

## BAB V

### IMPLEMENTASI DAN PENGUJIAN SISTEM

#### 5.1 HASIL IMPLEMENTASI

Pada tahap ini penulis mengimplementasikan hasil rancangan yang telah dibuat. Adapun hasil implementasi dari penelitian ini adalah sebagai berikut :



**Gambar 5.1** *Prototype* Sistem Parkir

Gambar diatas merupakan *prototype* sistem parkir yang telah dirancang penulis. Terlihat terdapat dua buah portal utama di pintu masuk dan keluar, dua buah servo dan dua buah tombol input, serta tambahan sensor *ultrasonik* HCSR-04 didepan portal. Lalu ada sepuluh buah plot parkir tersedia yang telah diletakkan masing-masing satu buah sensor *ultrasonik* HC-SR04 sebagai input untuk dikirimkan ke mikrokontroler.

Pada gambar 5.2 adalah cara kerja sistem yang pertama yaitu mobil datang dan bersiap untuk memasuki lahan parkir. Dalam kondisi portal masih tertutup, pengendara harus menekan tombol input yang terletak di depan pos masuk untuk

bisa membuka portal dan *interface* denah lokasi plot parkir akan muncul di layar monitor.



**Gambar 5.2 Keadaan Portal Tertutup**

Selanjutnya, pada gambar 5.3 keadaan portal sudah terbuka dan mobil bisa memasuki lahan parkir. Tidak jauh dari letak portal terdapat sensor *ultrasonik* HC-SR04. Apabila mobil lewat menghalangi sensor *ultrasonik* HC-SR04 maka portal akan tertutup kembali. Dan pengemudi yang dibelakangnya bisa memulai ulang proses sistem parkir dari awal.



**Gambar 5.3 Keadaan Portal Terbuka**

Setelah mobil masuk, pengendara dapat memarkirkan mobilnya pada pilihan plot yang tersedia sesuai dengan *interface*. Disaat telah masuk ke plot parkir yang dipilih, maka sensor *ultrasonik* HCSR-04 yang terhalang oleh mobil yang dimana itu merupakan input untuk mendeteksi keberadaan objek. Oleh karenanya pada *interface*, nomor plot akan terdeteksi dari yang tadi berwarna Hijau (kosong) akan menjadi warna Merah (terisi).

Adapun kondisi mobil yang telah masuk ke plot parkir seperti pada gambar 5.4 berikut :



**Gambar 5.4 Keadaan Mobil Pada Plot Parkir**

Pada gambar 5.5 merupakan kondisi mobil akan meninggalkan parkir. Disaat mobil telah keluar dari plot, maka sensor *ultrasonik* HCSR-04 kembali terbuka. Oleh karenanya pada *interface* nomor plot terdeteksi menjadi warna hijau semula. Untuk keluar dari tempat parkir, pengendara cukup menekan tombol didepan pos untuk membuka portal keluar. Setelah portal terbuka dan mobil melaju keluar, sama seperti di pintu masuk, di pintu keluar juga diletakkan sensor *ultrasonik* HCSR-04 jika mobil telah melewati sensor maka portal akan tertutup kembali.



**Gambar 5.5 Kondisi Mobil Akan Meninggalkan Parkiran**

## **5.2 PENGUJIAN PERANGKAT LUNAK**

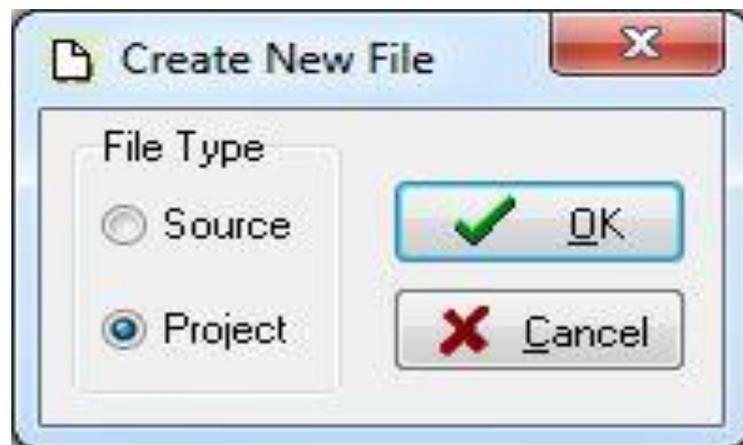
Hal pertama dilakukan dalam pengujian perangkat lunak adalah menentukan aplikasi (*software*) yang akan digunakan untuk mengisi program pada mikrokontroler ATmega 16.

