrik bujursangkar yang elemen-elemen matrik **A** bernilai sama dengan matrik transpose-nya ($\mathbf{A}^t$).

Contoh 5: Matrik simetrik

$$\mathbf{A} = \begin{bmatrix} 2 & -3 & 7 & 1 \\ -3 & 5 & 6 & -2 \\ 7 & 6 & 9 & 8 \\ 1 & -2 & 8 & 10 \end{bmatrix} \quad \mathbf{A}^t = \begin{bmatrix} 2 & -3 & 7 & 1 \\ -3 & 5 & 6 & -2 \\ 7 & 6 & 9 & 8 \\ 1 & -2 & 8 & 10 \end{bmatrix}$$

3) Matrik Diagonal

Matrik diagonal adalah matrik bujursangkar yang seluruh elemen-nya bernilai 0 (nol), kecuali elemen-elemen diagonalnya.

Contoh 6: Matrik diagonal orde 3

$$\mathbf{A} = \begin{bmatrix} 11 & 0 & 0 \\ 0 & 29 & 0 \\ 0 & 0 & 61 \end{bmatrix}$$

4) Matrik Diagonal Dominan

Matrik diagonal dominan adalah matrik bujursangkar yang memenuhi dimana i=1,2,3,..n.

Coba perhatikan matrik-matrik berikut ini

$$\mathbf{A} = \begin{bmatrix} 7 & 2 & 0 \\ 3 & 5 & -1 \\ 0 & 5 & -6 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 6 & 4 & -3 \\ 4 & -2 & 0 \\ -3 & 0 & 1 \end{bmatrix}$$

Pada elemen diagonal a_{ii} matrik **A**, $|7| > |2| + |0|$, lalu $|5| > |3| + |-1|$, dan $|-6| > |5| + |0|$. Maka matrik **A** disebut matrik diagonal dominan. Sekarang perhatikan elemen diagonal matrik **B**, $|6| < |4| + |-3|$, $|-2| < |4| + |0|$, dan $|1| < |-3| + |0|$. Dengan demikian, matrik **B** bukan matrik diagonal dominan.

5) Matrik Identitas

Matrik identitas adalah matrik bujursangkar yang semua elemen-nya bernilai 0 (nol), kecuali elemen-elemen diagonal yang seluruhnya bernilai 1.

Contoh 7: Matrik identitas orde 3

$$\mathbf{I} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{.....(2.4)}$$

6) Matrik *Upper-triangular*

Matrik upper-tringular adalah matrik bujursangkar yang seluruh elemen dibawah elemen diagonal bernilai 0 (nol).

Contoh 8: Matrik upper-triangular

$$\mathbf{A} = \begin{bmatrix} 3 & 6 & 2 & 1 \\ 0 & 4 & 1 & 5 \\ 0 & 0 & 8 & 7 \\ 0 & 0 & 0 & 9 \end{bmatrix}$$

7) Matrik *Lower-triangular*

Matrik lower-tringular adalah matrik bujursangkar yang seluruh elemen diatas elemen diagonal bernilai 0 (nol).

Contoh 9: Matrik lower-triangular

$$\mathbf{A} = \begin{bmatrix} 12 & 0 & 0 & 0 \\ 32 & -2 & 0 & 0 \\ 8 & 7 & 11 & 0 \\ -5 & 10 & 6 & 9 \end{bmatrix}$$

8) Matrik Tridiagonal

Matrik tridiagonal adalah matrik bujursangkar yang seluruh elemen bukan 0 (nol) berada disekitar elemen diagonal, sementara elemen lainnya bernilai 0 (nol).

Contoh 10: Matrik tridiagonal

$$\mathbf{A} = \begin{bmatrix} 3 & 6 & 0 & 0 \\ 2 & -4 & 1 & 0 \\ 0 & 5 & 8 & -7 \\ 0 & 0 & 3 & 9 \end{bmatrix}$$

9) Transpose Matrik

Operasi transpose terhadap suatu matrik akan menukar posisi kolom menjadi posisi baris demikian pula sebaliknya. Notasi matrik tranpose adalah $\mathbf{A}^T$ atau $\mathbf{A}_t$.

