BAB IV

METODE KERJA PRAKTEK

Pada bab empat menjelaskan tentang metode dalam pengerjaan kerja praktek yang disertai dengan cara-cara pembuatan sistem kontrol dan pemrograman robot menggunakan *Code Vision* AVR (CVAVR).

Metode yang digunakan dalam pengerjaan kerja praktek ini adalah sebagai berikut :

1. Wawancara, yaitu bertanya secara langsung kepada asisten laboratorium mikrokontroler untuk mendapatkan data-data serta informasi yang berhubungan dengan kerja praktek.
2. Studi literatur, yaitu dengan mempelajari dan membaca buku, maupun literatur lainnya yang berkaitan dengan objek kerja praktek termasuk permasalahan yang dihadapi.

Berikut merupakan cara-cara pembuatan sistem kontrol dengan menggunakan *Orcad 9.23* dan pemrograman menggunakan CVAVR.

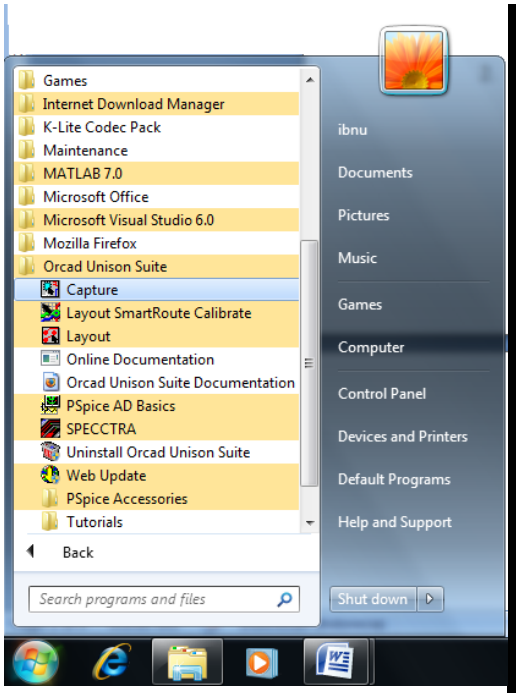
4.1 Pembuatan Minimum Sistem

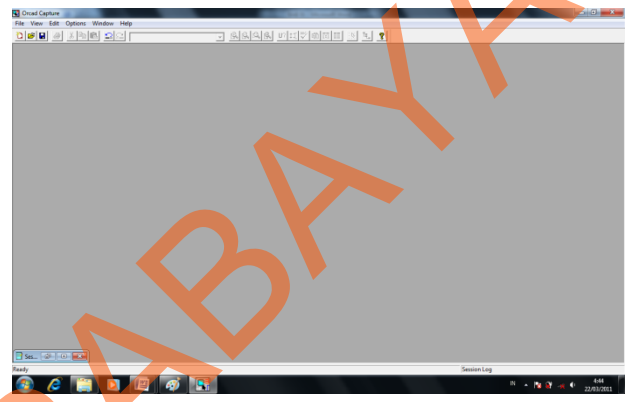
Program *Orcad 9.23* merupakan *software* yang digunakan penulis sebagai, pembuatan *schematic* rangkaian elektronika beserta cara pembuatan *layout Printed Circuit Board* (PCB). Berikut langkah-langkah untuk menjalankan *Orcad 9.23* :

1. Jalankan program *Orcad 9.23* yang ada pada komputer mulai dari

- Start windows ➔ pilih *all program*

- Setelah *all program* ➔ pilih *Orcad unison suite* ➔ pilih *capture* (digunakan untuk membuat *schematic* elektronik) seperti yang tampak pada Gambar 4.1.

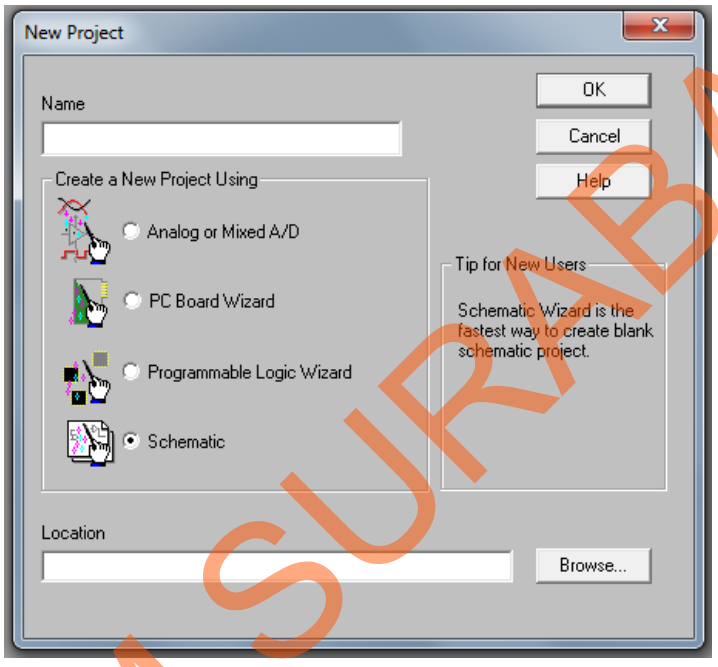




Gambar 4.1 Cara membuka program *Orcad 9.23* dan jendela awal *capture* pada *Orcad 9.23*

2. Persiapan untuk membuat *project* baru, langkah-langkahnya sebagai berikut:

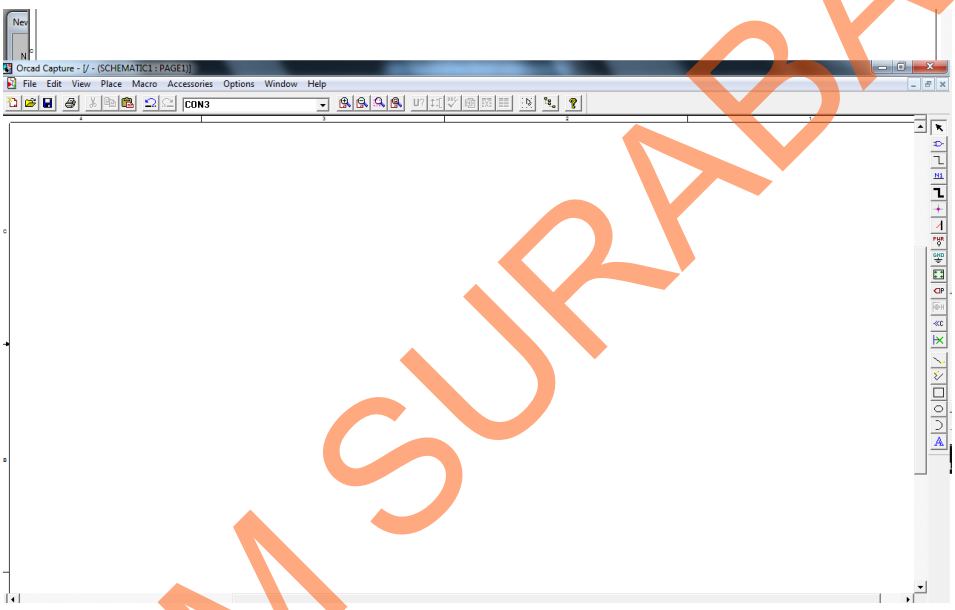
    - Klik menu *file*, kemudian pilih *new* ➔ *project*. Sehingga akan tampil seperti pada Gambar 4.2.



