

APLIKASI REMOTE KONTROL CPU/LAPTOP JARAK JAUH DENGAN MEDIA SERIAL HANDPHONE DENGAN MIKROKONTROLER

Rozali Toyib¹, Juni Hidayatullah²

^{1,2} Informatika, Fakultas Teknik Universitas Muhammadiyah Bengkulu
Jl. Bali Po. Box, 118 Kota Bengkulu 38119 INDONESIA
(Telp 0736-22765 Fak. 0736-26161)

¹ rozalitoiyib@gmail.com

² junihadi@yahoo.com

Abstrak : Teknologi *remote control* telah banyak dikembangkan dengan memanfaatkan berbagai media transmisi. Mikrokontrolerbekerja dalam bahasa mesin sedangkan manusia sulit untuk mengerti bahasa mesin. *Software* yang sering digunakan yaitu C, *Basic*, atau *Assembler*, selanjutnya dengan bantuan *Compiler* program akan diterjemahkan dalam bahasa mesin. Mikrokontroler yaitu tipe AVR salah satunya adalah *Atmega16*. Dari aspek *Android* merupakan *system* operasi untuk *handphone*. *System* operasi ini menyediakan *platform* terbuka bagi para pengembang untuk menciptakan *aplikasi* baru dari *CodeVisionAVR* dan *Microsoft Visual Studio*. Dilihat dari aspek pengendalian robot, *bluetooth* merupakan teknologi komunikasi tanpa kabel yang digunakan untuk transfer data antara robot dan aplikasi *blueterm* yang bisa digunakan pada sistem operasi untuk *handphone*.

Kata Kunci: *Remote Control*, *Compiler*, Mikrokontroler, *Hanphone*, *Bluetooth*

***Abstract* : Remote control technology has been developed by utilizing a variety of transmission media. microcontroller work in the machine while human language is difficult to understand the language of the machine. Software is often used, namely C, Basic, or Assembler, Compiler further with the help of the program will be translated into machine language. namely type AVR microcontroller which is ATmega16 one. From the aspect of Android is an operating system for mobile phones. The operating system provides an open platform for developers to create new applications from CodeVision AVR and Microsoft Visual Studio. Viewed from the aspect of robot control, Bluetooth is a wireless communication technology that is used to transfer data between the robot and blueterm applications that can be used in the operating system for mobile phones.**

Keyword: *Remote Control*, *Compiler*, *Microcontroler*, *Hanphone*, *Bluetooth*.

I. PENDAHULUAN

Teknologi *remote control* telah banyak dikembangkan dengan memanfaatkan berbagai

media transmisi. Beberapa diantaranya adalah *remote control* dengan memanfaatkan media inframerah, *bluetooth*, gelombang radio, internet dan saluran telepon baik pada bermacam macam peralatan elektronik, kendaraan dan lain-lain. Sistem *remote control* melalui saluran telepon memiliki keunggulan dalam hal jarak jangkauan dan kepraktisan dibanding media lainnya.

Perangkat pengembangan suatu sistem mikrokontroler adalah sangat penting untuk melakukan eksperimen dengan mikrokontroler yang dipilih, mikrokontroler bekerja dalam bahasa mesin sedangkan manusia sulit untuk mengerti bahasa mesin. *Software* yang sering digunakan yaitu C, *Basic*, atau *Assembler*, selanjutnya dengan bantuan *Compiler* program akan diterjemahkan dalam bahasa mesin.

CodeVision AVR C Compiler adalah *software* yang digunakan untuk membuat program

mikrokontroler AVR dalam bahasa C. Program tersebut kemudian diterjemahkan oleh *CodeVision AVR C* menjadi kode *heksadesimal* yang akan di *download* ke dalam *chip* mikrokontroler AVR. *CodeVision AVR* dapat mengimplementasi hampir semua instruksi bahasa C yang sesuai dengan arsitektur AVR, bahkan terdapat keunggulan spesifik dari AVR. Hasil kompilasi objek *CodeVision AVR* bisa digunakan sebagai *source debug* dengan AVR Studio *debugger* dari ATMEL.

Aplikasi kontrol CPU, dilihat dari aspek mikrokontroler yaitu tipe AVR salah satunya adalah *Atmega16*. Dari aspek *Android* merupakan sistem operasi untuk *handphone*. Sistem operasi ini menyediakan *platform* terbuka bagi para pengembang untuk menciptakan *aplikasi* baru dari *CodeVision AVR* dan *Microsoft Visual Studio*. Dilihat dari aspek pengendalian robot, *bluetooth* merupakan teknologi komunikasi tanpa kabel yang digunakan untuk transfer data antara robot dan *aplikasi blueterm* yang bisa digunakan pada sistem operasi untuk *handphone*.

Dari latar belakang di atas, maka peneliti tertarik untuk membangun sebuah **Aplikasi Remot Kontrol CPU/Laptop Jarak Jauh Dengan Media Serial Handphone Dengan Mikrokontroler**.

