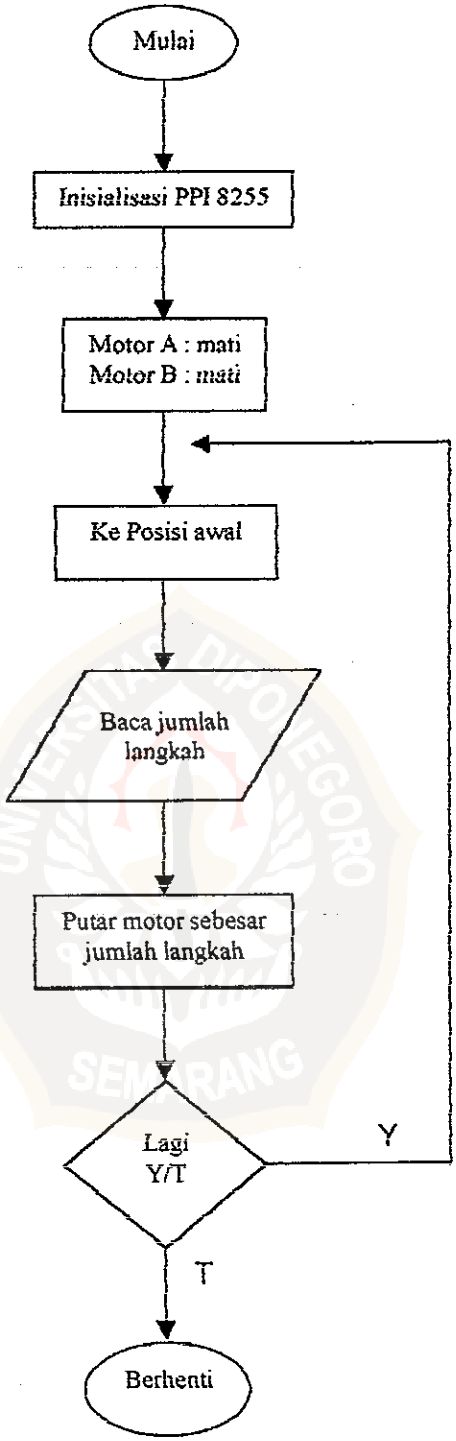
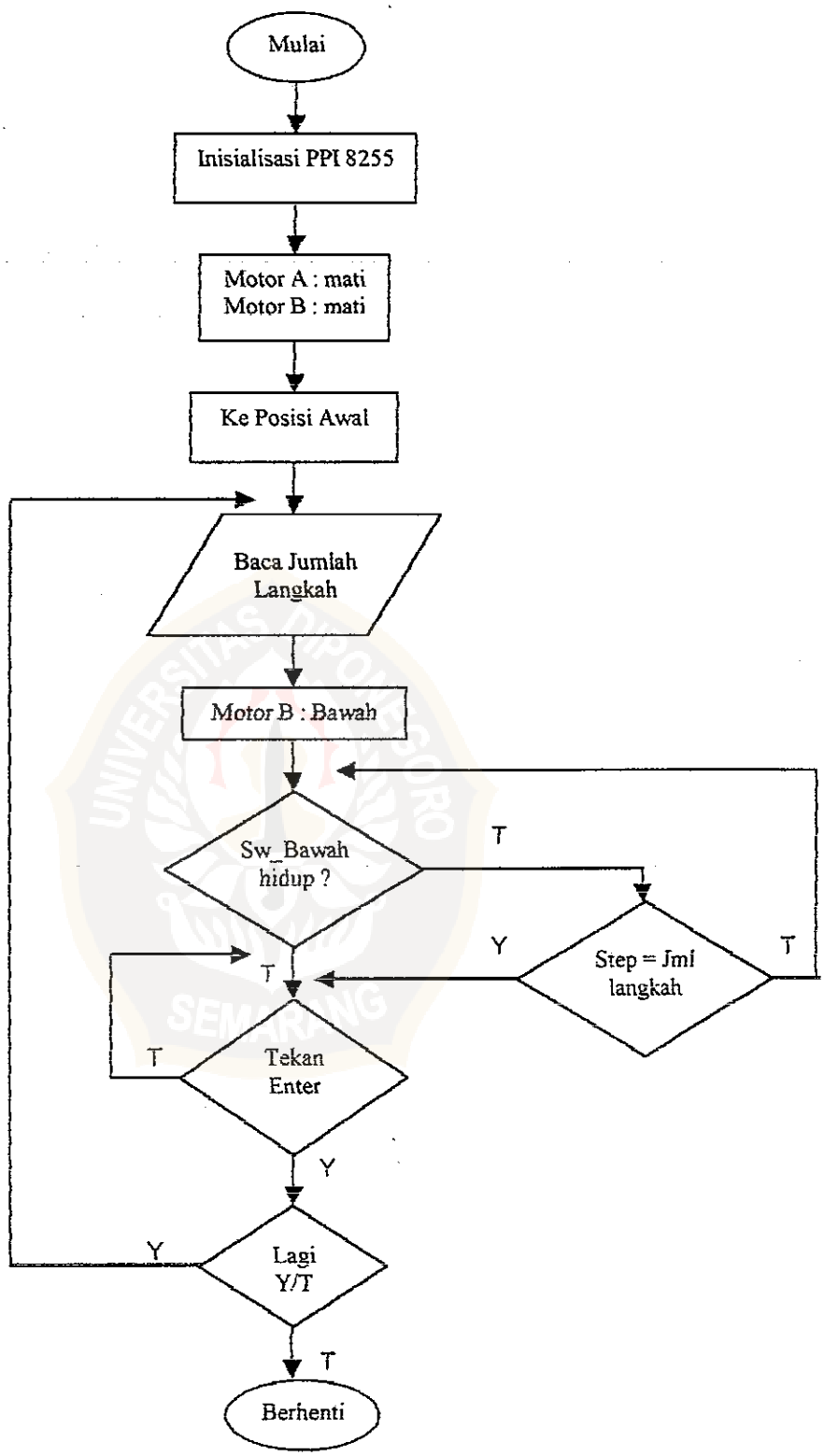
LAMPIRAN





```
{*************************************}
{  NAMA      : AGUS SUDARMANTO        }
{  NIM       : J2D096156              }
{  JURUSAN   : FISIKA                 }
{  FMIPA UNIVERSITAS DIPONEGORO       }
{            SEMARANG                 }
{*************************************}

PROGRAM TEStb2;
USES CRT,DOS;
CONST AA=$300;AB=$301;AC=$302;ACR=$303;
      JUDUL1 = '<<< LANGKAH VERSUS SUDUT >>>';

VAR DATAPC,DATA1,DATA2 : BYTE;
    TDB,TDA,POSISI,TD,I,I1 : INTEGER;
    SW_TABUNG,SW_AWAL,SENSOR_IR,SW_ATAS,SW_BAWAH : BYTE;
    FLGQUIT : BOOLEAN;
    L1,X1,X0,Y0,STEP : INTEGER;
    FLGOUT1 : BOOLEAN;
    PILIH : CHAR;


PROCEDURE OUT(AL:INTEGER;DATA : BYTE);
BEGIN
  PORT[AL] := DATA;
END;

PROCEDURE MOTPB_BAWAH(IA:INTEGER);
BEGIN
  CASE IA OF
  1 : OUT(AB,$03);
  2 : OUT(AB,$06);
  3 : OUT(AB,$0C);
  4 : OUT(AB,$09);
  END;
END;

PROCEDURE MOTPB_ATAS(IA:INTEGER);
BEGIN
  CASE IA OF
  1 : OUT(AB,$03);
  2 : OUT(AB,$09);
  3 : OUT(AB,$0C);
  4 : OUT(AB,$06);
  END;
END;
```

```
PROCEDURE MOTPA_KANAN(IA:INTEGER);
BEGIN
  CASE IA OF
  1 : OUT(AA,$03);
  2 : OUT(AA,$01);
  3 : OUT(AA,$09);
  4 : OUT(AA,$08);
  5 : OUT(AA,$0C);
  6 : OUT(AA,$04);
  7 : OUT(AA,$06);
  8 : OUT(AA,$02);
  END;
END;


PROCEDURE PUTKA(IX:INTEGER);
VAR IX1 : INTEGER;
BEGIN
    I := 0;
    IX1 := 0;
    REPEAT
      INC(I);
      INC(IX1);
      MOTPA_KANAN(I);DELAY(TDA);
      IF I=8 THEN I := 0;
    UNTIL (IX1=IX);
    OUT(AA,$00);
END;

PROCEDURE KE_ATAS;
BEGIN
  I := 0;
  REPEAT
    INC(I);
    MOTPB_ATAS(I);DELAY(tdb);
    SW_ATAS := (PORT[AC] AND $04) DIV $04;
    SW_BAWAH := (PORT[AC] AND $08) DIV $08;
    IF I=4 THEN I := 0;
  UNTIL SW_ATAS = 0;
  OUT(AB,$00);
END;


PROCEDURE KE_BAWAH;
BEGIN
  GOTOXY(X0+20,Y0+10);WRITE('BAWAH');
  I := 0;
  REPEAT
    INC(I);
    MOTPB_BAWAH(I);DELAY(tdb);
    SW_ATAS := (PORT[AC] AND $04) DIV $04;
    SW_BAWAH := (PORT[AC] AND $08) DIV $08;
    IF I=4 THEN I := 0;
  UNTIL SW_BAWAH = 0;
  OUT(AB,$00);
END;
```

```
PROCEDURE PUTAR_KE_POSISI_AWAL;
BEGIN
  I := 0;
  FLGOUT1 := FALSE;
  REPEAT
    INC(I);
    MOTPA_KANAN(I);DELAY(TDA);
    SENSOR_IR := PORT[AC] AND $01 ;
    SW_AWAL := (PORT[AC] AND $10) DIV $10;
    IF I=8 THEN I := 0;
    IF SW_AWAL = 1 THEN FLGOUT1 := TRUE
  UNTIL FLGOUT1;
  I := 0;
  FLGOUT1 := FALSE;
  REPEAT
    INC(I);
    MOTPA_KANAN(I);DELAY(TDA);
    SENSOR_IR := PORT[AC] AND $01 ;
    SW_AWAL := (PORT[AC] AND $10) DIV $10;
    IF I=8 THEN I := 0;
    IF SW_AWAL = 0 THEN FLGOUT1 := TRUE
  UNTIL FLGOUT1;
  I := 0;
  FLGOUT1 := FALSE;
  REPEAT
    INC(I);
    MOTPA_KANAN(I);DELAY(TDA);
    SW_AWAL := (PORT[AC] AND $10) DIV $10;
    SENSOR_IR := PORT[AC] AND $01 ;
    IF I=8 THEN I := 0;
    IF SENSOR_IR = 1 THEN FLGOUT1 := TRUE;
  UNTIL FLGOUT1;
  OUT(AA,$00);
END;


PROCEDURE PUTAR_1X;
BEGIN
  I := 0;
  REPEAT
    INC(I);
    MOTPA_KANAN(I);DELAY(TDA);
    SENSOR_IR := PORT[AC] AND $01 ;