Gambar 4.2 Dialog *new file project*

- Pilih *schematic,* ketik nama *file,* dan pilih *location* untuk penyimpanan *file*.

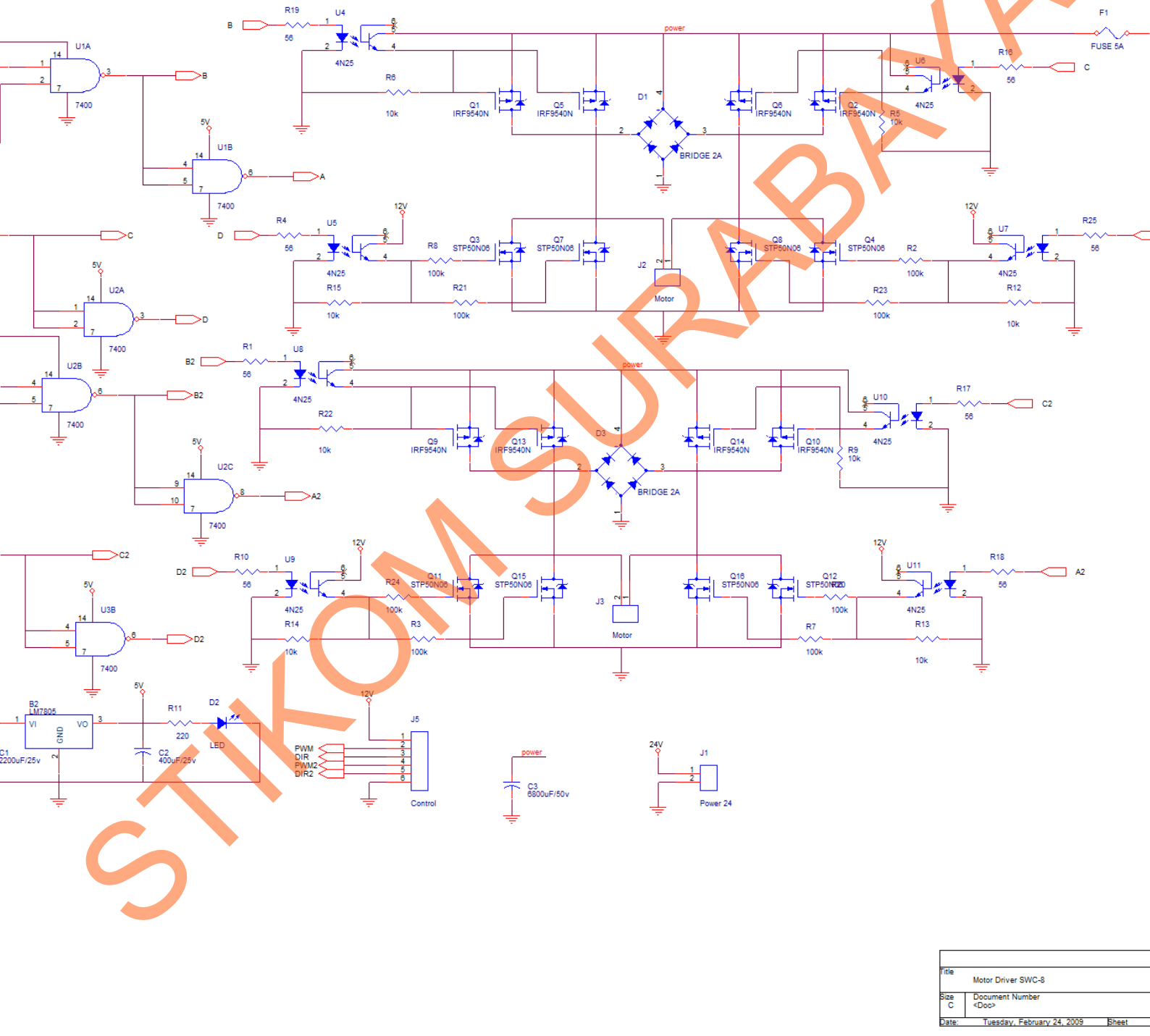
- Setelah itu klik OK, sehingga akan tampil seperti gambar 4.3



Gambar 4.3 Lembar kerja *Orcad 9.23*

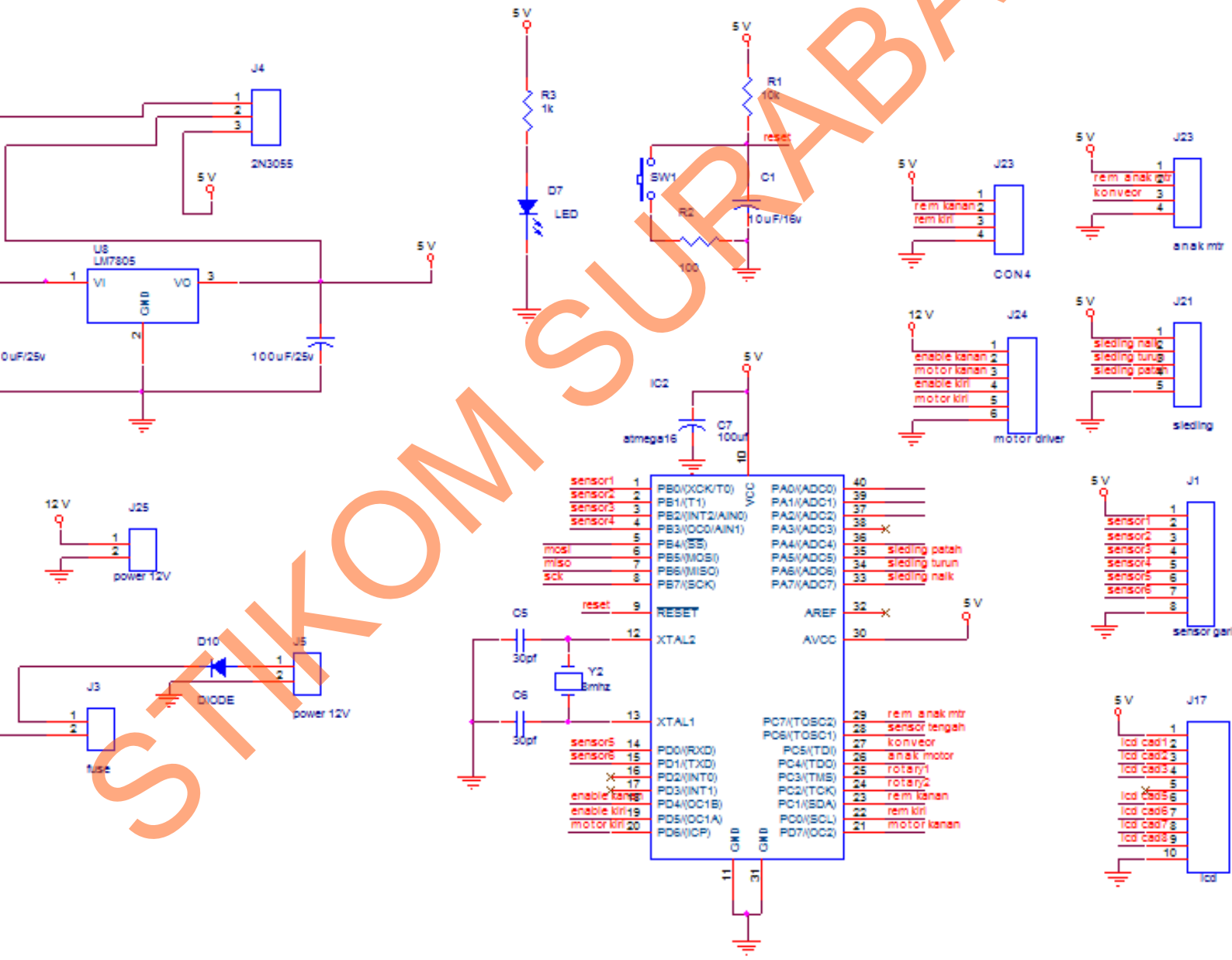
3. Siap untuk membuat sistem kontrol pada robot.