### **5.2.1 CodeVisionAVR**

Untuk mikrokontroler ATmega 16 sinkron dengan banyak bahasa pemrograman seperti bahasa pemrograman *Assembler*, CodeVision AVR, bahasa C, Bascom AVR dan lainnya. Dalam pembuatan alat ini penulis menggunakan CodeVision AVR, sebab CodeVision AVR sangat kompetibel dengan *downloader* yang penulis gunakan. CodeVisionAVR merupakan sebuah *cross-compiler C*, *Integrated Development Environment (IDE)*, dan *Automatic Program Generator* yang didesain untuk mikrokontroler buatan Atmel seri AVR. CodeVisionAVR dapat dijalankan pada sistem operasi Windows 98, Me, dan Windows XP. Namun

penulis menggunakan sistem operasi Windows 7. Cross-compiler C mampu menerjemahkan hampir semua perintah dari bahasa ANSI C, sejauh yang diijinkan oleh arsitektur dari AVR, dengan tambahan beberapa fitur untuk mengambil kelebihan khusus dari arsitektur AVR dan kebutuhan pada sistem *embedded*.

Untuk pengujian yaitu pembuatan *listing program* baru, tekan file kemudian pilih *New Project*, lalu *Checklist Project* dan OK.



**Gambar 5.6 Menu Membuat Program Baru**

Maka selanjutnya akan muncul kembali jendela konfirmasi *Project*, pilih

YES :



**Gambar 5.7 Konfirmasi *Project***

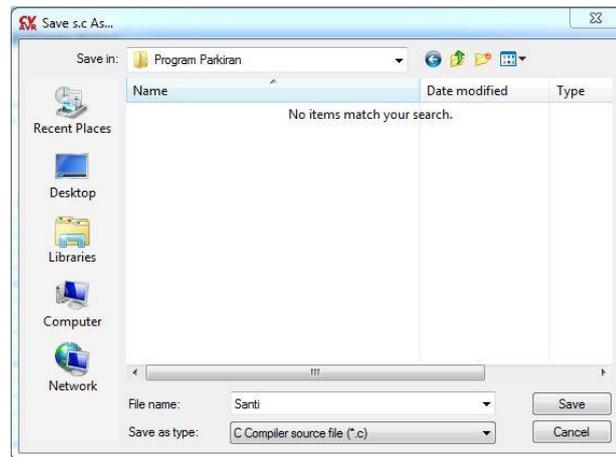
Kemudian akan muncul kembali jendela *Code Wizard AVR*, pada bagian ini tentukan *Chip* yang akan digunakan. karena *Chip* yang akan digunakan Atmega16 maka *checklist* pilihan yang pertama yaitu *AT90, Attiny, Atmega, FPSLIC* lalu klik OK.



**Gambar 5.8 Code Wizard AVR**

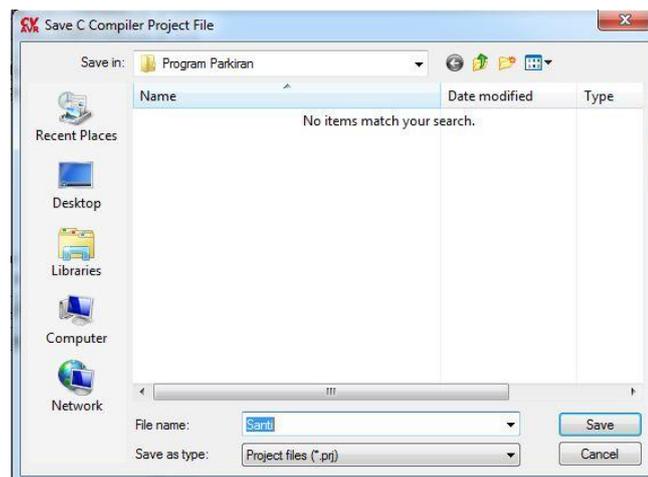
Berikutnya muncul jendela *Save C Compiler Source File* yaitu jendela untuk menyimpan file, pilih dimana file *project* akan kita simpan. Menyimpan file pada *Codevision AVR* terdiri dari tiga kali penyimpanan yaitu : *Save* yang pertama berupa *file* ekstensi *.C*, yang kedua *Save file* ekstensi *.prj*, dan yang ketiga *Save File* ekstensi *.cwp*,

untuk lebih jelasnya bisa dilihat pada gambar 5.10, 5.11, dan 5.12 berikut :



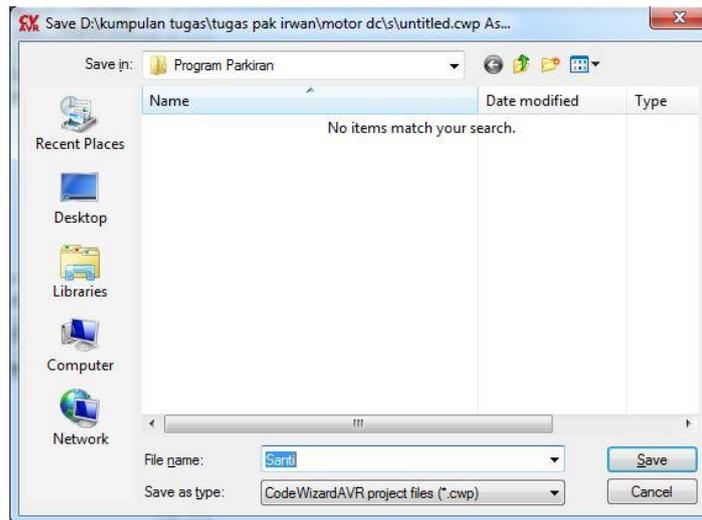
**Gambar 5.9 Save Pertama file ekstensi .C**

File ekstensi.c menyimpan data agar Codevision AVR diprogram menggunakan bahasa C.