    SW_AWAL := (PORT[AC] AND $10) DIV $10;
    IF I=8 THEN I := 0;
  UNTIL (SW_AWAL = 1) AND (SENSOR_IR=0);
  PUTKA(50);
  I := 0;
  FLGOUT1 := FALSE;
  REPEAT
    INC(I);
    MOTPA_KANAN(I);DELAY(TDA);
    SENSOR_IR := PORT[AC] AND $01 ;
        SW_AWAL := (PORT[AC] AND $10) DIV $10;
    IF I=8 THEN I := 0;
    IF SW_AWAL = 0 THEN
```

```
      BEGIN
        FLGOUT1 := TRUE;
        FLGQUIT := TRUE;
      END;
      IF SENSOR_IR = 1 THEN FLGOUT1 := TRUE;
    UNTIL FLGOUT1;
    OUT(AA,$00);
  END;


  PROCEDURE KE_POSISI_AWAL;
  BEGIN
    KE_ATAS;
    DELAY(100);
    PUTAR_KE_POSISI_AWAL;
  END;

  BEGIN
    X0 := 10;Y0 := 3;
    OUT(ACR,$89);TD := 100;
    TDA := 3; TDB := 2;
    REPEAT
      CLRSCR;
      GOTOXY(X0,Y0);WRITE('RESET KE POSISI AWAL .....');
      KE_POSISI_AWAL;
      CLRSCR;
      L1 := LENGTH(JUDUL1);
      X1 := (80 - L1) DIV 2;
      GOTOXY(X1,1);WRITE(JUDUL1);
      GOTOXY(X0,Y0);WRITE('MASUKKAN JUMLAH LANGKAH : ');
      READLN(STEP);
      PUTKA(STEP);
      REPEAT
        GOTOXY(X0,Y0+5);WRITE('LAGI (Y/T) : ');
        PILIH := UPCASE(READKEY);
      UNTIL PILIH IN['Y','T'];
    UNTIL (PILIH = 'T');
  END.
```

```
{****************************************}
{    NAMA      : AGUS SUDARMANTO         }
{    NIM       : J2D096156               }
{    JURUSAN   : FISIKA                  }
{    FMIPA UNIVERSITAS DIPONEGORO        }
{              SEMARANG                  }
{****************************************}

PROGRAM TESJARAK;
USES CRT,DOS;
CONST AA=$300;AB=$301;AC=$302;ACR=$303;
JUDUL1 = '<<< LANGKAH VERSUS JARAK >>>';

VAR DATAPC,DATA1,DATA2 : BYTE;
    TDB,TDA,POSISI,TD,I,I1 : INTEGER;
    SW_AWAL,SENSOR_IR,SW_ATAS,SW_BAWAH : BYTE;
    FLGQUIT : BOOLEAN;
    L1,X1,X0,Y0,STEP : INTEGER;
    FLGOUT1 : BOOLEAN;
    PILIH : CHAR;

PROCEDURE OUT(AL:INTEGER;DATA : BYTE);
BEGIN
  PORT[AL] := DATA;
END;

PROCEDURE MOTPB_BAWAH(IA:INTEGER);
BEGIN
  CASE IA OF
  1 : OUT(AB,$03);
  2 : OUT(AB,$06);
  3 : OUT(AB,$0C);
  4 : OUT(AB,$09);
  END;
END;

PROCEDURE MOTPB_ATAS(IA:INTEGER);
BEGIN
  CASE IA OF
  1 : OUT(AB,$03);
  2 : OUT(AB,$09);
  3 : OUT(AB,$0C);
  4 : OUT(AB,$06);
  END;
END;

PROCEDURE MOTPA_KANAN(IA:INTEGER);
BEGIN
  CASE IA OF
  1 : OUT(AA,$03);
  2 : OUT(AA,$01);
```

```
      3 : OUT(AA,$09);
      4 : OUT(AA,$08);
      5 : OUT(AA,$0C);
      6 : OUT(AA,$04);
      7 : OUT(AA,$06);
      8 : OUT(AA,$02);
     END;
   END;


   PROCEDURE PUTKA(IX:INTEGER);
   VAR IX1 : INTEGER;
   BEGIN
       I := 0;
       IX1 := 0;
       REPEAT
         INC(I);
         INC(IX1);
         MOTPA_KANAN(I);DELAY(TDA);
         IF I=8 THEN I := 0;
       UNTIL (IX1=IX);
       OUT(AA,$00);
   END;

   PROCEDURE KE_ATAS;
   BEGIN
     I := 0;
     REPEAT
       INC(I);
       MOTPB_ATAS(I);DELAY(TDB);
       SW_ATAS := (PORT[AC] AND $04) DIV $04;
       IF I=4 THEN I := 0;
     UNTIL SW_ATAS = 0;
     OUT(AB,$00);
   END;

   PROCEDURE KE_BAWAH;
   BEGIN
     I := 0;
     REPEAT
       INC(I);