4. Sebagai berikut merupakan rangkaian sistem kontrol pada robot mankara.

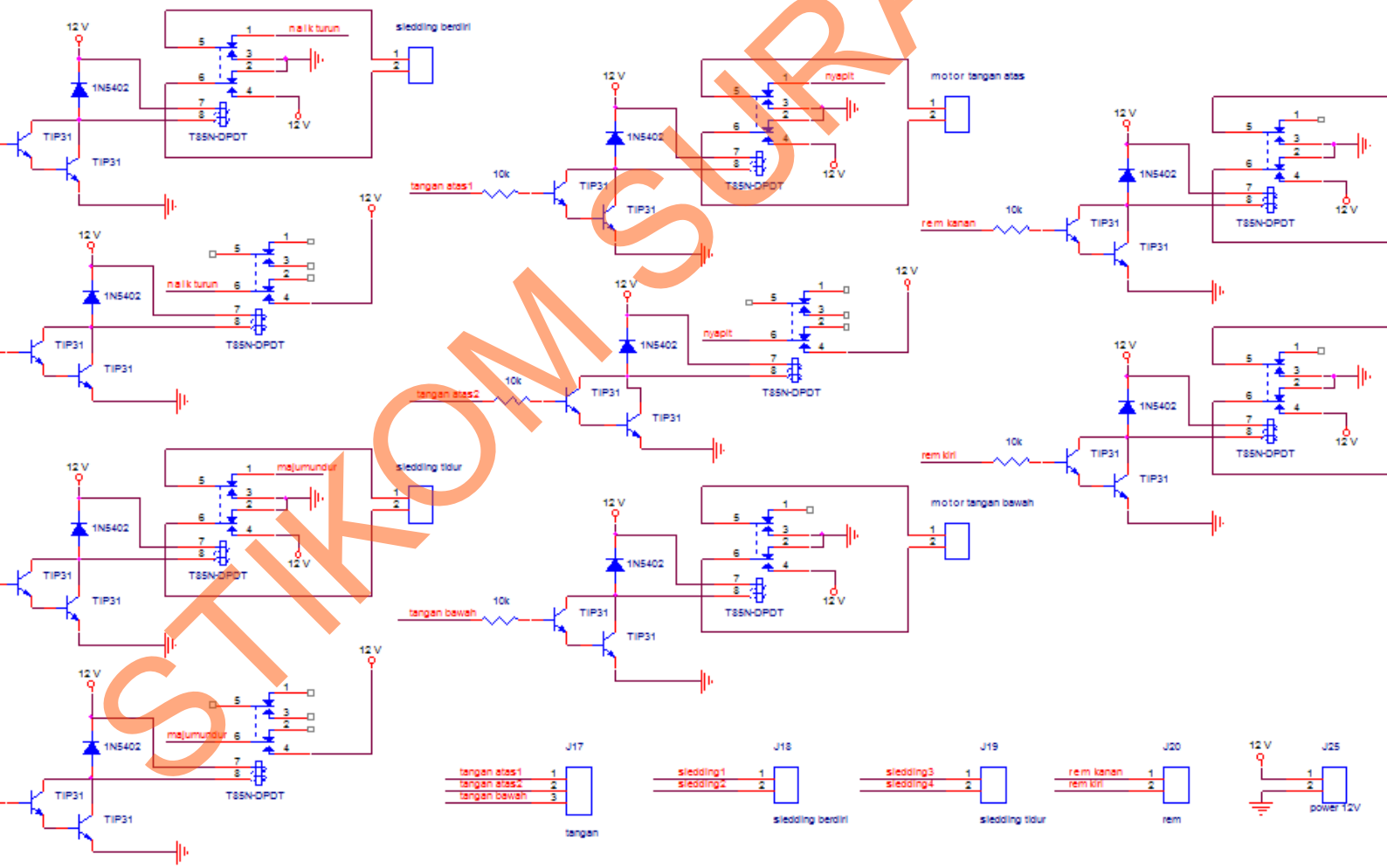


Title	Motor Driver SWC-8
Size C	Document Number <Doc>
Date: Tuesday, February 24, 2009	Sheet