II. LANDASAN TEORI

A. *Android*

Android adalah sebuah *toolkit software* yang baru untuk perangkat bergerak yang dibuat oleh Google dan *Open Handset Alliance*. Dalam beberapa tahun, *Android* diharapkan dapat ditemukan dalam jutaan *handphone* dan berbagai perangkat bergerak, membuat *Android* menjadi *platform* utama untuk pengembang *aplikasi*.

Sudah ada banyak *platformmobile* di pasar saat ini, termasuk *Symbian*, *iPhone*, *Windows*

Mobile, *BlackBerry*, *Java Mobile Edition*, *Linux Mobile (LiMo)*, dan banyak lagi. Meskipun beberapa fitur-fiturnya telah muncul sebelumnya, *Android* adalah *platform* pertama yang menggabungkan beberapa hal berikut :

1. *Android* merupakan sebuah *platform* yang berbasis *Linux* dan *open source*. Pembuat *handset* menyukai hal ini karena mereka dapat menggunakan dan menyesuaikan *platform* tanpa membayar *royalti*.
2. Sebuah arsitektur berbasis komponen. Bagian dari aplikasi *Android* dapat digunakan sebagai bahan lain yang bahkan tidak dibayangkan oleh *developer*. Kita dapat menggantikan *built-in* komponen aplikasi *Android* dengan versi pengembangan sendiri.
3. Banyak *built-in service* yang tidak biasa. Servis berdasarkan lokasi menggunakan GPS atau *cell tower triangulation* yang membuat pengalaman pemakai terjadi bergantung lokasi.

Android menyediakan jalur yang segar dalam aplikasi *mobile* berinteraksi dengan pemakai, bersama dengan teknik yang mendasar untuk membuatnya mungkin. Tetapi hal yang paling menarik dalam *Android* adalah kita dapat menulis sendiri aplikasinya [1].

Android adalah sistem operasi untuk *mobile device* yang awalnya dikembangkan oleh *Android Inc.* Perusahaan ini kemudian dibeli oleh *Google* pada tahun 2005. *Android* dibuat berdasarkan kernel *Linux* yang dimodifikasi. Aplikasi *Android* ditulis dengan bahasa *Java*, menggunakan *Java Core Libraries*.

B. *Bluetooth*

Bluetooth adalah Sebuah teknologi *wireless* yang mampu menyediakan layanan komunikasi data dan suara dengan jarak jangkauan yang terbatas. *Bluetooth* adalah sebuah teknologi

komunikasi *wireless* (tanpa kabel) yang beroperasi dalam pita *frekuensi* 2,4 GHz *unlicensed ISM (Industrial, Scientific and Medical)* dengan menggunakan sebuah *frequency hopping tranceiver* yang mampu menyediakan layanan komunikasi data dan suara secara *real-time* antara *host bluetooth* dengan jarak jangkauan layanan yang terbatas.

Bluetooth V3 keluaran dari DFRobot yang digunakan sebagai perantara komunikasi *bluetooth* antara *microcontroller* dengan *handphone* berbasis *Android*. Komunikasi *bluetooth* yang digunakan memiliki karakteristik sebagai berikut :

1. Kekuatan pengiriman sinyal sebesar 4dBm (maksimum).
2. Menggunakan tipe komunikasi serial dengan jangkauan komunikasi pada lapangan terbuka 10 meter.
3. *Bluetooth V3* dapat bekerja dengan baik pada temperature 0° hingga 70°C.
4. *Supply Power* sebesar 5 hingga 12 VDC.
5. Konsumsi arus sebesar 3mA hingga 35mA (bergantung pada kondisi koneksi dan *Baud Rate*).
6. Tidak dapat menggunakan kode enkripsi dan sistem *security* yang lain (misalnya *Passkey*).

Bluetooth V3 memiliki 2 mode operasi utama yaitu *command mode* dan *data mode*. Setiap kali dilakukan *power up*, *bluetooth V3* akan selalu masuk dalam *command mode* dan siap untuk menerima *serial command*. Pada *command mode* ini terdapat beberapa perintah yang dapat dikirim untuk menggunakan berbagai macam fitur yang dimiliki oleh *bluetooth V3*. Adapun gambar *bluetooth V3* dapat dilihat pada gambar dibawah ini.



Gambar 1. *Bluetooth V3*

Perintah-perintah itu antara lain adalah : *get con <CR>* (notasi *<CR>* merupakan 0Ah dan 0Dh) : perintah untuk melihat kondisi *connectable* dari *Bluetooth V3* yang digunakan, *get dis <CR>*. Perintah untuk melihat kondisi *discoverable* dari *bluetooth V3* yang digunakan untuk mencari *bluetooth* lain yang masuk dalam jangkauan *bluetooth V3* dan dalam keadaan *discoverable on*, *set dis status <CR>*. Perintah untuk merubah kondisi *discoverable* dari *bluetooth V3* variabel status diisi *on* (aktif) atau *off* (non aktif).