       MOTPB_BAWAH(I);DELAY(TDB);
       SW_ATAS := (PORT[AC] AND $04) DIV $04;
       SW_BAWAH := (PORT[AC] AND $08) DIV $08;
       IF I=4 THEN I := 0;
     UNTIL SW_BAWAH = 0;
     OUT(AB,$00);
   END;

   PROCEDURE TPL_SW_BAWAH;
   VAR STR1 : STRING[3];
   BEGIN
     IF SW_BAWAH = 0 THEN STR1 := 'O N'
     ELSE STR1 := 'OFF';
     GOTOXY(X0,Y0+3);WRITE('SW_BAWAH     : ',STR1)
   END;
```

```
PROCEDURE TURUN(STEP1:INTEGER);
VAR I_STEP : INTEGER;
BEGIN
  I := 0;
  I_STEP := 0;
  REPEAT
    INC(I);INC(I_STEP);
    MOTPB_BAWAH(I);DELAY(TDB);
    SW_BAWAH := (PORT[AC] AND $08) DIV $08;
    GOTOXY(X0,Y0+2);WRITE('JML LANGKAH : ',I_STEP);
    TPL_SW_BAWAH;
    IF I=4 THEN I := 0;
  UNTIL (SW_BAWAH = 0) OR (I_STEP = STEP1);
  OUT(AB,$00);
END;


BEGIN
  X0 := 10;Y0 := 3;
  OUT(ACR,$89);TD := 100;
  TDA := 3; TDB := 2;
  REPEAT
    CLRSCR;
    GOTOXY(X0,Y0);WRITE('RESET KE POSISI AWAL .....');
    KE_ATAS;
    CLRSCR;
    L1 := LENGTH(JUDUL1);
    X1 := (80 - L1) DIV 2;
    GOTOXY(X1,1);WRITE(JUDUL1);
    GOTOXY(X0,Y0);WRITE('MASUKKAN JUMLAH LANGKAH : ');
    READLN(STEP);
    TURUN(STEP);
    REPEAT
      GOTOXY(X0,Y0+5);WRITE('LAGI (Y/T) : ');
      PILIH := UPCASE(READKEY);
    UNTIL PILIH IN['Y','T'];
  UNTIL (PILIH = 'T');
END.
```

```
{************************************}
{    NAMA    : AGUS SUDARMANTO       }
{    NIM     : J2D096156             }
{    JURUSAN : FISIKA                }
{    FMIPA UNIVERSITAS DIPONEGORO    }
{            SEMARANG                }
{************************************}


PROGRAM TEStb2;
USES CRT,DOS;
CONST AA=$300;AB=$301;AC=$302;ACR=$303;
JUDUL1='<          RANCANG BANGUN PENGENDALI MOTOR STEPPER          >';
JUDUL2='< UNTUK DETEKSI JUMLAH SAMPEL PUTAR DENGAN MENGGUNAKAN KOMPUTER >';


VAR DATAPC,DATA1,DATA2 : BYTE;
    JML_TABUNG,TDB,TDA,POSISI,TD,I,I1 : INTEGER;
    D : ARRAY[1..8] OF BYTE;
    PC0,SW_TABUNG,SW_AWAL,SENSOR_IR,SW_ATAS,SW_BAWAH : BYTE;
    FLGQUIT : BOOLEAN;
    X0,Y0 : INTEGER;
    FLGOUT1 : BOOLEAN;
    PILIH : CHAR;

PROCEDURE GARIS_DATAR(JML1 : INTEGER);
VAR IG1 : INTEGER;
BEGIN
  FOR  IG1 := 1 TO JML1 DO
  WRITE(CHR($C4));
END;

PROCEDURE GARIS_TEGAK(X,Y,JML1 : INTEGER);
VAR IG1 : INTEGER;
BEGIN
  FOR  IG1 := 1 TO JML1 DO
  BEGIN
    GOTOXY(X,Y+IG1);
    WRITE(CHR($C4));
  END;
END;

PROCEDURE TPL_JUDUL;
VAR L1,L2 : INTEGER;
BEGIN
  L1 := LENGTH(JUDUL1);
  L2 := LENGTH(JUDUL2);
  GOTOXY((80-L1) DIV 2,1);WRITE(JUDUL1);
  GOTOXY((80-L2) DIV 2,2);WRITE(JUDUL2);
END;
```

```
PROCEDURE TAMPIL_AWAL;
BEGIN

  TEXTATTR := $0B;
  GOTOXY(X0,Y0);WRITE('POSISI TABUNG KE : ');

  TEXTATTR := $0F;
  gotoxy(X0-2,Y0+1);WRITE(CHR($DA));
  FOR I := 1 TO 28 DO WRITE(CHR($C4));WRITE(CHR($BF));
  gotoxy(X0-2,Y0+2);WRITE(CHR($B3));

  gotoxy(X0+27,Y0+2);WRITE(CHR($B3));

  TEXTATTR := $06;
  GOTOXY(X0,Y0+2);WRITE('SENSOR & MOTOR');
  TEXTATTR := $06;
  GOTOXY(X0+20,Y0+2);WRITE('STATUS');
  TEXTATTR := $0F;

  gotoxy(X0-2,Y0+3);WRITE(CHR($C3));