**Gambar 5.10 Save Kedua File Ekstensi .prj**

File ekstensi.prj digunakan untuk menyimpan data proyek dan pengaturan dan dapat mencakup rujukan pada berkas lainnya yang digunakan oleh proyek.



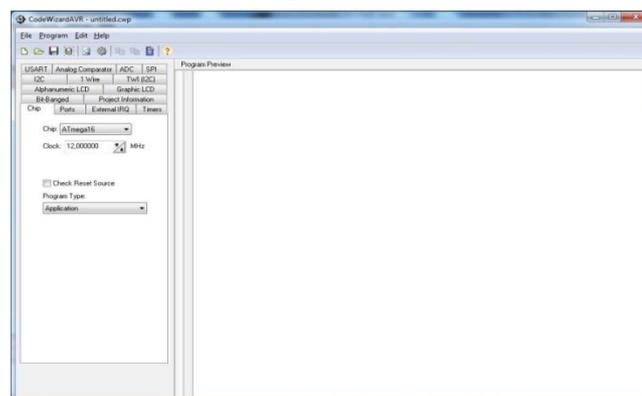
**Gambar 5.11 Save Ke Tiga File Ekstensi .cwp**

File ekstensi.cwp dipilih untuk mempermudah penulisan *source code*.

Setelah tahapan *Save File*, yang akan dilakukan selanjutnya adalah masuk ke dalam jendela *Code Wizard AVR*. Di jendela ini terdiri dari beberapa *Tab* pilihan yaitu terdiri dari :

1. *Tab Chip*

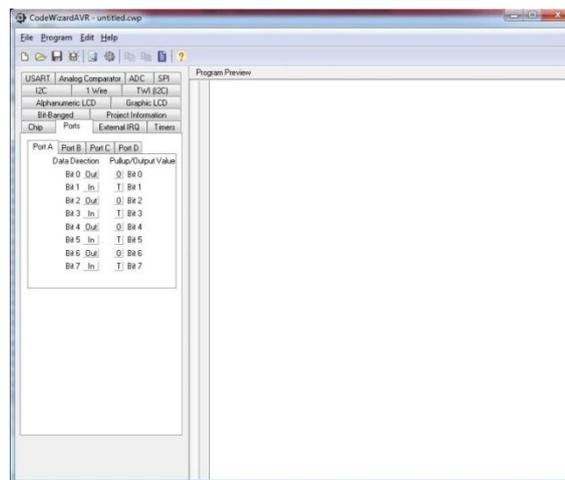
Yaitu *Tab* yang menentukan pilihan *Chip* yang digunakan Atmega16, *Clock* yang digunakan 12.000.000 *MHz*



**Gambar 5.12 Tab Chip**

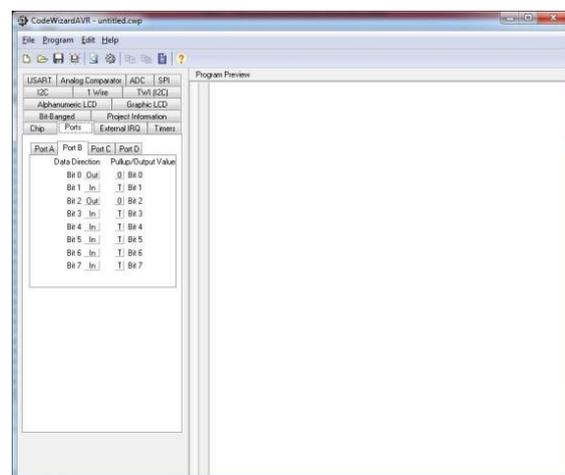
## 2. Tab *Ports*

Port A : Pada Tab *Ports* akan dipilih I/O Port mana saja yang akan digunakan yaitu Port A : Pin A.0, Pin A.2, Pin A.4, dan Pin A.6 *pulup/output value* dirubah menjadi out.



**Gambar 5.13 Menentukan Pin Untuk Sensor *Ultrasonik HC-SR04***

Port B : Pada Tab *Ports* akan dipilih I/O Port mana saja yang akan digunakan yaitu Port A : Pin A.0 dan Pin A.2 *pulup/output value* dirubah menjadi out.