Contoh 11: Operasi transpose terhadap matrik **A**

$$\mathbf{A} = \begin{bmatrix} 3 & 8 & 5 \\ 6 & 4 & 7 \end{bmatrix} \mathbf{A}^T = \begin{bmatrix} 3 & 6 \\ 8 & 4 \\ 5 & 7 \end{bmatrix}$$

2.4.3. Operasi Matrik dan Sifat-sifatnya

1. Penjumlahan dan Pengurangan Dua Matrik

Pengertian penjumlahan matrik: jika A + C = D, maka elemen-elemen D diperoleh dari penjumlahan elemen-elemen A dan C yang seletak, yaitu: $d_{ij} = a_{ij} + c_{ij}$ untuk elemen D pada baris ke-i dan kolom ke-j. Akibatnya matrik A da C dapat dijumlahkan apabila kedua matrik memiliki ordo yang sama.

$$\mathbf{C} = \begin{bmatrix} 9 & 5 & 3 \\ 7 & 2 & 1 \end{bmatrix}$$

Dijumlahkan dengan matrik $\mathbf{A}_{2\times3}$, lalu hasilnya (misalnya) dinamakan matrik $\mathbf{D}_{2\times3}$

$$\mathbf{D} = \mathbf{A} + \mathbf{C}$$

$$\begin{aligned} \mathbf{D} &= \begin{bmatrix} 3 & 8 & 5 \\ 6 & 4 & 7 \end{bmatrix} + \begin{bmatrix} 9 & 5 & 3 \\ 7 & 2 & 1 \end{bmatrix} \\ &= \begin{bmatrix} 3+9 & 8+5 & 5+3 \\ 6+7 & 4+2 & 7+1 \end{bmatrix} \\ &= \begin{bmatrix} 12 & 13 & 8 \\ 13 & 6 & 8 \end{bmatrix} \end{aligned}$$

Tanpa mempedulikan nilai elemen-elemen masing-masing matrik, operasi penjumlahan antara matrik $\mathbf{A}_{2\times3}$ dan $\mathbf{C}_{2\times3}$, bisa juga dinyatakan dalam indeks masing-masing dari kedua matrik tersebut, yaitu:

$$\begin{bmatrix} d_{11} & d_{12} & d_{13} \\ d_{21} & d_{22} & d_{23} \end{bmatrix} = \begin{bmatrix} a_{11}+c_{11} & a_{12}+c_{12} & a_{13}+c_{13} \\ a_{21}+c_{21} & a_{22}+c_{22} & a_{23}+c_{23} \end{bmatrix} \text{.....(2.5)}$$

Sifat-sifat penjumlahan matrik:

a. $A + B = B + A$ (hukum komutatif untuk penjumlahan)

b. $A + (B + C) = (A + B) + C$ (hukum asosiatif untuk penjumlahan)

c. $A + 0 = 0 + A = A$

d. $(A + B)^T = A^T + B^T$

e. Ada matrik B yang sedemikian sehingga $A + B = B + A = 0$ yaitu:

$B = - A$

Pengertian pengurangan matrik: jika $A - B = C$, maka elemen-elemen C diperoleh dari pengurangan elemen-elemen A dan B yang seletak, yaitu : $c_{ij} = a_{ij} + b_{ij}$ atau pengurangan matrik ini dapat dipandang sebagai penjumlahan, yaitu: $A + (-B)$.