Gambar 4.4 Rangkaian *h-bridge*



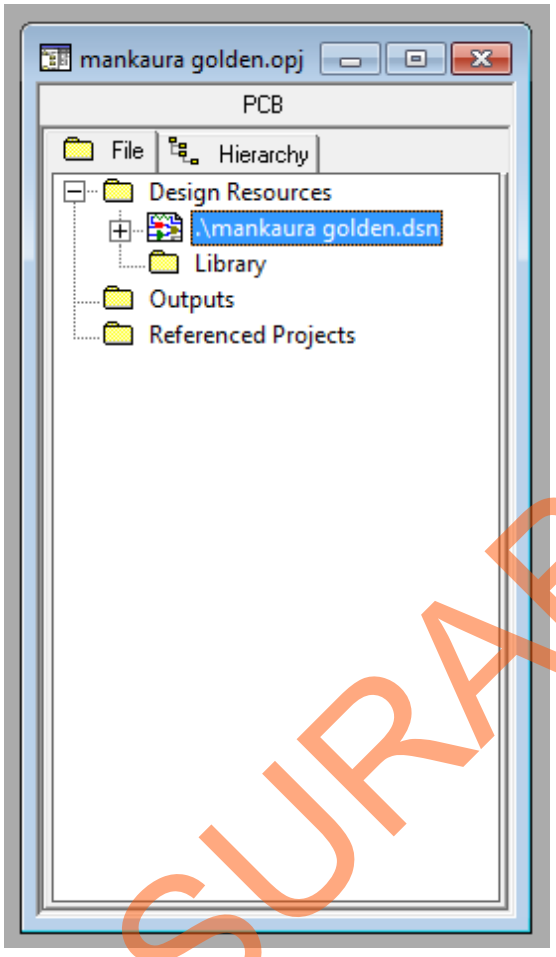
Gambar 4.5 *Schematic minimum system*



Gambar 4.6 Rangkaian *relay*

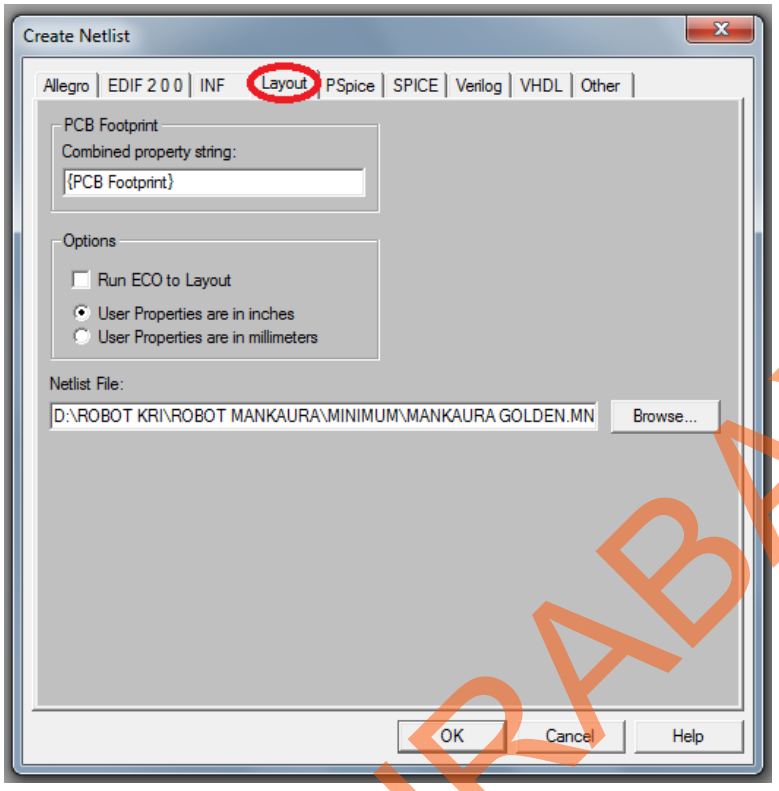
5. Untuk membuat schematic menjadi PCB maka langkah berikutnya membuat *layout,* langkah-langkah sebagai berikut :

    - Gambar *schematic* yang telah dibuat merupakan *file* yang ber ekstensi (.dsn). Untuk membuat *layout* PCB membutuhkan *file* yang berekstensi (.MNL) langkah yang harus dilakukan adalah *minimize project* hingga terlihat dialog seperti gambar 4.7.



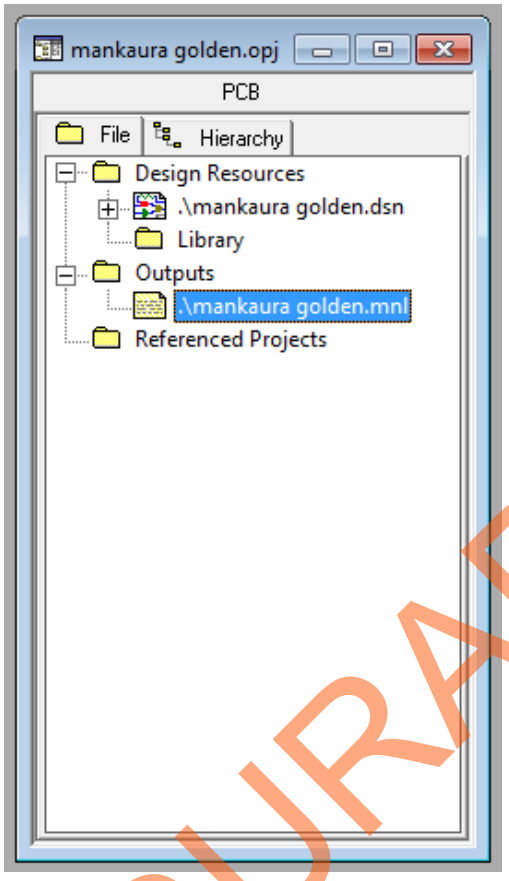
Gambar 4.7 Dialog *file* ekstensi (.dsn)

- Lalu klik pada *project* lalu pilih menu *Tool* ➔ *creat netlist*, sehingga tampil seperti pada gambar 4.8.



Gambar 4.8 *Create netlist*

- Setelah *create netlist* akan tampil diolog seperti gambar 4.8 ➔ lalu pilih *layout* ➔ pada menu *option* pilih *user properties are in inches* ➔ lalu klik OK. Selanjtnya lihat pada dialog *project* tadi pastikan pada Outputnya sudah membentuk ekstension *file* .mnl seperti pada gambar 4.9.