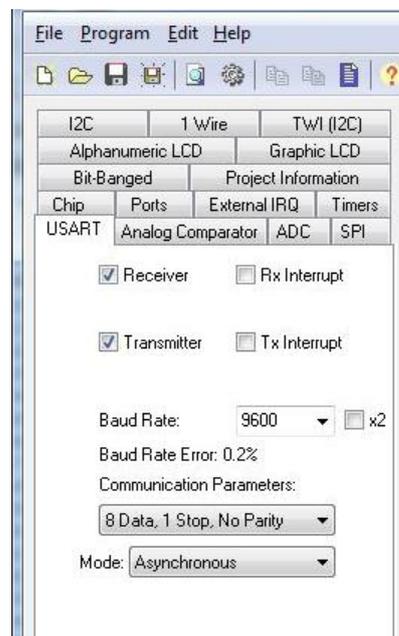


**Gambar 5.14 Menentukan Pin Untuk Sensor *Ultrasonik HC-SR04***



### 3. Tab USART

USART (*Universal Synchronous and Asynchronous Serial Receiver and Transmitter*) merupakan komunikasi yang memiliki fleksibilitas tinggi, yang dapat digunakan untuk melakukan transfer data baik antar mikrokontroler maupun dengan modul-modul eksternal termasuk PC yang memiliki fitur UART. Tab USART digunakan untuk mengaktifkan *transmitter*, *receiver*, *baud rate*, dan *communication parameters*.



**Gambar 5.17 Tab USART**

*Transmitter* data (TxD) adalah sinyal *actual* yang dikirimkan dari satu perangkat ke perangkat lain.

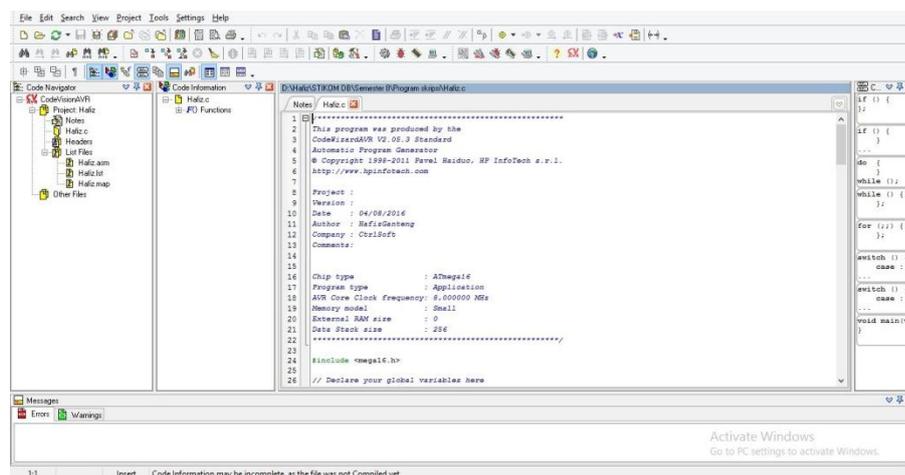
*Received* data (RxD) adalah sinyal yang diterima dari perangkat lain, pada perangkat lain tersebut sinyal didapat dari sinyal TxD (*Transmitted* data).

*Baud Rate* merupakan kecepatan pengiriman data antara perangkat dengan komputer. 1 *baud* merupakan 1 buah karakter yang dikirim. Besaran baud

rate ini ada beberapa: 110, 1200 2400, 9600 19200, 38400, 57600, 115200. Satuan *baud rate* adalah bps, yang berarti *baud per second*. Pada tab USART ini *baud rate* yang digunakan adalah 9600 bps, berarti data yang dikirim memiliki laju 9600 karakter per detik.

*Communication parameters* digunakan untuk berkomunikasi antara perangkat/instrumen dengan komputer. Pengaturan komunikasi yang digunakan adalah 8 Data, 1 *stop*, *no parity* yang berarti jumlah data bit yang dikirim sebanyak 8, tanda bit yang dikirim sudah selesai, dan tanpa ada bit yang *error*.

Kemudian setelah menentukan *Chip*, *Port I/O*, *ADC*, akan muncul jendela *coding* program yang berupa *coding default*. Disini mulai dilakukan pengetikan *listing program*.



```

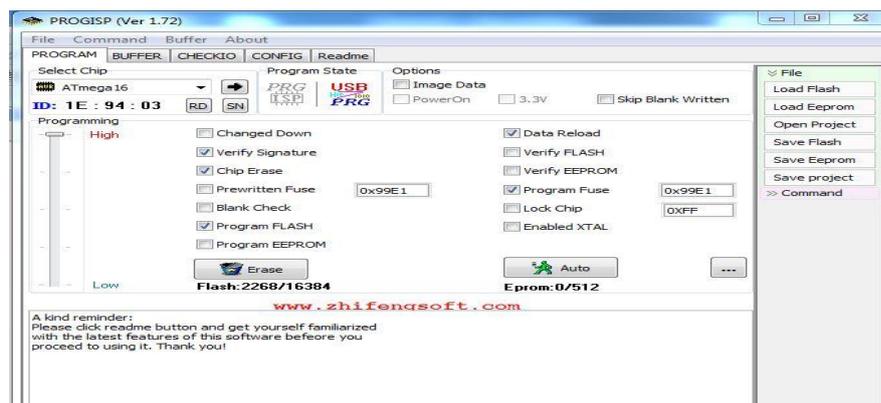
1  *****
2  This program was produced by the
3  CodeWizardAVR V2.05.3 Standard
4  Automatic Program Generator
5  © Copyright 1998-2011 Pavel Radzou, HP InfoTech s.r.l.
6  http://www.hpinfoTech.com
7
8  Project :
9  Version :
10 Date : 04/08/2016
11 Author : RafisGeteng
12 Company : CtrlSoft
13 Comments:
14
15
16 Chip type : ATmega16
17 Program type : Application
18 AVR Core Clock frequency: 8,000,000 MHz
19 Memory model : Small
20 External RAM size : 0
21 Data Stack size : 256
22 *****
23
24 #include <mega16.h>
25
26 // Declare your global variables here