Tabel 1. Konfigurasi Pin *Bluetooth V3*

Nama Pin	Type Pin	Keterangan
VSS	+ 5V	Mengambil daya dari catu daya
GND	GND	Menghubungkan pada port D
RX	TX	USART data output
TX	TX	USART data input

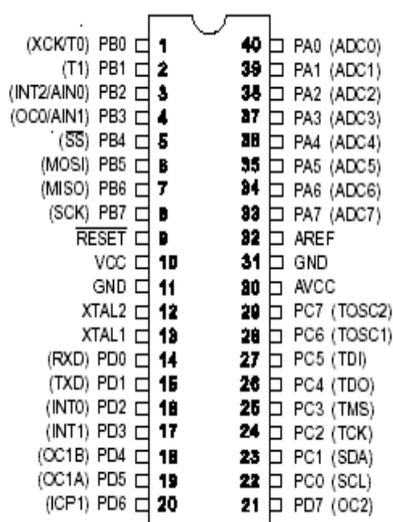
C. Mikrokontroler Atmega16

Mikrokontroler merupakan suatu alat elektronika digital yang mempunyai masukan dan keluaran serta kendali dengan program yang bisa ditulis dan dihapus dengan cara khusus, cara kerja mikrokontroler sebenarnya membaca dan menulis data [2].

Mikrokontroler yang biasa digunakan pada *board Arduino*. *Atmega32* merupakan mikrokontroler keluarga AVR 8 bit. Beberapa tipe mikrokontroler yang sama dengan Atmega 8 ini antara lain *Atmega8535*, *Atmega16*, *Atmega32*,

Atmega 328, yang membedakan antara mikrokontroler antara lain adalah, ukuran memori, banyaknya GPIO (pin *input/output*), peripheral (USART, timer, counter, dan lain-lain). Dari segi ukuran fisik, Atmega328 memiliki ukuran fisik lebih kecil dibandingkan dengan beberapa mikrokontroler di atasnya.

Namun untuk segi memori dan peripheral lainnya Atmega328 tidak kalah dengan yang lainnya karena ukuran memori dan peripheral relatif sama dengan Atmega8535, Atmega32, hanya saja jumlah GPIO lebih sedikit dibandingkan mikrokontroler di atasnya. Pin out IC mikrokontroler Atmega16 yang berpackage DIP dapat dilihat pada gambar 2.



Gambar 2. Pin Mikrokontroler ATmega16

Penjelasan dari masing-masing pin mikrokontroler Atmega 16 adalah sebagai berikut :

1. VCC merupakan pin yang berfungsi sebagai pin masukkan catu daya.
2. GND merupakan pin ground yang terhubung pada port D.
3. Port A (PA7...PA0) merupakan terminal masukkan analog A/D converter. Port ini juga berfungsi sebagai port I/O 8 bit dua

arah (*bidirectional*), jika A/D converter tidak diaktifkan.

4. Port B (PB7...PB0) merupakan port I/O 8 bit dua arah (*bidirectional*) dengan resistor pull-up internal. Port B juga dapat berfungsi sebagai terminal khusus yaitu timer/counter, komparator analog, dan SPI.
5. Port C (PC7...PC0) merupakan port I/O 8 bit dua arah (*bidirectional*) dengan resistor pull-up internal. Port C juga dapat berfungsi sebagai terminal khusus yaitu komparator analog, dan timer oscillator.
6. Port D (PD7...PD0) merupakan port I/O 8 bit dua arah (*bidirectional*) dengan resistor pull-up internal. Port D juga dapat berfungsi sebagai terminal khusus yaitu komparator analog, interupsi internal, dan komunikasi serial.
7. Reset merupakan pin yang digunakan untuk me-reset mikrokontroler.
8. XTAL1 dan XTAL2 merupakan pin masukkan clock eksternal.
9. AVCC merupakan pin masukkan tegangan untuk ADC.
10. AREF merupakan pin masukkan tegangan referensi ADC.

Atmega16 memiliki 3 buah port utama yaitu port B, port C, dan port D dengan total pin input/output sebanyak 23 pin. Port tersebut dapat difungsikan sebagai input/output digital atau difungsikan sebagai peripheral lainnya :

1. Port B

Port B merupakan jalur data 8 bit yang dapat difungsikan sebagai input/output. Selain itu port B juga dapat memiliki fungsi alternatif seperti di bawah ini.