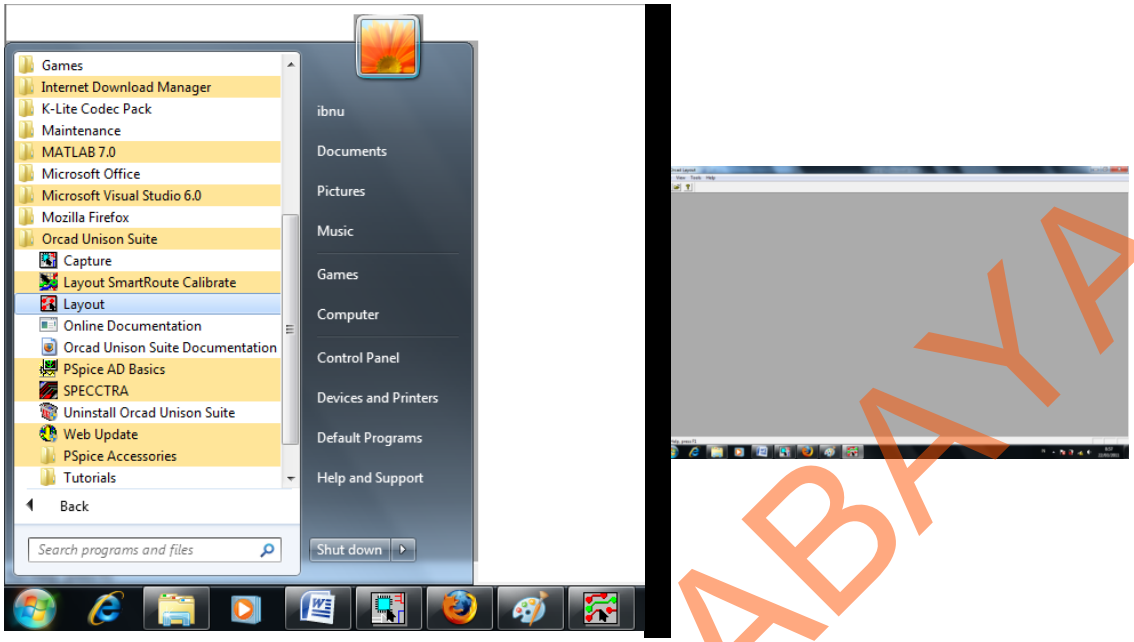


Gambar 4.9 *file* ekstensi (.mnl)

6. Setelah *file* telah berekstensikan (.mnl), maka langkah selanjutnya sebagai berikut :

   - *Start windows* ➔ pilih *all program*

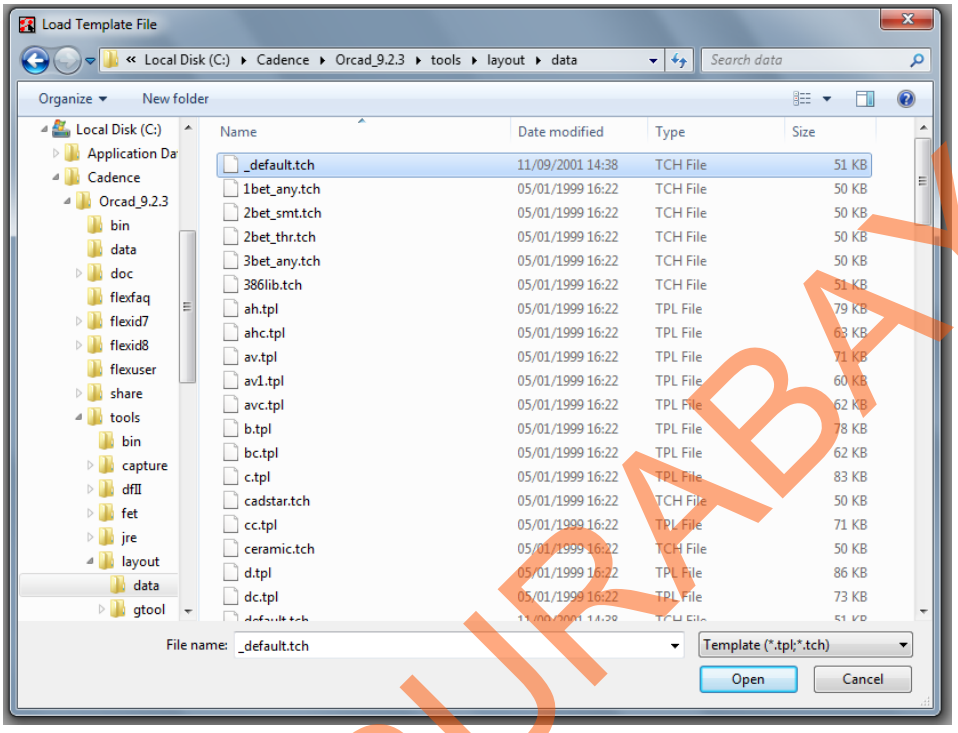
   - Setelah *allprogram* ➔ pilih Orcad *unison suite* ➔ pilih *layout* (digunakan untuk membuat *layout* PCB) seperti yang tampak pada Gambar 4.10.



Gambar 4.10 Cara membuka program *Orcad 9.23* dan jendela awal *layout* pada *Orcad 9.23*

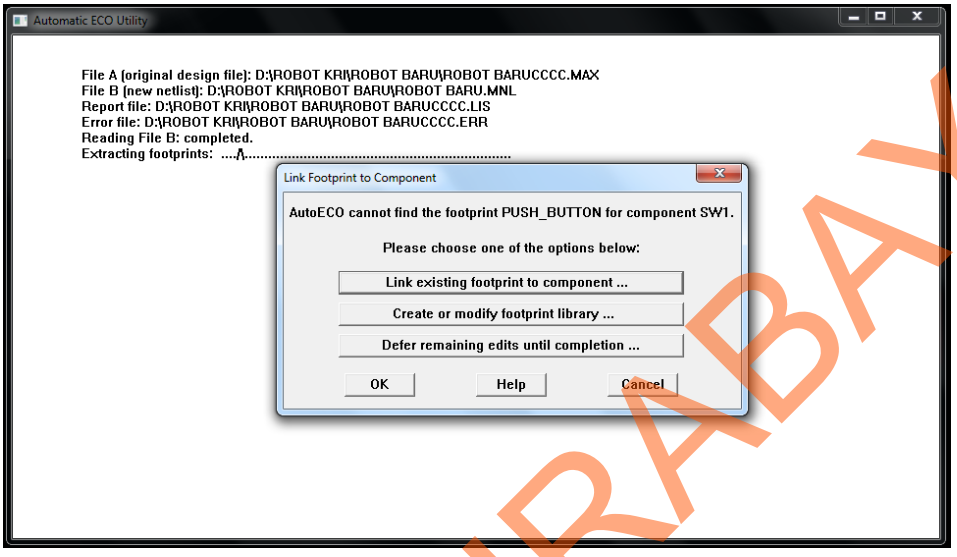
- Klik *file* ➔ *new*

- Lalu pilih *directory c: cadence* ➔ *Orcad 9.23* ➔ *tools* ➔ *layout* ➔ *data* ➔ *default.tch.*



