

RANCANG BANGUN SISTEM PENGANTI KUNCI KONTAK KENDARAAN BERMOTOR RODA DUA BERBASIS *BUETOOTH*

Deddy Kurniawan, Sulisty MB, Anang Budikarso
Politeknik Elektronika Negeri Surabaya, Jurusan Teknik Telekomunikasi
Institut Teknologi Sepuluh Nopember
Kampus ITS Sukolilo, Surabaya 60111
Email : ddy_47@yahoo.co.id

Abstrak

Pesatnya perkembangan kendaraan roda dua di kota-kota besar berdampak pada semakin tingginya angka kehilangan atau curanmor. Selain itu penggunaan kunci pengaman tambahan yang repot menyebabkannya kurang populer di masyarakat.

Pada proyek akhir ini dibuat sistem kontrol kelistrikan kendaraan roda dua pengganti kunci kontak. Melainkan menggantinya secara *wireless* menggunakan *easy bluetooth*.

Dalam sistem ini relay bekerja bila mendapatkan logika 0 (nol) yang menilai 5 volt dari mikrokontroler. sehingga mengaktifkan sistem yang dikontrol yaitu sistem kelistrikan dan sistem pengapian.

Kata Kunci :

Microprosesor AVR, Bluetooth, Relay

1. Pendahuluan

Pesatnya pertumbuhan kendaraan roda dua baik pada kota-kota besar maupun pada daerah-daerah atau desa berbanding lurus dengan kenaikan angka kehilangan pada kendaraan roda dua hal ini juga dipicu oleh dampak dari era globalisasi yang menuntut seseorang untuk bekerja dengan cepat sehingga membuat seseorang menjadi lebih ceroboh yaitu lupa untuk mencabut kunci padakontak kendaraan bermotornya sehingga hilangnya kendaraan tidak terelakan.

Pada umumnya saat ini sudah banyak masyarakat yang menggunakan peralatan tambahan guna meningkatkan tingkat keamanan pada kendaraanya, akan tetapi karena penggunaannya yang tidak bisa dibbilang mudah sehingga membuat banyak orang yang mengindahkan untuk menggunakannya. Teknologi *bluetooth* yang memungkinkan kita agar bisa bertukar data maupun informasi dari satu perangkat ke perangkat yang lain tanpa memerlukan bantuan dari bantuan operator, tetapi dengan jangkauan koneksi yang terbatas. *Bluetooth* merupakan sebuah teknologi komunikasi *wireless* (tanpa kabel) yang beroperasi dalam pita frekuensi 2,4 GHz unlicensed ISM (Industrial, Scientific and Medical) dengan menggunakan sebuah *frekuensi hopping tranceiver* yang mampu menyediakan layanan komunikasi data secara real-time antara host-host *bluetooth* dengan jarak jangkauan yang terbatas.

Pada dasarnya *bluetooth* diciptakan bukan hanya untuk menggantikan atau menghilangkan

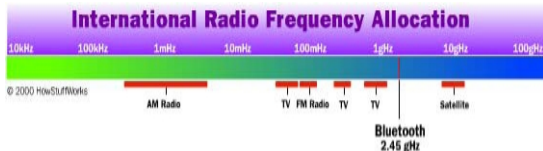
penggunaan kabel di dalam melakukan pertukaran informasi, tetapi juga mampu menawarkan fitur yang baik untuk teknologi *wireless* dengan biaya yang relatif murah. Cukup banyak orang mulai dari kalangan anak muda hingga para eksekutif telah menggunakan teknologi nirkabel layanan tanpa bayar *bluetooth* kapan saja dan dimana saja. Berdasarkan hal-hal tersebut di atas, memunculkan gagasan bagi penulis untuk membuat sebuah switch kontrol sebagai pengganti kunci kontak dengan memanfaatkan teknologi *bluetooth*. Di dalam aplikasinya, user dapat mengontrol switch kunci kontak dengan mudah melalui perangkat yang memanfaatkan konektifitas dari dua *bluetooth*.

2. Teori Penunjang

2.1 Bluetooth

Bluetooth merupakan sebuah teknologi komunikasi *wireless* (tanpa kabel) yang beroperasi dalam pita frekuensi 2,4 GHz unlicensed ISM (Industrial, Scientific and Medical) dengan menggunakan sebuah *frekuensi hopping tranceiver* yang mampu menyediakan layanan komunikasi data dan suara secara real-time antara host-host *Bluetooth* dengan jarak jangkauan layanan yang terbatas. *Bluetooth* sendiri dapat berupa card yang bentuk dan fungsinya hamper sama dengan card yang digunakan untuk *wireless* local area network (WLAN) dimana menggunakan frekuensi radio standar IEEE 802.11, hanya saja pada *Bluetooth* mempunyai jangkauan jarak layanan yang lebih pendek dan kemampuan transfer data yang lebih rendah.