- a. ICP1 (PB0), berfungsi sebagai Timer Counter 1 input capture pin

- b. OC1A (PB1), OC1B (PB2) dan OC2 (PB3) dapat difungsikan sebagai keluaran PWM (*pulse width modulation*).
 - c. MOSI (PB3), MISO (PB4), SCK (PB5), SS (PB2) merupakan jalur komunikasi SPI.
 - d. Selain itu pin ini juga berfungsi sebagai jalur pemrograman serial (ISP).
 - e. TOSC1 (PB6) dan TOSC2 (PB7) dapat difungsikan sebagai sumber *clock external* untuk *timer*.
 - f. XTAL1 (PB6) dan XTAL2 (PB7) merupakan sumber *clock* utama mikrokontroler.
2. *Port C*
- Port C* merupakan jalur data 7 bit yang dapat difungsikan sebagai *input/output* digital. Fungsi alternatif *port C* antara lain sebagai berikut :
- a. ADC6 *channel* (PC0,PC1,PC2,PC3,PC4,PC5) dengan resolusi sebesar 10 bit. ADC dapat kita gunakan untuk mengubah *input* yang berupa tegangan analog menjadi data digital. I2C (SDA dan SDL) merupakan salah satu fitur yang terdapat pada *port C*.
 - b. I2C digunakan untuk komunikasi dengan *sensor* atau *device* lain yang memiliki komunikasi data tipe I2C seperti sensor kompas, *accelerometer nunchuck*.
3. *Port D*
- Port D* merupakan jalur data 8 bit yang masing-masing pin-nya juga dapat difungsikan sebagai *input/output*. Sama seperti *Port B* dan *Port C*, *Port D* juga memiliki fungsi alternatif di bawah ini :
- a. USART (TXD dan RXD) merupakan jalur data komunikasi serial dengan level sinyal TTL. Pin TXD berfungsi untuk mengirimkan data serial, sedangkan RXD kebalikannya yaitu sebagai pin yang berfungsi untuk menerima data serial.
 - b. *Interrupt* (INT0 dan INT1) merupakan pin dengan fungsi khusus sebagai *interupsi hardware*. *Interupsi* biasanya digunakan sebagai selaan dari program, misalkan pada saat program berjalan kemudian terjadi *interupsi hardware/software* maka program utama akan berhenti dan akan menjalankan program *interupsi*.
 - c. XCK dapat difungsikan sebagai sumber *clock external* untuk USART, namun kita juga dapat memanfaatkan *clock* dari CPU, sehingga tidak perlu membutuhkan *external clock*.
 - d. T0 dan T1 berfungsi sebagai masukan *counter external* untuk *timer 1* dan *timer 0*.
 - e. AIN0 dan AIN1 keduanya merupakan masukan *input* untuk analog *comparator*.
- D. *Arsitektur CPU Atmega16*
- Fungsi utama *CPU* adalah memastikan pengekseskuan instruksi dilakukan dengan benar. Oleh karena itu *CPU* harus dapat mengakses memori, melakukan kalkulasi, mengontrol *peripheral*, dan menangani *interupsi*. Ada 32 buah *General Purpose Register* yang membantu *ALU* bekerja. Untuk operasi aritmatika dan logika,

operand berasal dari dua buah *general register* dan hasil operasi ditulis kembali ke *register*. *Status and Control* berfungsi untuk menyimpan instruksi aritmatika yang baru saja dieksekusi.

Informasi ini berguna untuk mengubah alur program saat mengeksekusi operasi kondisional. Instruksi dijemput dari *flash memory*. Setiap *byte flash memory* memiliki alamat masing-masing. Alamat instruksi yang akan dieksekusi senantiasa disimpan program *counter*. Ketika terjadi interupsi atau pemanggilan rutin biasa, alamat di program *counter* disimpan terlebih dahulu *distack*. Alamat interupsi atau rutin kemudian ditulis ke program *counter*, instruksi kemudian dijemput dan dieksekusi. Ketika *CPU* telah selesai mengeksekusi rutin interupsi atau rutin biasa, alamat yang ada di *stack* dibaca dan ditulis kembali ke program *counter* [3].

E. Program memori

AVR Atmega16 memiliki ruang pengalamatan *memory data* dan *memory program* terpisah. *Memory data* terbagi menjadi 3 bagian, yaitu 32 buah *register* umum, 64 buah *register I/O*, dan 512 *byte SRAM internal*.

Register keperluan umum menempati *space data* pada alamat terbawah, yaitu \$00 sampai \$1F. Sementara itu, *register* khusus menangani *I/O* dan kontrol terhadap *mikrokontroler* menempati 64 alamat berikutnya, yaitu mulai dari \$20 hingga \$5F. *Register* tersebut merupakan *register* yang khusus digunakan untuk mengatur fungsi terhadap berbagai *peripheral mikrokontroler*, seperti kontrol *register*, *timer/counter*, fungsi-fungsi *I/O*, dan sebagainya. Alamat *memory* berikutnya digunakan untuk *SRAM 512 byte*, yaitu pada lokasi \$60 sampai dengan \$25F. konfigurasi *memory data* ditunjukkan pada gambar dibawah ini.