  FOR I := 1 TO 28 DO WRITE(CHR($C4));WRITE(CHR($B4));

  GOTOXY(X0+17,Y0+1);
  WRITE(CHR($C2));

  GOTOXY(X0+17,Y0+2);
  WRITE(CHR($B3));

  GOTOXY(X0+17,Y0+3);
  WRITE(CHR($C5));

  FOR I := 1 TO 7 DO
  BEGIN
    GOTOXY(X0-2,Y0+3+I);
    WRITE(CHR($B3));
    GOTOXY(X0+17,Y0+3+I);
    WRITE(CHR($B3));
    GOTOXY(X0+27,Y0+3+I);
    WRITE(CHR($B3));
  END;

  TEXTATTR := $0E;
  GOTOXY(X0,Y0+4);WRITE('POSISI AWAL');
  GOTOXY(X0,Y0+5);WRITE('POSISI TABUNG');
  GOTOXY(X0,Y0+6);WRITE('TABUNG');
  GOTOXY(X0,Y0+7);WRITE('BATAS ATAS');
  GOTOXY(X0,Y0+8);WRITE('BATAS BAWAH');
  GOTOXY(X0,Y0+9);WRITE('MOTOR - A');
  GOTOXY(X0,Y0+10);WRITE('MOTOR - B');
  TEXTATTR := $0F;

  GOTOXY(X0-2,Y0+11);WRITE(CHR($C0));
```

```
FOR I := 1 TO 28 DO
  BEGIN
    GOTOXY(X0-2+I,Y0+11);
    WRITE(CHR($C4));
  END;

  WRITE(CHR($D9));
  GOTOXY(X0+17,Y0+11);

  WRITE(CHR($C1));

  GOTOXY(X0-2,Y0+12);WRITE(CHR($DA));

  GOTOXY(X0-1,Y0+12);GARIS_DATAR(63);
  GOTOXY(X0+7,Y0+12);WRITE(CHR($C2));

  GOTOXY(X0+62,Y0+12);WRITE(CHR($BF));

  GOTOXY(X0-2,Y0+13);WRITE(CHR($B3));

  TEXTATTR := $02;
  GOTOXY(X0,Y0+13); WRITE('TABUNG');

  GOTOXY(X0+8,Y0+13);
  WRITE(' TB1    TB2    TB3    TB4    TB5    TB6    TB7    TB8');

  TEXTATTR := $0F;
  GOTOXY(X0+7,Y0+13);WRITE(CHR($B3));
  GOTOXY(X0+62,Y0+13);WRITE(CHR($B3));

  GOTOXY(X0-2,Y0+14);WRITE(CHR($C3));
  GOTOXY(X0-1,Y0+14);GARIS_DATAR(63);
  GOTOXY(X0+7,Y0+14);WRITE(CHR($C5));


  GOTOXY(X0+62,Y0+14);WRITE(CHR($B4));

  GOTOXY(X0-2,Y0+15);WRITE(CHR($B3));
  TEXTATTR := $05;
  GOTOXY(X0,Y0+15); WRITE('STATUS');
  TEXTATTR := $0F;

  GOTOXY(X0+7,Y0+15);WRITE(CHR($B3));
  GOTOXY(X0+62,Y0+15);WRITE(CHR($B3));

  GOTOXY(X0-2,Y0+16);WRITE(CHR($C0));
  GOTOXY(X0-1,Y0+16);GARIS_DATAR(63);
  GOTOXY(X0+7,Y0+16);WRITE(CHR($C1));
  GOTOXY(X0+62,Y0+16);WRITE(CHR($D9));

  TEXTATTR := $09;
  GOTOXY(X0,Y0+18); WRITE('JUMLAH TABUNG : ');
  TEXTATTR := $0F;
END;
```

```
FUNCTION TPL_ON_OFF(STS1:BYTE):STRING;
VAR STRSTS : STRING[3];
BEGIN
  IF STS1 = 0 THEN STRSTS := 'O N'
  ELSE STRSTS := 'OFF';
  TPL_ON_OFF := STRSTS;
END;

PROCEDURE TAMPIL_SW_AWAL;
BEGIN
  GOTOXY(X0+20,Y0+4);
  IF SW_AWAL = 0 THEN
  BEGIN
    TEXTATTR := $0A;
    WRITE('O N');
  END
  ELSE
  BEGIN
    TEXTATTR := $0C;
    WRITE('OFF');
  END;
  TEXTATTR := $0F;
END;

PROCEDURE TAMPIL_SENSOR_IR;
BEGIN
  GOTOXY(X0+20,Y0+5);
  IF SENSOR_IR = 1 THEN
  BEGIN
    TEXTATTR := $0A;
    WRITE('O N');
  END
  ELSE
  BEGIN
    TEXTATTR := $0C;
    WRITE('OFF');
  END;
  TEXTATTR := $0F;
END;

PROCEDURE TAMPIL_SW_TABUNG;
BEGIN
  GOTOXY(X0+20,Y0+6);
  IF SW_TABUNG = 0 THEN
  BEGIN
    TEXTATTR := $0A;
    WRITE('O N');
  END
  ELSE
  BEGIN
    TEXTATTR := $0C;
    WRITE('OFF');
  END;
    TEXTATTR := $0F;
END;
```

```
PROCEDURE TAMPIL_SW_ATAS;
BEGIN
  GOTOXY(X0+20,Y0+7);
  IF SENSOR_IR = 0 THEN
  BEGIN

    TEXTATTR := $0A;
    WRITE('O N');
  END
  ELSE
  BEGIN
    TEXTATTR := $0C;
    WRITE('OFF');
  END;
  TEXTATTR := $0F;
END;

PROCEDURE TAMPIL_SW_BAWAH;
BEGIN
  GOTOXY(X0+20,Y0+8);