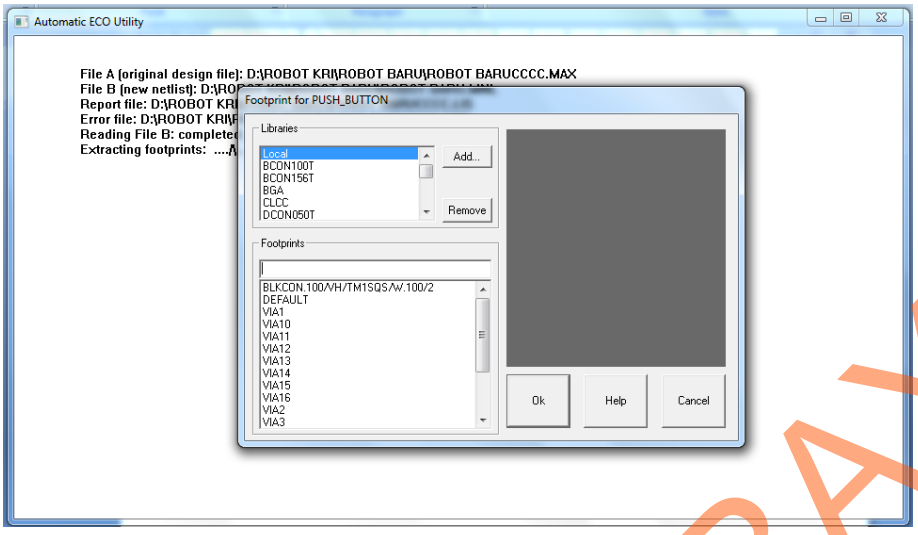
Gambar 4.11 Pencarian *default.tch*

- Setelah memilih *default.tch*, cari *file* dari *capture* yang telah dibuat yang telah berekstensikan (.mnl), hingga muncul dialog seperti pada gambar 4.12.



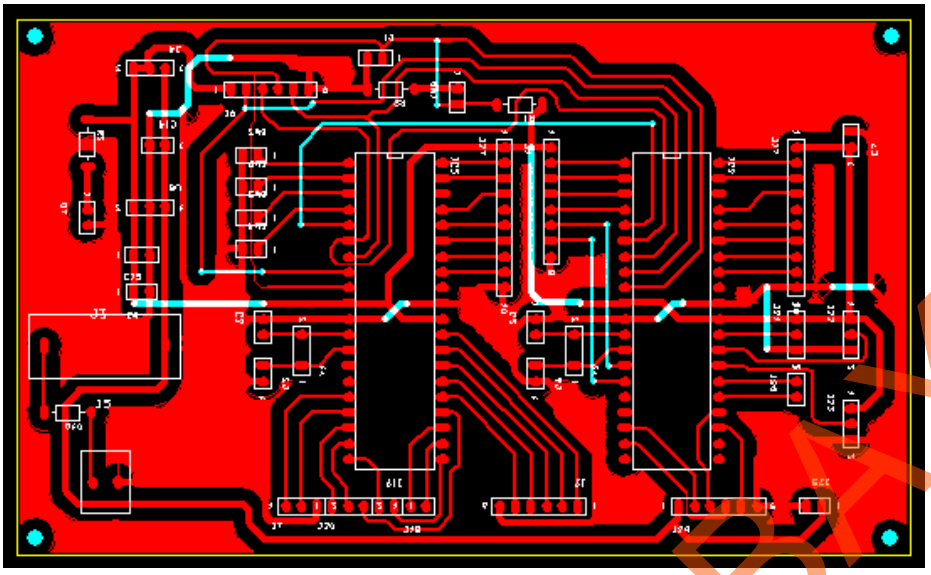
Gambar 4.12 Memilih tipe komponen

- Kemudian pilih *link exiting footprint to component*, hingga muncul gambar untuk pemilihan komponen.



Gambar 4.13 Langkah pencarian komponen

- Setelah semua komponen telah dipilih, langkah selanjutnya adalah me-*routing* jalur untuk dibuat menjadi PCB. Ada dua cara untuk me-*routing* jalur yaitu dengan manual *route* dan *auto route*. Berikut contoh gambar desain PCB yang sudah jadi.



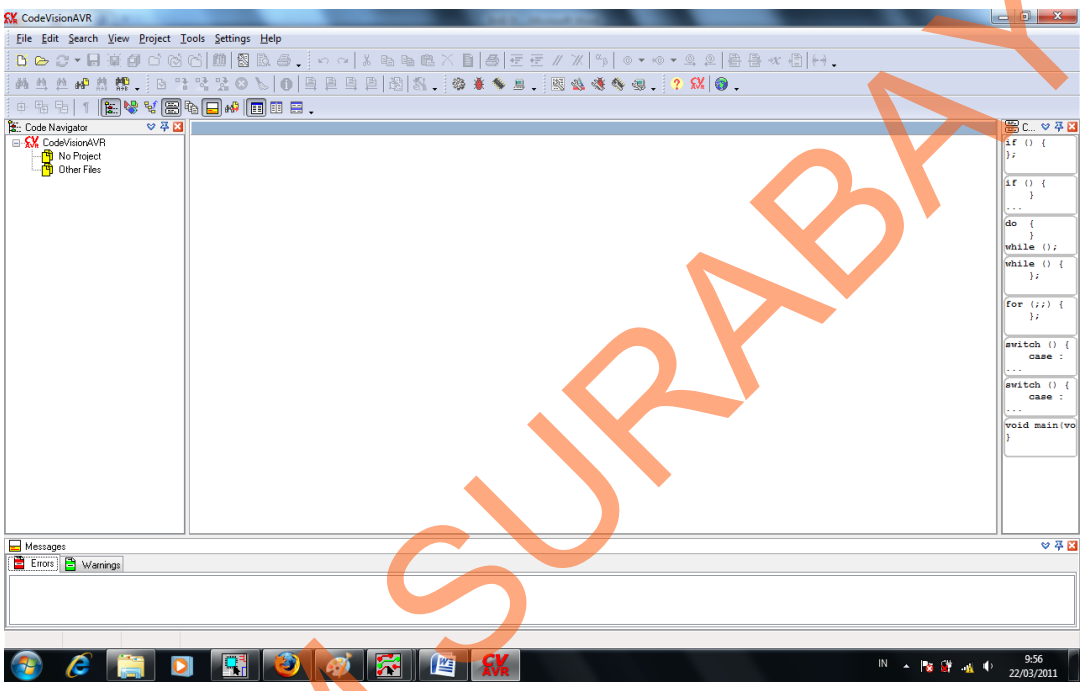
Gambar 4.14 Desain PCB *minimum system*

6..2CVAVR *Programmer*

CVAVR *Programmer* adalah *software* yang digunakan untuk membuat program dengan menggunakan bahasa C. Di dalam program CVAVR hanya digunakan untuk mikrokontroler tipe Atmel yang memiliki beberapa kelebihan daripada tipe MCS. Salah satu kelebihan CVAVR yaitu program yang diketikkan dengan menggunakan bahasa C dapat *dicompile* secara langsung tanpa *compiler*

lain untuk *mendownload* ke dalam *chip* mikrokontroler. Berikut langkah-langkah menjalankan CVAVR :

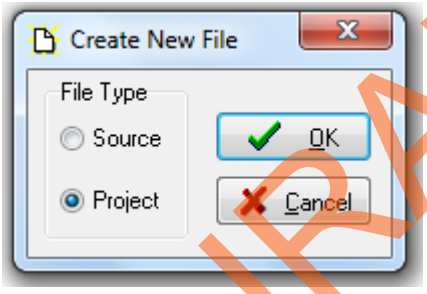
1. Jalankan program CVAVR yang ada pada komputer,hingga muncul jendela awal sesuai pada Gambar 4.15.