```

**Gambar 5.18 Listing Program**

Setelah *listing rogram* selsai dibuat, kemudian tekan tombol *Ctrl+F9* atau klik *Program* pilih *Build All* lalu OK. Tahapan berikutnya adalah memasukan program ke dalam mikrokontroler, *USB ISP Programmer* ini adalah *programmer tool* untuk mengunggah kode program terkompilasi (berkas dalam format *Intel*

HEX) ke mikrokontroler Atmel yang mendukung ISP (*In-System Programming*). Alat ini dapat digunakan dari Windows 7, dikenali sebagai USB HID (*Human Interface Device*) dengan *Vendor ID (VID) 0x03EB* dan *Product ID (PID) 0xC8B4*. Untuk menggunakan alat ini penulis menggunakan piranti lunak yang mendukung *USB ISP*, seperti *ProgISP Programmer*. Untuk lebih jelasnya perhatikan gambar 5.19 berikut.



**Gambar 5.19** *PROGISP (Ver 1.72)*

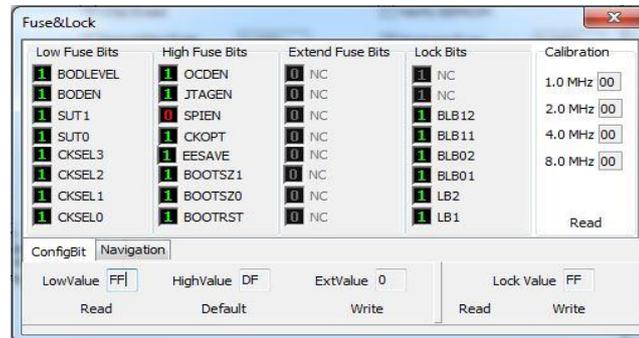
Pada gambar 5.19 adalah gambar tampilan awal *progisp* disini penulis menggunakan ver 1.72. Langkah berikutnya untuk memasukan program ke dalam mikrokontroler yaitu merubah *Fuse & Lock* pada *progisp*, untuk lebih jelasnya perhatikan gambar 5.20.



**Gambar 5.20** *Setting Fuse & Lock ProgISP*

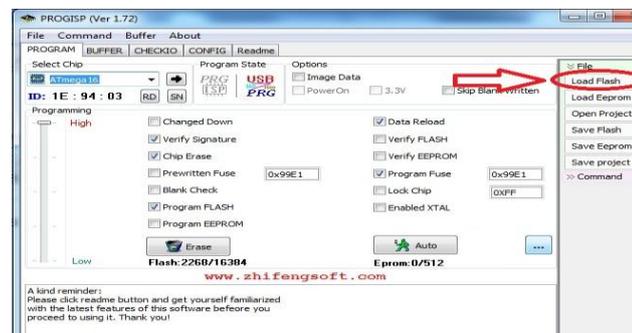
Selanjutnya setting *Fuse & Lock* seperti yang terlihat pada gambar 5.21

berikut :



**Gambar 5.21 Fuse & Lock ProgISP**

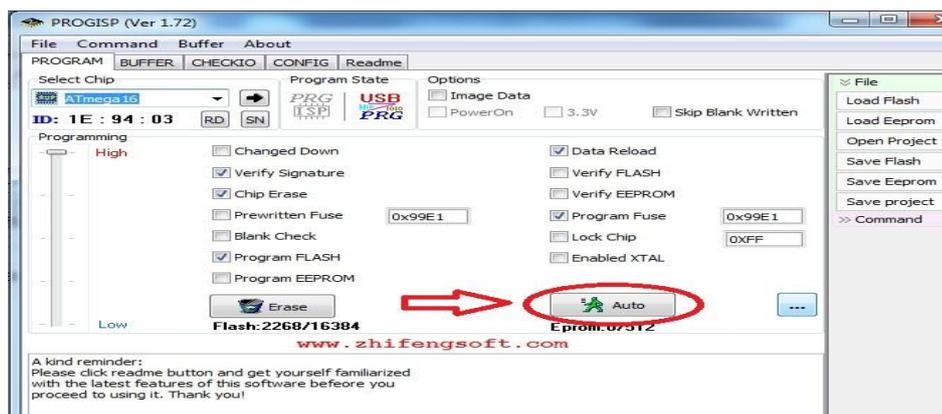
Setelah setting *fuse & lock progISP* kemudian klik *Load Flash* pilih program yang akan dimasukkan kedalam mikrokontroler lalu klik OK.