  IF SENSOR_IR = 0 THEN
  BEGIN
    TEXTATTR := $0A;
    WRITE('O N');
  END
  ELSE
  BEGIN
    TEXTATTR := $0C;
    WRITE('OFF');
  END;
  TEXTATTR := $0F;
END;


PROCEDURE OUT(AL:INTEGER;DATA : BYTE);
BEGIN
  PORT[AL] := DATA;
END;


PROCEDURE MOTPB_BAWAH(IA:INTEGER);
BEGIN
  CASE IA OF
  1 : OUT(AB,$03);
  2 : OUT(AB,$06);
  3 : OUT(AB,$0C);
  4 : OUT(AB,$09);
  END;
END;


PROCEDURE MOTPB_ATAS(IA:INTEGER);
BEGIN
  CASE IA OF
  1 : OUT(AB,$03);
  2 : OUT(AB,$09);
```

```
  3 : OUT(AB,$0C);
  4 : OUT(AB,$06);
  END;
END;

PROCEDURE MOTPA_KANAN(IA:INTEGER);
BEGIN
  CASE IA OF
  1 : OUT(AA,$03);
  2 : OUT(AA,$01);
  3 : OUT(AA,$09);
  4 : OUT(AA,$08);
  5 : OUT(AA,$0C);
  6 : OUT(AA,$04);
  7 : OUT(AA,$06);
  8 : OUT(AA,$02);
  END;
END;


PROCEDURE TAMPIL(ALX:INTEGER;DATAX:BYTE);
BEGIN
  D[1] :=  DATAX AND $01;
  D[2] := (DATAX AND $02) DIV $02;
  D[3] := (DATAX AND $04) DIV $04;
  D[4] := (DATAX AND $08) DIV $08;
  D[5] := (DATAX AND $10) DIV $10;
  D[6] := (DATAX AND $20) DIV $20;
  D[7] := (DATAX AND $40) DIV $40;
  D[8] := (DATAX AND $80) DIV $80;
  IF ALX = AA THEN
  BEGIN
    GOTOXY(4,4);
    WRITE('PA7  PA6  PA5  PA4  PA3  PA2  PA1  PA0');
    FOR I1 := 1 TO 8 DO
      BEGIN
      GOTOXY(5*I1,5);WRITE(D[9-I1]);
    END;
  END;
  IF ALX = AB THEN
  BEGIN
    GOTOXY(4,7);
    WRITE('PB7  PB6  PB5  PB4  PB3  PB2  PB1  PB0');
    FOR I1 := 1 TO 8 DO
    BEGIN
      GOTOXY(5*I1,8);WRITE(D[9-I1]);
    END;
  END;
  IF ALX = AC THEN
  BEGIN
    GOTOXY(4,10);
    WRITE('PC7  PC6  PC5  PC4  PC3  PC2  PC1  PC0');
    FOR I1 := 1 TO 8 DO
    BEGIN
      GOTOXY(5*I1,11);WRITE(D[9-I1]);
```

```
      END;
  END;
END;

PROCEDURE PUTKA(IX:INTEGER);
VAR IX1 : INTEGER;
BEGIN
    I := 0;
    IX1 := 0;
    REPEAT
      INC(I);
      INC(IX1);
      MOTPA_KANAN(I);DELAY(TDA);
      SW_AWAL := (PORT[AC] AND $10) DIV $10;
      TAMPIL_SW_AWAL;
      IF I=8 THEN I := 0;
    UNTIL (SW_AWAL=0) OR (IX1=IX);
    OUT(AA,$00);
END;

PROCEDURE KE_ATAS;
BEGIN
  TEXTATTR := $0A;
  GOTOXY(X0+20,Y0+10);WRITE('ATAS ');
  TEXTATTR := $0F;
  I := 0;
  REPEAT
    INC(I);
    MOTPB_ATAS(I);DELAY(TDB);
    SW_ATAS := (PORT[AC] AND $04) DIV $04;
    SW_BAWAH := (PORT[AC] AND $08) DIV $08;
    TAMPIL_SW_ATAS;
    TAMPIL_SW_BAWAH;
    IF I=4 THEN I := 0;
  UNTIL SW_ATAS = 0;
  OUT(AB,$00);
  TEXTATTR := $0C;
  GOTOXY(X0+20,Y0+10);WRITE('STOP ');
  TEXTATTR := $0F;
END;

PROCEDURE KE_BAWAH;
BEGIN
  TEXTATTR := $02;
  GOTOXY(X0+20,Y0+10);WRITE('BAWAH');
  TEXTATTR := $0F;
  I := 0;
  REPEAT
    INC(I);
    MOTPB_BAWAH(I);DELAY(TDB);
    SW_ATAS := (PORT[AC] AND $04) DIV $04;
    SW_BAWAH := (PORT[AC] AND $08) DIV $08;
    TAMPIL_SW_ATAS;
    TAMPIL_SW_BAWAH;
    IF I=4 THEN I := 0;
  UNTIL SW_BAWAH = 0;
```

```
  TEXTATTR := $0C;
  GOTOXY(X0+20,Y0+10);WRITE('STOP ');
  TEXTATTR := $0F;
  OUT(AB,$00);
END;

PROCEDURE PUTAR_KE_POSISI_AWAL;
BEGIN
  TEXTATTR := $0A;
  GOTOXY(X0+20,Y0+9);WRITE('PUTAR');
  TEXTATTR := $0F;
  I := 0;