Gambar 4.15 Tampilan awal CVAVR

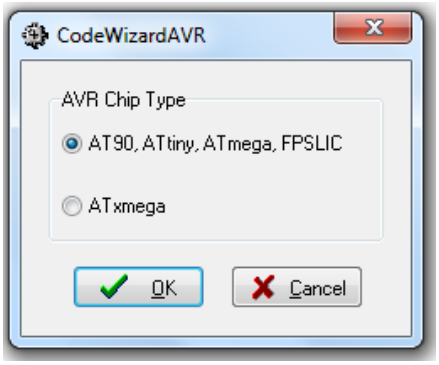
2. Langkah-langkah membuat program sebagai berikut :

   - Pilih menu *file* ➔ *new*, jika muncul dialog seperti pada gambar 4.16 pilih *project.*



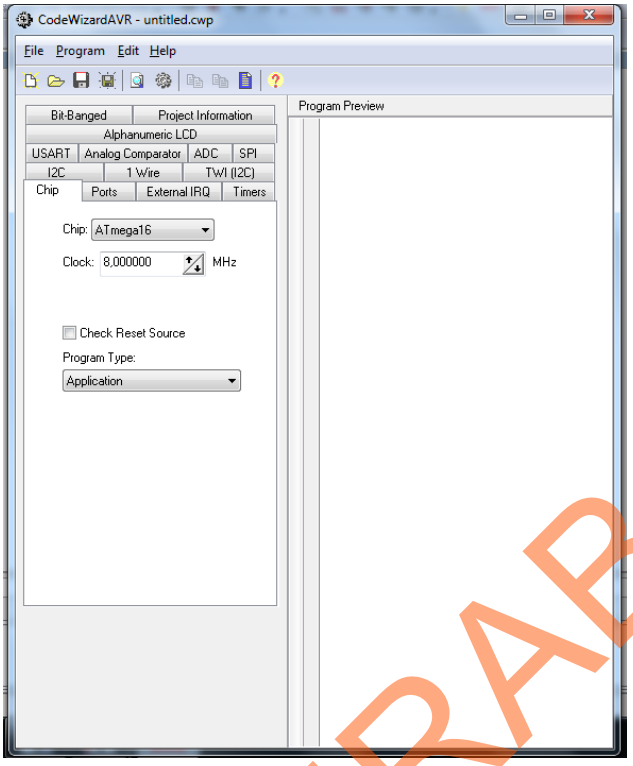
Gambar 4.16 *Create new file*

   - Setelah itu pilih OK dan akan muncul dialog konfirmasi pilih *yes*.
   - Setelah itu akan muncul pemilihan tipe *ic* mikrokontroler yang akan dibuat.



Gambar 4.17 pemilihan tipe *ic* mikrokontroler

- Kemudian muncul *codewizardAVR* yang digunakan untuk konfigurasi pin I/O, LCD, komunikasi serial, komunikasi *I2C*, *timer/counter,* dll.

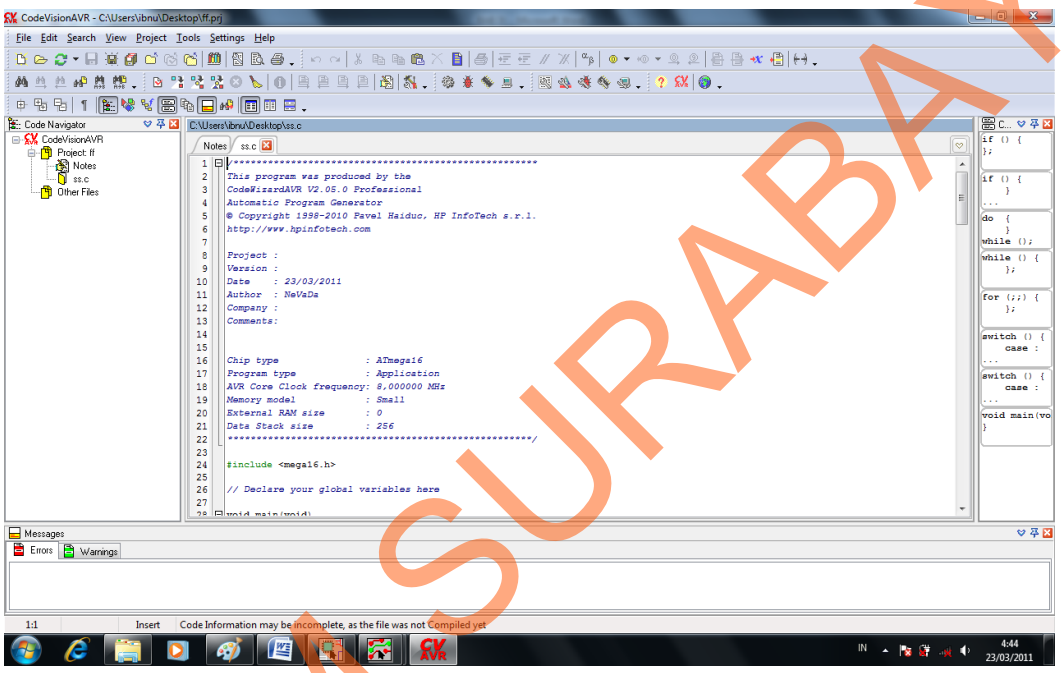


Gambar 4.18 *CodewizardAVR*

- Langkah selanjutnya pilih program ➔ *generate, save and exit*

- Setelah itu akan muncul dialog untuk penyimpanan *file* dan dialog tersebut akan muncul sebanyak 3 kali dengan ekstensi .c, .prj, dan .cwp.