Register file		Data Address	Space
R0		\$0000	
R1		\$0001	
R2		\$0002	
...		...	
R29		\$001D	
R30		\$001E	
R31		\$001F	
I/O Register			
\$00		\$0020	
\$01		\$0021	
\$02		\$0022	
...		...	
\$3D		\$005D	
\$3E		\$005E	
\$3F		\$005F	
Internal SRAM			
		\$0060	
		\$0061	
		...	
		\$0025E	
		\$0025F	

Gambar 3. Konfigurasi *Memory data AVR Atmega*

Memory program yang terletak dalam *Flash PEROM* dalam *word* atau 2 *byte* karena setiap instruksi memiliki lebar 16-bit atau 32-bit. *AVR Atmega16* memiliki 4K*byte* *15-bit Flash PEROM* dengan alamat mulai dari \$000 sampai \$FFF. *AVR* tersebut memiliki 12-bit *programcounter (PC)* sehingga mampu mengamati isi *Flash*.

F. SRAM Memori

Atmega16 memiliki 2 Kilo *Byte SRAM*. Memori ini dipakai untuk menyimpan *variabel*. Tempat khusus di *SRAM* yang ditunjuk *register SP* disebut *stack*. *Stack* berfungsi untuk menyimpan nilai yang *push*.

G. EEPROM Data Memori

Atmega16 memiliki 1024 *byte data EEPROM*. Data di *EEPROM* tidak akan hilang walaupun *catu daya* ke sistem mati. Parameter sistem yang penting disimpan di *EEPROM*. Saat sistem pertama kali menyala parameter tersebut dibaca dan sistem diinisialisasi sesuai dengan nilai parameter tersebut.

H. Sistem Interupsi

Interupsi adalah kondisi yang membuat CPU berhenti dari rutinitas yang sedang dikerjakan (rutin utama) untuk mengerjakan rutin lain (rutin interupsi). AVR Atmega16 memiliki 21 sumber interupsi yang ditunjukkan pada tabel dibawah ini

Tabel 2. Tabel Interupsi

Vector No	Program Address ⁽²⁾	Source	Interrupt Definition
1	0x000 ₁₃	RESET	External Pin, Power-on Reset, Brown-out Reset and Watchdog Reset
2	0x001	INT0	External Interrupt Request 0
3	0x002	INT1	External Interrupt Request 1
4	0x003	TIMER2 COMP	Timer/Counter2 Compare Match
5	0x004	TIMER2 OVF	Timer/Counter2 Overflow
6	0x005	TIMER1 CAPT	Timer/Counter1 Capture Event
7	0x006	TIMER1 COMPA	Timer/Counter1 Compare Match A
8	0x007	TIMER1 COMPE	Timer/Counter1 Compare Match B
9	0x008	TIMER1 OVF	Timer/Counter1 Overflow
10	0x009	TIMER0 OVF	Timer/Counter0 Overflow
11	0x00A	SPI, STC	Serial Transfer Complete
12	0x00B	USART, RXC	USART, Rx Complete
13	0x00C	USART, UDRE	USART Data Register Empty
14	0x00D	USART, TXC	USART, Tx Complete
15	0x00E	ADC	ADC Conversion Complete
16	0x00F	EE_RDY	EEPROM Ready
17	0x010	ANA_COMP	Analog Comparator
18	0x011	I2C	Two-wire Serial Interface
19	0x012	INT2	External Interrupt Request 2
20	0x013	TIMER0 COMP	Timer/Counter0 Compare Match
21	0x014	SPM_RDY	Store Program Memory Ready

I. Port I/O

Inisialisasi Port berfungsi untuk memilih fungsi port sebagai input atau sebagai output. Pada konfigurasi port sebagai output dapat dipilih pada saat awal setelah reset kondisi port berlogika 1 atau 0, sedangkan pada konfigurasi port sebagai input terdapat dua pilihan yaitu kondisi pin input toggle state atau pull-up, maka sebaiknya dipilih pull up untuk memberi default pada input selalu berlogika 1. setiap port berjumlah 8 bit, konfigurasi dari port dapat diatur sesuai dengan kebutuhan. Pengaturan konfigurasi dapat dilakukan perbit, jadi dalam satu port dapat difungsikan sebagai input dan output dengan nilai defaultnya berbeda-beda, menunjukkan setting konfigurasi pada port a dengan kombinasi input dan output yang berbeda-beda.

Perintah dasar sistem input dan output adalah sebagai berikut [6] :

a. Output

PORTX = data;

Yaitu untuk mengirimkan data secara byte ke port X (X = A, B, C, D). perintah out dalam bahasa assembly.

b. Input

data_in = PINX;

Yaitu untuk mengambil data byte dari PINX (X = A, B, C, D) yang kemudian disimpan dalam variabel data_in. Perintah ini sama dengan in dalam bahasa assembly.

Sebelum memulai pemrograman dasar input dan output dengan bahasa C perlu diketahui bahwa mikrokontroler perlu disetting DDR dan PORT agar digunakan sebagaimana mestinya.