**Gambar 5.22 Load Flash**

Setelah itu baru kemudian klik *Auto*, selanjutnya perhatikan gambar 5.23

berikut :



**Gambar 5.23 Auto**

Jika dibagian keterangan kiri bawah terdapat pesan *successfully done* itu tandanya program yang kita buat telah berhasil dimasukkan kedalam mikrokontroler.

### 5.2.2 Visual Basic.Net 2008

Pengujian pada perangkat lunak (*software*) *Visual Basic.Net 2008* ini terdiri atas 4 (empat) bagian utama yaitu :

1. *Form* Menu Utama

*Form* ini merupakan halaman utama saat kita menjalankan aplikasi sistem informasi parkir mobil.

Adapun bentuk tampilan dari menu utama dapat dilihat pada gambar 5.24 berikut :



**Gambar 5.24 Tampilan *Form* Menu Utama**

2. *Form* Komunikasi Serial

Pada *form* ini berisikan cara mengkoneksikan port serial, supaya dapat terhubung pada aplikasi sistem informasi parkir mobil.

Adapun tampilan dari *form* komunikasi *serial* dapat dilihat pada gambar 5.25 berikut :

**Komunikasi Serial**

Baudrate

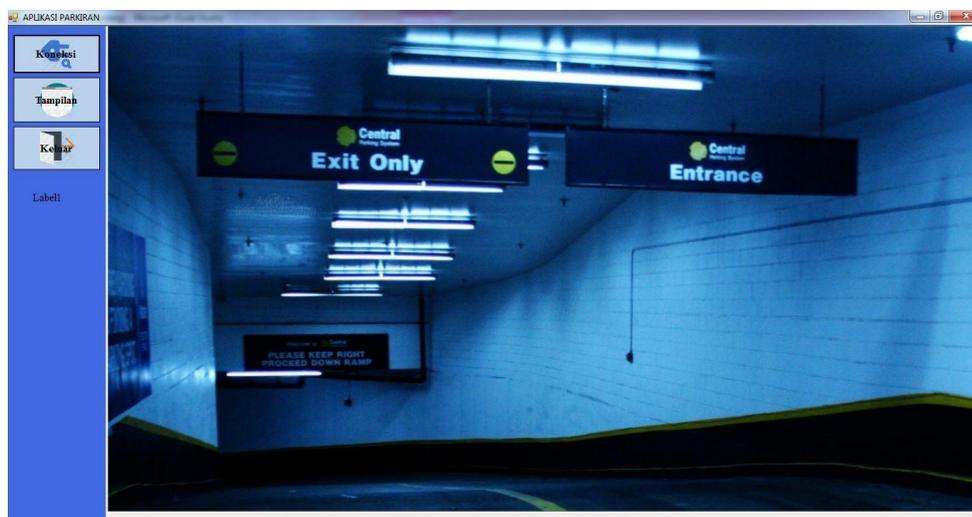
Serial Com Port

Sambung Refresh Putus Tutup

**Gambar 5.25 Tampilan *Form* Komunikasi Serial**

### 3. *Form* Aplikasi Parkir

*Form* ini merupakan tampilan awal untuk kita memulai pengujian aplikasi pada sistem parkir. Dimulai dengan kita mengkoneksikan serial, lalu membuka menu tampilan sehingga tampil di layar monitor, maupun menu keluar untuk mengakhiri proses sistem parkir. Adapun *form* aplikasi parkir dapat dilihat pada gambar 5.26 seperti berikut :



**Gambar 5.26 *Form* aplikasi parkir**

Selain pengujian *software*, penulis juga melakukan pengujian dari aplikasi yang dibuat. Adapun beberapa tahap pengujian yang telah penulis lakukan adalah sebagai berikut :

1. Pengujian Modul Komunikasi *Serial*

Pada tahap ini dilakukan pengecekan komunikasi serial untuk mengetahui apakah *device* sudah terkoneksi dengan baik atau belum.

Hasil pengujian pada modul ini dapat dilihat pada table 5.1 :

**Tabel 5.1 Pengujian Modul Komunikasi Serial**

Modul yang diuji	Prosedur pengujian	Masukan	Keluaran yang diharapkan	Hasil yang didapat	Kesimpulan
Komunikasi serial	- Hubungkan port serial yang tersambung pada alat dengan port yang ada pada laptop	- Masuk ke tampilan form menu utama lalu klik tombol komunikasi serial - Pilih nama port, sesuaikan yang sudah diatur dalam program - Pilih baudrate - Klik tombol connect	- Koneksi berhasil	- Koneksi berhasil	Baik

## 2. Pengujian Modul *Form Tampilan Interface*

Pada tahap ini dilakukan pengujian pada modul melihat tampilan untuk mengetahui apakah nomor pada *interface* telah berjalan dengan baik.