- Mikrokontroler siap diprogram seperti pada gambar 4.19



Gambar 4.19 Lembar kerja CVAVR

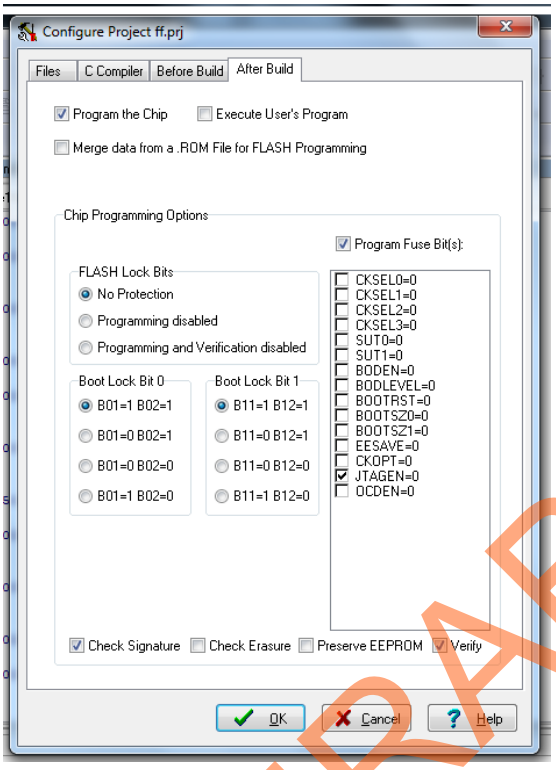
- Berikut contoh program :

```
while (1)
      {
      // Place your code here
        jalan();
        //step1_kiri();
        while (!(flag1 == 3))
        {
            if (sensor == 0b11111111)
            {
                dir_ka = 1;
                motor_ka = 1023;
                dir_ki = 1;
                motor_ki = 1023;
            }
            flag1 = flag1 + 1;
        }

        if (flag1 == 3)
        {
            do
            {
                dir_ka = 1;
                motor_ka = 1023;
                dir_ki = 0;
                motor_ki = 1023;
            }while(sensor == 0b00110000);
            flag1 = 0;
        }

        //step2_ambil()
        while (!(flag1 == 2))
        {
            if (sensor == 0b11111111)
            {
                dir_ka = 1;
                motor_ka = 1023;
                dir_ki = 1;
                motor_ki = 1023;
            }
            flag1 = flag1 + 1;
        }

        if (flag1 == 2)
        {
            dir_ka = 0;
            motor_ka = 1023;
            dir_ki = 0;
            motor_ki = 1023;
            delay_ms(100);
            //stop
            dir_ka = 1;
            motor_ka = 0;
            dir_ki = 1;
            motor_ki = 0;
            delay_ms(100);
        }
        master_out = 1;
        //program slave
        while (!(slave_in == 1))
        if (slave_in == 1)
        {
            flag_msk = 1;
            if (flag_msk == 1)
            {
                while (!(flag_angkat == 2))
```

```
                {
                    angkat_ka = 1;
                    flag_angkat = flag_angkat + 1;
                }

                if (flag_angkat == 2)
                {
                    angkat_ka = 0;
                    naik = 1;
                    delay_ms(1000);
                    naik = 0;
                    flag_angkat = 0;
                }

                while (!(sensor_ptr == 0))

                if (sensor_ptr == 1)
                {
                    while (!(flag_angkat == 2))
                    {
                        angkat_ka = 1;
                        flag_angkat = flag_angkat + 1;
                    }

                    if (flag_angkat == 2)
                    {
                        angkat_ki = 1;
                        naik = 1;
                        delay_ms(1000);
                        naik = 0;
                        flag_angkat = 0;
                    }
                }
                flag_msk = 0;
            }
            slave_out = 1;
        }

        //balik ke master
        if (master_in == 1)
        {
            flag_in = 1;
            if (flag_in == 1)
            {
                master_out = 0;
                //mundur
            }
        }
      };
}

/*
void step1_kiri()
{

}*/

void jalan()
{
    if (sensor == 0b11000000)
    {
        dir_ka = 1;
        motor_ka = 1000;
        dir_ki = 1;
        motor_ki = 1023;
    }

    if (sensor == 0b00110000)
    {
        dir_ka = 1;
        motor_ka = 1110;
        dir_ki = 1;
        motor_ki = 1023;
    }
```

```
    if (sensor == 0b00001100)
    {
        dir_ka = 1;
        motor_ka = 1023;
        dir_ki = 1;
        motor_ki = 1110;
    }

    if (sensor == 0b00000011)
    {
        dir_ka = 1;
-        motor_ka = 1023;
        dir_ki = 1;
        motor_ki = 1023;
    }

    if (sensor == 0b00001100)
    {
        dir_ka = 1;
        motor_ka = 1023;
        dir_ki = 1;
        motor_ki = 1023;
    }

    if (sensor == 0b00000110)
    {
        dir_ka = 1;
        motor_ka = 1000;
        dir_ki = 1;
        motor_ki = 1023;
    }

    if (sensor == 0b00000011)
    {
        dir_ka = 0;
        motor_ka = 950;
        dir_ki = 1;
        motor_ki = 1023;
    }

    if (sensor == 0b00000001)
    {
        dir_ka = 0;
        motor_ka = 1023;
        dir_ki = 1;
        motor_ki = 1023;
    }
}
```

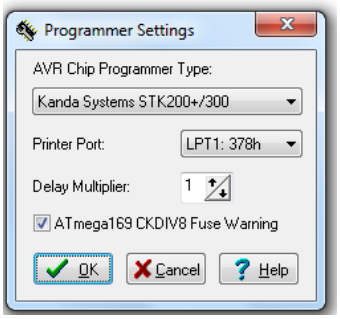
- Setelah pembuatan program selesai, maka program siap di *download* ke mikrokontroler.
- Sebelum *download* program *setting project configure* : pilih menu *project* ➔ pilih *after build* ➔ centang *program the chip*.



Gambar 4.20 *Configure project*

- Setelah itu OK

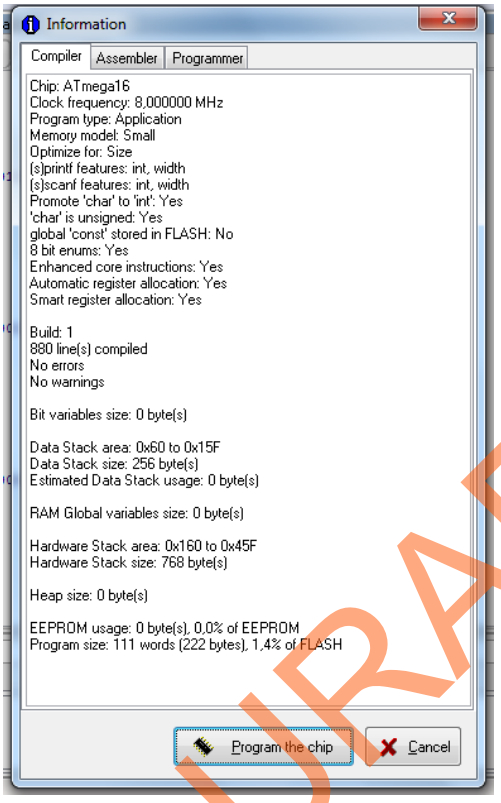
- Kemudian pilih menu *settings* ➔ *programmer* ➔ pilih *kanda system STK200+/300* (untuk konfigurasi paralel port), seperti muncul pada gambar 4.21.



Gambar 4.21 *Programmer settings*

- Mikrokontroler siap untuk *didownload*
- Pilih menu *project* ➔ *build all* ➔ hingga muncul dialog pada gambar 4.22.

- Jika tidak ada *error*, klik *program the chip*



Gambar 4.22 Proses *download* program

- Program telah masuk dan siap untuk dicoba.