Tabel 3. Tabel Input dan Output

	DDR bit = 1	DDR bit = 0
PORT bit = 1	Output ; High	Input ; R pull-up
PORT bit = 0	Output ; Low	Input ; Floating

Misalnya :

```
...
PORTA=0xCC;
DDRA=0x0F;
```

Dari DDR terlihat bahwa port A0-3 sebagai output dan port A4-7 sebagai input, sedangkan dari port terlihat bahwa PA0-1 = Low, PA2-3 = High, PA4-5 sebagai Rpull-up (floating) dan PA6-7 dengan Rpull-up.

J. Clear Timer on Compare Match (CTC)

CTC adalah salah satu mode Timer/Counter1, selain itu ada Normal mode, Fast PWM mode, Phase Correct PWM mode. Pada CTC mode maka nilai TCNT1 menjadi nol jika nilai TCNT1 telah sama dengan OCR1A atau ICRT1. Jika nilai top ditentukan OCR1A dan interupsi diaktifkan untuk Compare Match A maka saat nilai TCNT1 sama dengan nilai OCR1A interupsi terjadi. CPU

melayani interupsi ini dan nilai *TCNT1* menjadi nol.

K. *USART*

Selain untuk general *I/O*, pin *PD1* dan *PD0* *ATMEGA32* berfungsi untuk mengirim dan menerima bit secara serial. Perubahan fungsi ini dibuat dengan mengubah nilai beberapa *register* serial. Untuk menekankan fungsi ini, pin *PD1* disebut *TxD* dan pin *PD0* disebut *RxD*. Nilai *UBRR* dan *clock* sistem menentukan laju bit pengirim dan penerima serial.

III. METODE PENELITIAN

A. Teknik Pengumpulan Data

Dalam mengumpulkan data, teknik pengumpulan data yang digunakan dalam penelitian ini yaitu: Pustaka, Observasi, Wawancara Dan Dokumentasi.

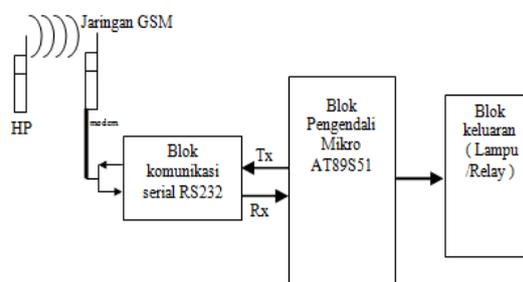
B. Metode Pengembangan Sistem

Metode yang digunakan untuk pengembangan sistem dalam penelitian ini adalah model proses *Prototype*. Model *prototype* (*Prototyping model*) merupakan suatu teknik atau proses untuk mengumpulkan informasi tertentu kemudian membangun sebuah model dari sebuah sistem berdasarkan pada kebutuhan pengguna, dengan kondisi pengguna tidak memberikan detail *input*, proses, dan *output*, dalam situasi seperti ini maupun situasi lain, paradigma *prototyping* bisa memberikan pendekatan terbaik, Model tersebut dapat berupa tiga bentuk:

1. Bentuk *prototype* di atas kertas atau model berbasis komputer yang menggambarkan interaksi manusia yang mungkin terjadi.
2. *Working prototype*, yang mengimplementasikan sebagian dari fungsi yang ditawarkan perangkat lunak.

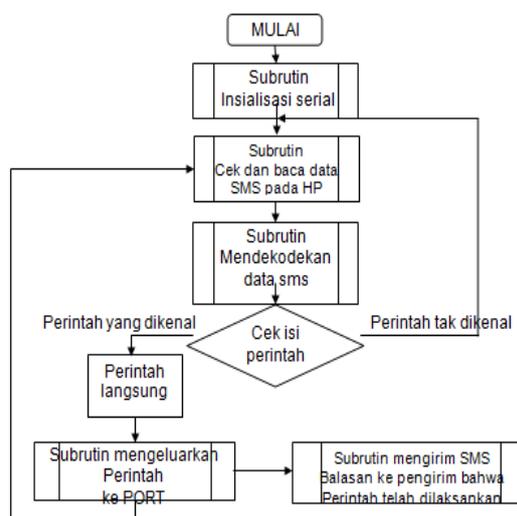
3. Program jadi yang melakukan sebagian atau seluruh fungsi yang akan dilakukan, tapi masih ada fitur yang masih dikembangkan

C. Prinsip Kerja Sistem



Gambar 4. Bagan Kotak Prinsip Kerja Sistem

D. Flowchart Perancangan Perangkat Lunak



Gambar 5. Flowchart Program Utama

IV. HASIL DAN PEMBAHASAN

A. Hasil

Implementasi Aplikasi *Remote* Kontrol *CPU/Laptop* Jarak Jauh Dengan Media Serial *Handphone* Dengan Mikrokontroler merupakan pengembangan sistem. Salah satu aspek pengembangan sistem aplikasi ini adalah aspek pengendalian dan kecepatan proses terkait dengan prosesor yang digunakan.