Hasil pengujian pada modul ini penulis sajikan pada tabel 5.2

**Tabel 5.2 Pengujian Modul *Form Tampilan***

Modul yang diuji	Prosedur pengujian	Masukan	Keluaran yang diharapkan	Hasil yang didapat	Kesimpulan
Melihat form tampilan (berhasil)	<ul style="list-style-type: none"> <li>- Jalankan program dengan menekan tombol F5</li> <li>- Hubungkan terlebih dahulu komunikasi serial</li> <li>- Memilih port yang digunakan dalam pengujian</li> <li>- Memilih baudrate</li> <li>- Mengklik tombol connect</li> </ul>	<ul style="list-style-type: none"> <li>- Mengklik tombol tampilan yang ada pada form menu utama</li> </ul>	<ul style="list-style-type: none"> <li>- Menampilkan form tampilan dengan benar</li> </ul>	<ul style="list-style-type: none"> <li>- Dapat melihat langsung ketersediaan lahan parkir pada form tampilan <i>interface</i></li> </ul>	Baik

## 5.3 PENGUJIAN ALAT

### 5.3.1 Pengujian Tegangan Sumber

Tahap pertama yang dilakukan adalah pengecekan catu daya. *Battery* yang digunakan adalah memiliki keluaran sebesar 5 V.

Hasil pengujian tegangan yang dihasilkan oleh *battery* dapat dilihat pada tabel 5.3.

**Tabel 5.3 Pengujian Tegangan Sumber**

<b>Sumber Arus</b>	<b>Tegangan Input</b>	<b>Tegangan Output</b>
<i>Battery</i>	5 V	5 V

### 5.3.2 Pengujian Tegangan AT-Mega16

Rangkaian ini merupakan otak dari seluruh rangkaian. Semua rangkaian yang ada dikendalikan *input output*-nya oleh rangkaian mikrokontroler ini. Proses pengujian rangkaian ini adalah dengan menghubungkan setiap *port* dengan beberapa LED. Adapun hasil dari pengujian tegangan AT Mega16 ini dapat dilihat pada tabel 5.4.

**Tabel 5.4 Pengujian Tegangan ATMega16**

<b>Sumber</b>	<b>Tegangan Input</b>	<b>Tegangan Output</b>
AT Mega 16	5 V	4.8 V

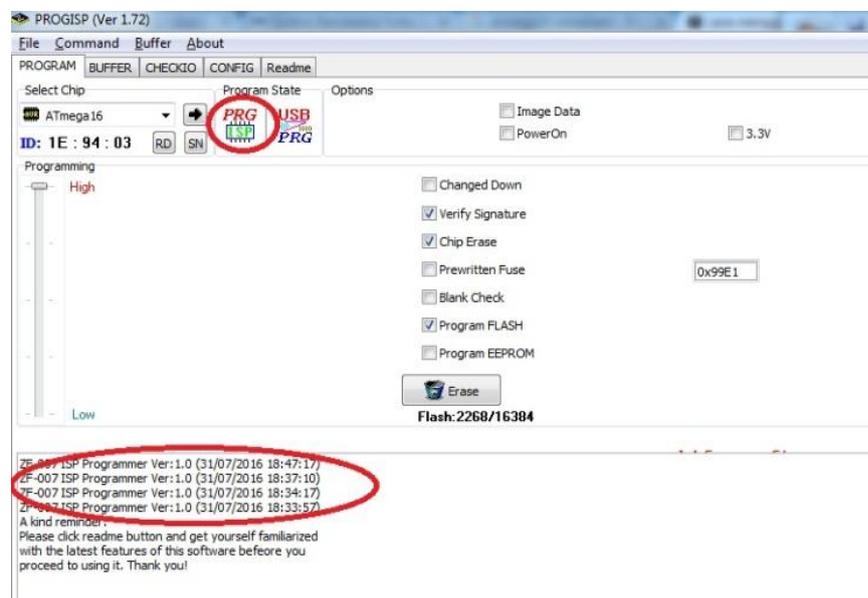
### 5.3.3 Pengujian Downloader

Tahap pertama dilakukan *testing downloader*, yang pertama dilakukan adalah pengecekan rangkaian *regulator* ke mikrokontroler apakah sudah berukuran 5 V DC, sebab tegangan yang baik untuk mikrokontroler adalah 5 V DC, apabila kurang maupun lebih dapat mengganggu kinerja mikrokontroler bahkan dapat merusak mikrokontroler.