  FLGOUT1 := FALSE;
  REPEAT
    INC(I);
    MOTPA_KANAN(I);DELAY(TDA);
    SENSOR_IR := PORT[AC] AND $01 ;
    SW_AWAL := (PORT[AC] AND $10) DIV $10;
    TAMPIL_SENSOR_IR;
    TAMPIL_SW_AWAL;
    IF I=8 THEN I := 0;
    IF SW_AWAL = 1 THEN FLGOUT1 := TRUE
  UNTIL FLGOUT1;

  I := 0;
  FLGOUT1 := FALSE;
  REPEAT
    INC(I);
    MOTPA_KANAN(I);DELAY(TDA);
    SENSOR_IR := PORT[AC] AND $01 ;
    SW_AWAL := (PORT[AC] AND $10) DIV $10;
    TAMPIL_SENSOR_IR;
    TAMPIL_SW_AWAL;
    IF I=8 THEN I := 0;
    IF SW_AWAL = 0 THEN FLGOUT1 := TRUE
  UNTIL FLGOUT1;
  I := 0;
  FLGOUT1 := FALSE;
  REPEAT
    INC(I);
    MOTPA_KANAN(I);DELAY(TDA);
    SW_AWAL := (PORT[AC] AND $10) DIV $10;
    SENSOR_IR := PORT[AC] AND $01 ;
    TAMPIL_SENSOR_IR;
    TAMPIL_SW_AWAL;
    IF I=8 THEN I := 0;
    IF SENSOR_IR = 1 THEN FLGOUT1 := TRUE;
  UNTIL FLGOUT1;
  OUT(AA,$00);
  TEXTATTR := $0C;
  GOTOXY(X0+20,Y0+9);WRITE('STOP ');
  TEXTATTR := $0F;
END;
```

```
PROCEDURE PUTAR_1X;
BEGIN
  TEXTATTR := $0A;
  GOTOXY(X0+20,Y0+9);WRITE('PUTAR');
  TEXTATTR := $0F;
  I := 0;
  REPEAT
    INC(I);
    MOTPA_KANAN(I);DELAY(TDA);
    SENSOR_IR := PORT[AC] AND $01 ;
    SW_AWAL := (PORT[AC] AND $10) DIV $10;
    TAMPIL_SENSOR_IR;
    TAMPIL_SW_AWAL;
    IF I=8 THEN I := 0;
  UNTIL (SW_AWAL = 1) AND (SENSOR_IR=0);

  PUTKA(50);
  I := 0;
  FLGOUT1 := FALSE;
  REPEAT
    INC(I);
    MOTPA_KANAN(I);DELAY(TDA);
    SENSOR_IR := PORT[AC] AND $01 ;
    SW_AWAL := (PORT[AC] AND $10) DIV $10;
    TAMPIL_SENSOR_IR;
    TAMPIL_SW_AWAL;
    IF I=8 THEN I := 0;
    IF SW_AWAL = 0 THEN
    BEGIN
      FLGOUT1 := TRUE;
      FLGQUIT := TRUE;
    END;
    IF SENSOR_IR = 1 THEN FLGOUT1 := TRUE;
  UNTIL FLGOUT1;
  OUT(AA,$00);
  TEXTATTR := $0C;
  GOTOXY(X0+20,Y0+9);WRITE('STOP ');
  TEXTATTR := $0F;
END;


PROCEDURE KE_POSISI_AWAL;
BEGIN
  KE_ATAS;
  DELAY(100);
  PUTAR_KE_POSISI_AWAL;
END;

PROCEDURE BACA_SW_TABUNG;
BEGIN
  SW_TABUNG := (PORT[AC] AND $02) DIV $02;
END;
```

```
PROCEDURE CEK_TABUNG;
BEGIN
  SW_TABUNG := (PORT[AC] AND $02) DIV $02;
  GOTOXY(X0+10+7*(POSISI-1),Y0+15);
  IF SW_TABUNG = 0 THEN
  BEGIN
    INC(JML_TABUNG);
    WRITE('X');
  END
  ELSE
    WRITE('0');
  GOTOXY(X0+16,Y0+18); WRITE(JML_TABUNG:1);
END;

BEGIN
  OUT(ACR,$89);TD := 100;
  TDA := 4; TDB := 2;
  X0 := 10;Y0 := 4;
  REPEAT
    FLGQUIT := FALSE;
    CLRSCR;
    TPL_JUDUL;
    TAMPIL_AWAL;
    BACA_SW_TABUNG;
    TAMPIL_SW_TABUNG;
    JML_TABUNG := 0;
    GOTOXY(X0+16,Y0+18); WRITE(JML_TABUNG:1);
    KE_POSISI_AWAL;
    POSISI := 1;
    REPEAT
      GOTOXY(X0+19,Y0);WRITE(POSISI);
      BACA_SW_TABUNG;
      TAMPIL_SW_TABUNG;
      KE_BAWAH;
      DELAY(1000);
      BACA_SW_TABUNG;
      TAMPIL_SW_TABUNG;
      CEK_TABUNG;
      KE_ATAS;
      BACA_SW_TABUNG;
      TAMPIL_SW_TABUNG;
      PUTAR_1X;
      INC(POSISI);
    UNTIL  FLGQUIT;
    REPEAT
      GOTOXY(X0,23);WRITE('LAGI (Y/T) : ');
      PILIH := UPCASE(READKEY);
    UNTIL PILIH IN['Y','T'];
  UNTIL (PILIH = 'T');
END.
```