Contoh:

$$A = \begin{bmatrix} 5 & 4 \\ 6 & 9 \\ 7 & 0 \end{bmatrix}, B = \begin{bmatrix} 3 & 4 \\ 5 & 6 \\ 1 & 2 \end{bmatrix}$$

$$A - B = \begin{bmatrix} 5 & 4 \\ 6 & 9 \\ 7 & 0 \end{bmatrix} - \begin{bmatrix} 3 & 4 \\ 5 & 6 \\ 1 & 2 \end{bmatrix} = \begin{bmatrix} 2 & -2 \\ 1 & 5 \\ 6 & -2 \end{bmatrix}$$

$$\text{Atau } A - B = A + (-B) = \begin{bmatrix} 5 & 4 \\ 6 & 9 \\ 7 & 0 \end{bmatrix} + \begin{bmatrix} -3 & -4 \\ -5 & -6 \\ -1 & -2 \end{bmatrix} = \begin{bmatrix} 2 & -2 \\ 1 & 5 \\ 6 & -2 \end{bmatrix}$$

2. Perkalian Matrik Dengan bilangan Real (Skalar)

Matrik A dikalaikan dengan suatu bilangan real k maka kA diperoleh dari hasil kali setiap elemen A dengan k.

$$\text{Contoh: } P = \begin{bmatrix} 3 & 8 \\ 5 & 1 \end{bmatrix} \text{ maka } 4P = 4\begin{bmatrix} 3 & 8 \\ 5 & 1 \end{bmatrix} = \begin{bmatrix} 12 & 32 \\ 20 & 4 \end{bmatrix}$$

Jika a dan b bilngan real dan B, C matrik dengan ordo sedemikian hingga dapat dilakukan operasi hitung berikut; maka berlaku sifat-sifat perkalian matrik dengan skalar:

1) $a (B + C) = aB + Ca$

2) $a (B - C) = aB – Ca$

3) (a + b) C = aC + bC

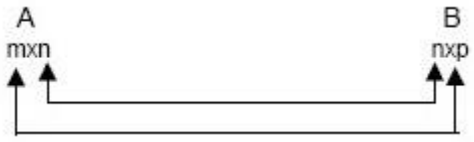
4) (a - b) C = aC - bC

5) (ab) C = a (C b)

6) $(aB)^T = aB^T$

3. Perkalian Dua Matrik

Dua matrik AB dapat dikalikan bila dan hanya bila jumlah kolom matrik a sama dengan jumlah baris matrik B.jadi $A_{mxn}B_{nxp}$ dapat didefinisikan tapi $A_{mxn}B_{nxp}$ tidak dapat didefinisikan.



$$= \underset{mxp}{AB} \quad \text{.....(2.6)}$$

Perhatikan bahwa hasil matrik AB berordo mxp.

Elemen-elemen AB diperoleh dari hasil kali setiap baris pada matrik A dengan setiap kolom pada matrik B, kemudian dijumlahkan pada satu elemen.

Untuk lebih jelasnya berikut contoh-contohnya:

- Perkalian matrik 1xp dengan matrik px1:

$$B = \begin{bmatrix} 6 & 8 & 7 \end{bmatrix} \text{ dan } C \begin{bmatrix} 4 \\ 7 \\ 2 \end{bmatrix}, \; B_{3x1}C_{3x1} = \begin{bmatrix} (6\times 4)+(8\times 7)+(7\times 2) \end{bmatrix} = \begin{bmatrix} 94 \end{bmatrix}$$

- Perkalian matrik px1 dengan matrik 1xp:

$$A = \begin{bmatrix} 2 \\ 5 \\ 4 \end{bmatrix} \text{ dan } B = \begin{bmatrix} 6 & 8 & 7 \end{bmatrix}, \; A_{3x1}B_{1x3} = \begin{bmatrix} 2x6 & 2x8 & 2x7 \\ 5x6 & 5x8 & 5x7 \\ 4x6 & 4x8 & 4x7 \end{bmatrix} = \begin{bmatrix} 12 & 16 & 14 \\ 30 & 40 & 35 \\ 24 & 32 & 28 \end{bmatrix}$$

Hasil kalinya merupakan suatu matrik berordo 3x3.