Hal berikutnya yang akan dilakukan adalah menghubungkan PC dengan *downloader* menggunakan *port* serial usb. Rangkaian *downloader* terdiri dari USB ISP, 1 buah mikrokontroler dan *socket* 40 *pin*, dan 1 buah Xtal 12 *Mhz* serta

2 buah *capasitor*. Untuk port ke 1 pada *header usb isp* dihubungkan ke pin ke 6 pada Atmega16, port ke 2 *header usb isp* dihubungkan ke pin Vcc Atmega 16, port ke 3 pada *header usb isp* tidak dihubungkan atau *Nc (No Connection)*, port ke 4, 6, 8, 10 pada *header usb isp* dihubungkan ke Gnd Atmega16, port ke 5 pada *header usb isp* dihubungkan ke pin ke 9 *reset* Atmega16, port ke 7 pada *header usb isp* dihubungkan ke pin 8 SCK Atmega16, dan port ke 9 pada *header usb isp* dihubungkan ke pin ke 7 MISO Atmega16.

Jika pada bagian *PROGISP* tulisannya berubah menjadi warna merah maka *USB ISP* siap dipakai. Perhatikan gambar 5.27 berikut :



**Gambar 5.27 USB ISP Siap Digunakan**

### 5.3.4 Pengujian Sensor *Ultrasonnik* HC-SR04

Untuk pengujian sensor *ultrasonnik* HC-SR04, hal yang harus dilakukan adalah mengukur tegangan keluaran (*output*) dari HC-SR04 tersebut. Untuk mengetahui sensor bekerja dengan baik, Hal berikutnya yang dilakukakn yaitu pengujian terhadap respon dari *interface* ketika sensor *ultrasonnik* HC-SR04 telah

terhalangi objek lalu mengirimkan data melalui mikrokontroler. Berapa lama waktu yang dibutuhkan pada *interface* untuk mengubah warna hijau menjadi warna merah.

**Tabel 5.5 Pengujian Jarak Respon pada *Interface***

Nomor Plot Parkir	Warna Plot parkir saat kosong	Warna Plot parkir saat terisi	Jarak Respon pada <i>interface</i>
P1	Hijau	Merah	$\leq 5$ cm
P2	Hijau	Merah	$\leq 5$ cm
P3	Hijau	Merah	$\leq 5$ cm
P4	Hijau	Merah	$\leq 5$ cm
P5	Hijau	Merah	$\leq 5$ cm
P6	Hijau	Merah	$\leq 5$ cm
P7	Hijau	Merah	$\leq 5$ cm
P8	Hijau	Merah	$\leq 5$ cm

### 5.3.5 Pengujian Komunikasi Serial

Pengujian ini dilakukan untuk mengetahui apakah komunikasi antara laptop dan mikrokontroler sudah berjalan sesuai dengan semestinya.

Adapun hasilnya dapat dilihat pada gambar 5.28 berikut :



**Gambar 5.28 Hasil Pengujian Dari Komunikasi Serial**

**Tabel 5.6 Pengujian HC-SR04**

No	Nilai Sensor (cm)	Nilai pada Aplikasi	Status pada <i>interface</i>
1	1 - 5	Tebaca	Merah
2	> 5	Tidak Terbaca	Hijau

#### 5.4 ANALISIS SISTEM SECARA KESELURUHAN

Untuk mendeteksi apabila terjadi kesalahan setelah uji coba, maka perlu dilakukan analisa rangkaian secara keseluruhan. Dari seluruh proses yang telah dilakukan, baik pengujian perangkat keras maupun perangkat lunak, dapat dikatakan bahwa alat ini dapat berfungsi sebagaimana yang penulis inginkan. Proses pembacaan sensor *ultrasonik* HC-SR04 pun tidak terjadi kesalahan pembacaan data, motor servo dapat berputar sesuai program yang di buat penulis,

dan pada layar monitor dapat menampilkan *interface* yang sesuai dengan kondisi sebenarnya untuk pengendara.

Pengujian ini dilakukan untuk menunjukkan bahwa sistem informasi parkir mobil berbasis *interface* ini dapat bekerja sesuai dengan tujuan dari pembuatan. Pengujian ini dilakukan dengan cara sebagai berikut :

1. Input tombol parkir akan membaca keadaan sensor *ultrasonik* HC-SR04 yang terhalang oleh mobil, dimana hasil dari pembacaan sensor ini akan dijadikan input pada mikrokontroler untuk diproses. Kemudian data dari mikrokontroler untuk selanjutnya akan dikirimkan ke laptop melalui rangkaian MAX 232 dimana untuk penghubung antara MAX dan laptop menggunakan *converter serial to USB*.
2. Apabila ada plot parkir yang masih kosong, maka aplikasi akan menampilkan *interface* dengan plot parkir berwarna hijau, sementara untuk yang sudah diisi plot akan berwarna merah dan portal terbuka. Apabila plot parkir sudah penuh, maka aplikasi tidak akan menampilkan *interface* serta portal akan tetap tertutup.