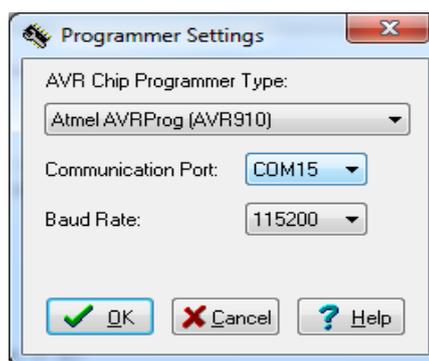
Mikrokontroler *Atmega16* sebuah rangkaian digital pada satu sirkuit terpadu yang berisi inti prosesor, memory dan *Input/Output*. Memori

program dalam bentuk *flash* atau *ROM* juga sering disertakan pada *chip*, serta jumlah yang kecil *RAM*.

Mikrokontroler merupakan sinyal campuran yang mengintegrasikan komponen analog yang diperlukan untuk sistem kontrol elektronik *non-digital*. Sedangkan prinsip kerja sebuah mikrokontroler dapat dijelaskan sebagai berikut :

1. Berdasarkan data yang ada pada *register* program *counter*. Mikrokontroler mengambil data dari *ROM* dengan alamat sebagaimana ditunjukkan dalam program *counter*. Selanjutnya program *counter* ditambah nilainya dengan 1 secara otomatis. Data yang diambil tersebut merupakan urutan instruksi program pengendali mikrokontroler yang sebelumnya telah dituliskan oleh *programmer*.
2. Instruksi diolah dan dijalankan. Proses pengerjaan bergantung pada jenis instruksi, bisa membaca, mengubah nilai-nilai dalam *register*, *RAM*, isi *port* atau melakukan pembacaan dan dilanjutkan dengan perubahan data.
3. Program *counter* telah berubah. Selanjutnya yang dilakukan mikrokontroler adalah mengulang kembali siklus ini pada langkah program *counter* dengan nilai 1.
- 4.

Kemudian dalam melakukan uji analisis sistem yang dipakai adalah *CodeVisionAVR C 2.05.3* dan *Microsoft Visual Studio 2008*. Sebelum menggunakan *CVAVR* sekaligus sebagai *downloader*, lakukan *setting* sebagai berikut :

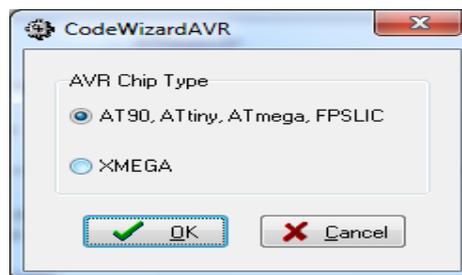


Gambar 6. Programmer Setting



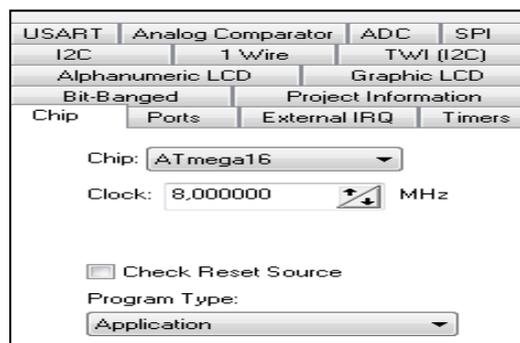
Gambar 7. Pembuatan New Project

Maka akan muncul tampilan *CodeWizardProject*, kemudian sesuaikan bagian yang diinginkan.



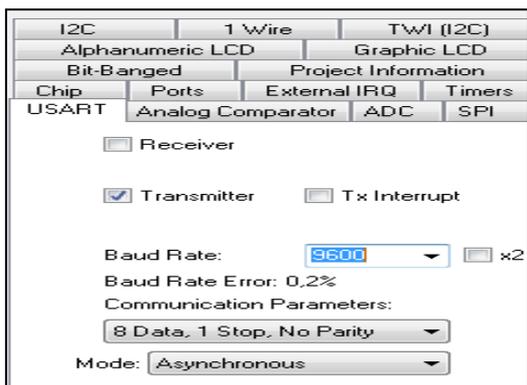
Gambar 8. Konfirmasi Code WizardAVR

Atur bagian *chip*, pilih *Atmega16* dan *clock* 8.000000 terlihat pada Gambar 9.