Hasil deteksi jumlah obyek untuk obyek berjumlah 5

< RANCANG BANGUN PENGENDALI MOTOR STEPPER UNTUK >
< DETEKSI JUMLAH OBYEK DENGAN MENGGUNAKAN KOMPUTER >

POSISI TABUNG KE : 8

SENSOR & MOTOR	STATUS
POSISI AWAL	OFF
POSISI TABUNG	ON
TABUNG	OFF
BATAS ATAS	OFF
BATAS BAWAH	OFF
MOTOR - A	STOP
MOTOR - B	ATAS

TABUNG	TB1	TB2	TB3	TB4	TB5	TB6	TB7	TB8
STATUS	1	1	1	0	0	1	1	0

JUMLAH TABUNG : 5

KETERANGAN :
1 = ADA TABUNG
0 = TIDAK ADA TABUNG

Hasil deteksi jumlah obyek untuk obyek berjumlah 2

< RANCANG BANGUN PENGENDALI MOTOR STEPPER UNTUK >
< DETEKSI JUMLAH OBYEK DENGAN MENGGUNAKAN KOMPUTER >

POSISI TABUNG KE : 8

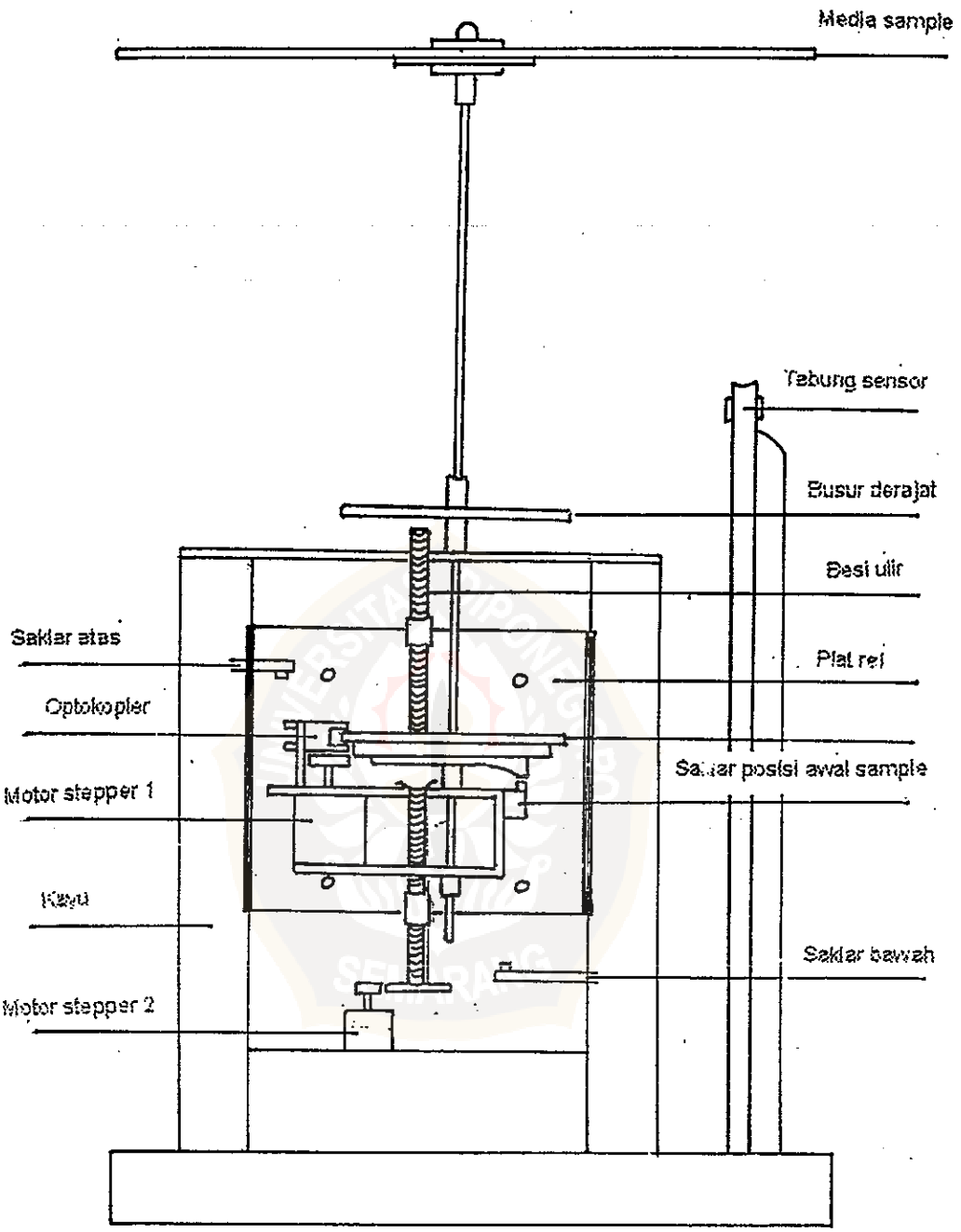
SENSOR & MOTOR	STATUS
POSISI AWAL	OFF
POSISI TABUNG	ON
TABUNG	OFF
BATAS ATAS	OFF
BATAS BAWAH	OFF
MOTOR - A	STOP
MOTOR - B	ATAS