- Perkalian matrik mxn dengan matrik nxp:

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \text{ dan } B = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 2 & 0 \end{bmatrix}$$

$$A_{2x2}B_{2x3} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}\begin{bmatrix} 1 & 0 & 1 \\ 0 & 2 & 0 \end{bmatrix}$$

$$AB = \begin{bmatrix} (1x1)+(2x0) & (1x0)+(2x2) & (1x1)+(2x0) \\ (3x1)+(4x0) & (3x0)+(4x2) & (3x1)+(4x0) \end{bmatrix} = \begin{bmatrix} 1 & 4 & 1 \\ 3 & 8 & 3 \end{bmatrix}$$

Perhatikan hal-hal berikut:

1) Pada umumnya $AB \neq BA$ (tidak komutatif)
2) Apabila A matrik persegi maka A^2: A.A ; A^3: A^2.A ; A^4:A^3.A dan seterusnya.
3) Apabila AB = BC maka tidak dapat disimpulkan B = C (tidak berlaku sifat penghapusan)
4) Apabila AB = 0 maka tidak dapat disimpulkan A = 0 atau B = 0

Sifat-sifat perkalian matriks dengan matriks:

1. A(BC) = (AB)C
2. A (B + C) = AB + AC
3. (B + C)A = BA + CA
4. A (B - C) = AB – AC
5. (B - C)A = BA - CA
6. a(BC) = (aB)C = B(aC)
7. A1 = 1A = A

2.5 Transformasi Matrik

Untuk memindahkan suatu titik atau bangun pada sebuah bidang dapat dikerjakan dengan transformasi. Transformasi T pada suatu bidang memetakan tiap titik P pada bidang menjadi P’ pada bidang itu pula. Titik P’ disebut bayangan atau peta titik P.

Adapun transformasi-transformasi dasar pada dua dimensi adalah:

1) Transalasi (pergeseran)
2) Rotasi (putaran) pada titik origin

3) Skala (perubahan skala ukuran)

4) Refleksi (pencerminan) terhadap sumbu x atau sumbu y

5) Shear (skala proporsional) terhadap sumbu x atau sumbu y

Salah satu perbedaan transformasi dasar pada ruang tiga dimensi dengan ruang dua dimensi adalah membedakan rotasi terhadap masing-masing sumbu.

Matrik transformasi adalah matrik yang membuat sebuah obyek mengalami perubahan baik berupa perubahan posisi, maupun perubahan ukuran. Matrik transformasi 2D dinyatakan dalam ukuran 3x3, dimana kolom ke-3 digunakan untuk menyediakan tempat untuk proses translasi.

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \quad \text{.....(2.7)}$$

2.6. Invers Matrik Ordo 3x3

Misalkan matriks A adalah matriks persegi berodo 3 yang berbentuk

$$A = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}$$

Berdasarkan kaidah Sarrus, nilai determinan matriks A ditentukan oleh :

$$\text{Det } A = \begin{bmatrix} abc \\ def \\ ghi \end{bmatrix} = (aei + bfg + cdh) - (ceg + afh + bdi)$$

Penyelesaian Sistem Persamaan Liniear Tiga Variabel (Aturan Cramer)

$\begin{cases} ax + by + cz = p \\ dx + ey + fz = q \\ gx + hy + iz = r \end{cases}$ ditentukan oleh $\boxed{x = \frac{Dx}{D},\ y = \frac{Dy}{D},\ z = \frac{Dz}{D}}$ untuk $D \neq 0$, dengan

$$\boxed{D = \begin{vmatrix} a & b & c \\ d & e & f \\ g & h & i \end{vmatrix},\ Dx = \begin{vmatrix} p & b & c \\ q & e & f \\ r & h & i \end{vmatrix},\ Dy = \begin{vmatrix} a & p & c \\ d & q & f \\ g & r & i \end{vmatrix},\ Dz = \begin{vmatrix} a & b & p \\ d & e & q \\ g & h & r \end{vmatrix}}$$