Gambar 9. Bagian Tab Chip

Atur bagian *USART*, beri *checklist* pada *receiver*, *rx interrupt*, dan *transmitter*. Terlihat pada Gambar 10



Gambar 10. Bagian Tab USART

B. Pembahasan

Dari skenario yang dibuat, diperoleh hasil pengujian sistem Aplikasi Remot Kontrol CPU/Laptop yaitu *bluetooth* adalah media perantara nirkabel. Data yang dikirim melalui dalam bentuk serial *asinkron*. Aplikasi terhubung dengan *bluetooth* secara serial *asinkron*. *Handphone* mengirim data secara serial, yang kemudian diteruskan ke *bluetooth* yang terhubung dengan mikrokontroler *atmega16*. aplikasi menerima data serial dari *bluetooth* melalui cara interupsi serial. Adapun tabel hasil pengujian Aplikasi Remote Kontrol CPU/Laptop sebagai berikut :

Tabel 4. Hasil Pengujian

No	Perintah	Kode ASCII	Jarak Bluetooth	Jumlah Percobaan	Persentase Keberhasilan
1	restart	/r	10 Meter	10	100%
2	Shutdown	/s	10 Meter	10	100%

Tabel 5. Hasil Pengujian Rintangan

No	Kode ASCII	Keterangan
1	/i	Tampilan antar muka harus menjadi pilihan pertama, tidak menggunakan opsi lain.
2	/l	Log Off, tidak dapat digunakan dengan pilihan /m atau /d berdasarkan Id Windows
3	/g	Shutdown dan restart komputer. Setelah sistem reboot, restart jika aplikasi terdaftar.
4	/a	Hal ini hanya dapat digunakan selama periode time-out.
5	/p	Matikan komputer lokal tanpa batas waktu atau warning. Jika digunakan dengan / d dan / opsi f. berdasarkan Id Windows
6	/h	Hibernate komputer lokal. Dapat digunakan dengan / f opsi berdasarkan Id Windows
7	/e	Dokumen untuk shutdown dari sebuah komputer.

Tabel 6 Hasil Pengujian Bluetooth

No	Jarak Bluetooth	Kendala Yang dihadapi	Keterangan
1	1 Meter	Tidak Ada	Sesuai Yang diinginkan
2	2 Meter	Tidak Ada	Sesuai Yang diinginkan
3	3 Meter	Tidak Ada	Sesuai Yang diinginkan
4	4 Meter	Tidak Ada	Sesuai Yang diinginkan
5	5 Meter	Tidak Ada	Sesuai Yang diinginkan
6	6 Meter	Tidak Ada	Sesuai Yang diinginkan
7	7 Meter	Tidak Ada	Sesuai Yang diinginkan
8	10 Meter	Tidak Ada	Sesuai Yang diinginkan
9	15 Meter	Koneksi Bluetooth Gagal	Tidak Sesuai
10	20 Meter	Koneksi Bluetooth Gagal	Tidak Sesuai

Tabel 7 Hasil Pengujian Rintangan

No	Jarak Bluetooth	Kendala Yang dihadapi	Keterangan
1	1 Meter	Tembok Kaca	Sesuai Yang diinginkan
2	3 Meter	Tembok Kaca dan Tembok Semen	Sesuai Yang diinginkan
3	4 Meter	Tembok Kaca dan 2 Tembok Semen	Sesuai Yang diinginkan
4	6 Meter	Tembok Kaca dan 3 Tembok Semen	Sesuai Yang diinginkan

V. KESIMPULAN DAN SARAN

A. Kesimpulan

Ada beberapa kesimpulan yang dapat diambil dari hasil pengujian sistem, yaitu :

1. Dalam melakukan pengujian, aplikasi memiliki batas jarak 10 Meter, jika lebih dari 10 Meter koneksi *bluetooth* gagal, karena sifat *bluetooth* memancarkan frekuensi, sehingga tidak ada halangan, kecuali hanya jarak jangkauan pada *bluetooth*.
2. Aplikasi yang digunakan harus *handphone* berbasis *Android*, dikarenakan ada *software* tambahan yaitu *bluterm*.
3. Dalam melakukan kriteria pengujian aplikasi kontrol jarak jauh menggunakan aplikasi *Code Vision AVR*, kelebihan *tools* pada *code wizard* dalam pengembangan sistem bisa lebih kompleks, tampilan editor *Code Vision AVR* lebih mudah dipahami oleh pemakai namun terkendala pada *licensed software* yang cukup mahal.

B. Saran

Peneliti menyadari masih banyak kekurangan oleh karena itu berdasarkan masalah yang ditemukan dalam penelitian, diharapkan implementasi aplikasi kontrol jarak jauh bisa menggunakan sms gateway sehingga aplikasi dapat menggunakan *handphone* apa saja dan aplikasi kontrol jarak jauh menggunakan

mikrokontroler *Atmega32* dapat dikembangkan lebih komplek lagi.

REFERENSI

- [1] Immanuel, Alpha, 2010, *Pembuatan Aplikasi Pengontrol Robot Berbasis Android*.
- [2] Sumardi, 2013, *Mikrokontroler Belajar AVR Mulai Dari Nol*, Graha Ilmu, Yogyakarta.
- [3] Syahid, 2012, *Rancang Bangun Robot yang Memiliki Lima Derajat Kebebasan*, jurnal Ilmiah Foristel Vol. 1, No 1, Maret 2011. Aplikasi Pengontrol Robot Berbasis Android'.