TABUNG	TB1	TB2	TB3	TB4	TB5	TB6	TB7	TB8
STATUS	1	0	0	0	0	1	0	0

JUMLAH TABUNG : 2

KETERANGAN :
1 = ADA TABUNG
0 = TIDAK ADA TABUNG

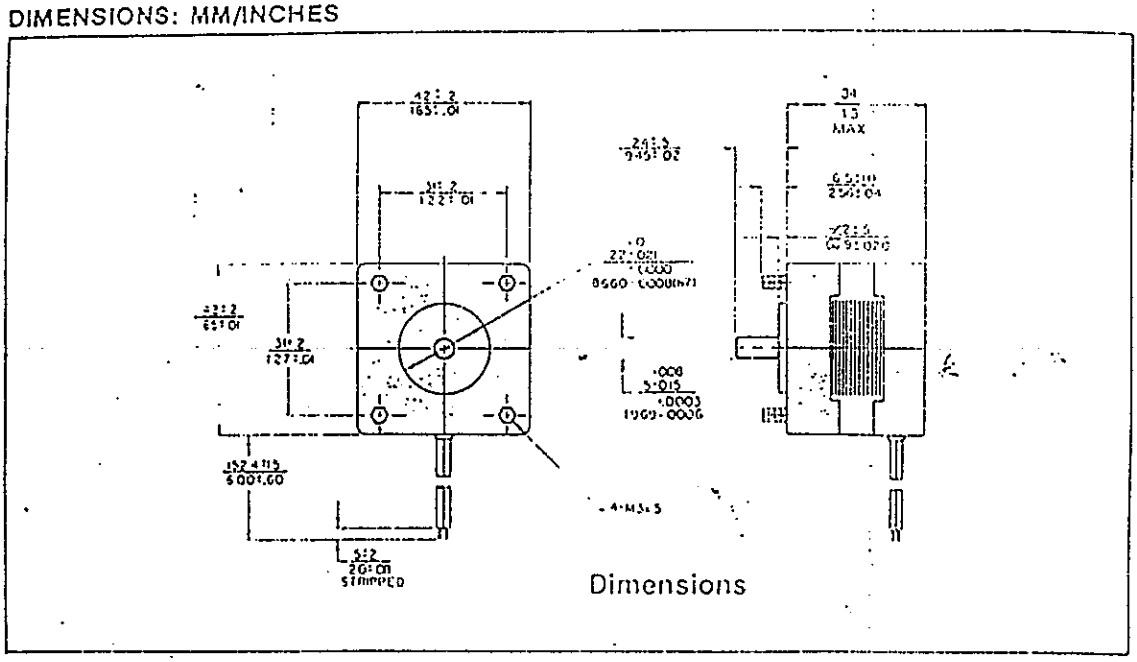
Lampiran 7



Gambar mekanik secara keseluruhan

Lampiran 8 Data sheet motor langkah

DIMENSIONS: MM/INCHES



Dimensions

Specifications	Ordering Part Number 4SO – 120834S
DC Operating Voltage	12
Res. per Winding Ω	74
Ind. per Winding mH	26
Holding Torque mNm/oz-in.	65/9.2
Rotor Moment of Inertia g·m²	1.9×10^{-3}
Detent Torque mNm/oz-in.	8.5/1.2
Step Angle	1.8°
Step Angle Tolerance	± 5%
Steps per Rev.	200
Max. Radial Load* Kg/Lbs	4/8.8
Max. Axial Load Kg/Lbs	8/17.6
Max. Temp. Rise	55°C
Ambient Temp. Range Operating Storage	 – 20°C to + 50°C – 20°C to + 60°C
Bearing Type	Ball, Double Shielded
Insulation Res. at 500 Vdc	50 MΩ
DieLectric Withstanding Voltage	500 Vac for 60 Sec.
Weight g/oz	195/7

Note: (1) Unless otherwise indicated all values shown are typical
(2) Other windings available on special order
(3) Consult factory for availability of motors with .9°, 3.6°, and 7.5° step angle
*Measured 10mm from mounting plate surface

CATALOG PN CONSTRUCTION

Example 1. Single Shaft Extension

4SO — 060 BA 34 S
- S — Single Shaft Extension
- 34 — 34mm/1.3 in length
- BA — Aluminum End Bells
- 060 — 6 volts
- 4SO — Basic Motor

Example 2. Double Shaft Extension

4SO — 120 BF 34 W
- W — Double Shaft Extension
- 34 — 34mm/1.3 in length
- BF — Steel End Bells
- 120 — 12 volts
- 4SO — Basic Motor