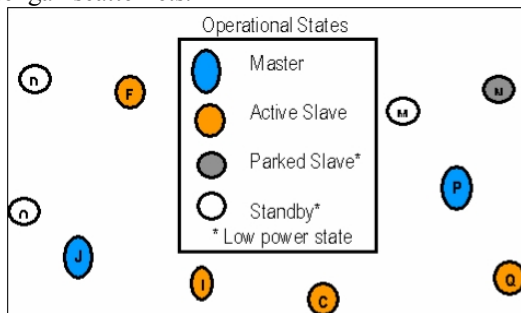
Bluetooth terdiri dari microchip radio penerima/pemancar yang sangat kecil/pipih dan beroperasi pada pita frekuensi standar global 2,4 GHz. Teknologi ini menyesuaikan daya pancar radio sesuai dengan kebutuhan. Ketika radio pemancar mentransmisikan informasi pada jarak tertentu, radio penerima akan melakukan modifikasi sinyal-sinyal sesuai dengan jarak yang selaras sehingga terjadi fine tuning. Data yang ditransmisikan oleh chipset pemancar akan diacak, diproteksi melalui inskripsi serta otentifikasi dan diterima oleh chipset yang berada di peralatan yang dituju.



Gambar 1. Alokasi Frekuensi Radio

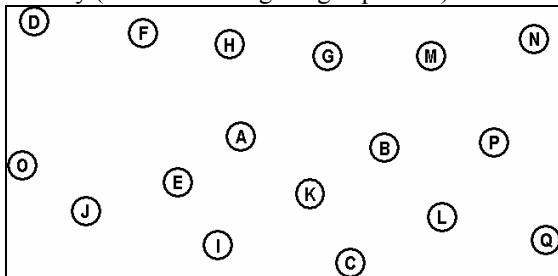
2.1.1 Cara Kerja Bluetooth

Pada Gambar 2. menunjukkan bagaimana *Bluetooth* device melakukan koneksi ke dalam piconet. Piconet terdiri dari sebuah master device dan active slave devices, dimana jumlah maksimum active slaves adalah 7. Kumpulan dari beberapa piconet yang saling berhubungan disebut dengan scatternets.



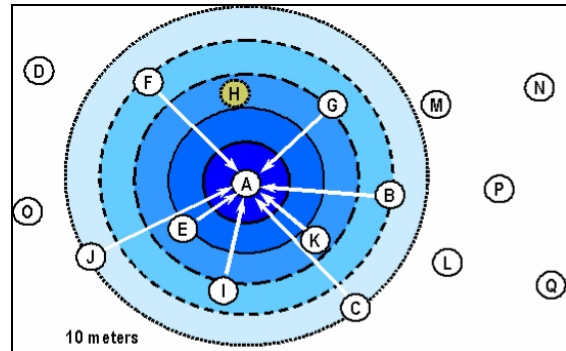
Gambar 2. Operational State of Bluetooth

Bluetooth devices mempunyai 4 basic states. Antara lain adalah master (yang mengontrol sebuah piconet), active slave (terhubung dan secara aktif memonitor Piconet), parked slave (secara logik masih bagian dari piconet tetapi low power,) dan standby (tidak terhubung dengan piconet).



Gambar 3. Bluetooth pada Awalnya

Bluetooth device pada awalnya hanya mengetahui hanya sekitar diri mereka dan di dalam status ini mereka akan berada di mode Standby. Standby adalah suatu mode pasif di mana suatu *Bluetooth* device sekali-kali mendengarkan jika ada *bluetooth* device lain yang ingin berkomunikasi, hal ini disebut Inquiry. Proses ini dilakukan selama 10 miliseconds tiap 1.28 detik. Di dalam mode Standby *Bluetooth* device dapat mengurangi konsumsi kekuatannya atas 98%.

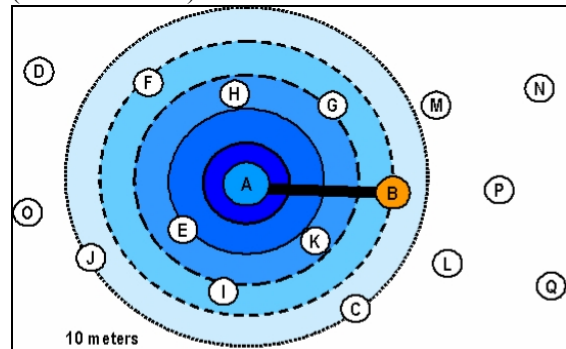


Gambar 4. Proses Inquiry Bluetooth

Inquiry adalah suatu proses bagaimana *Bluetooth* device belajar tentang *bluetooth* devices lain yang ada di dalam jangkauannya. Pada Gambar 4, node A mengeluarkan fungsi page pada BT Inquiry ID dan menerima balasan dari devices B, C, E, F, G, I, J, and K.

Dari balasan ini, A mengetahui identitas dari devices lain (contohnya, *Bluetooth* device ID mereka yang unik).

Selama proses inquiry, device A secara terus-menerus melakukan broadcasts Page command dengan menggunakan reserved Inquiry ID. Broadcasts ini tersebar sepanjang pola standard dari 32 Standby radio frequencies dimana semua devices pada mode standby memonitor pada sebuah occasional basis. Kemudian setiap standby device dalam jangkauannya akan menerima inquiry page. Dengan melakukan persetujuan, node-node ini akan merespon dengan sebuah standar FHS packet yang menyediakan BT ID-nya yang unik dan clock offset-nya. Node H pada Gambar 4 menunjukkan bagaimana sebuah *Bluetooth* device dapat diprogram sebagai anonymous (Undiscoverable).



Gambar 5. Proses Paging Bluetooth

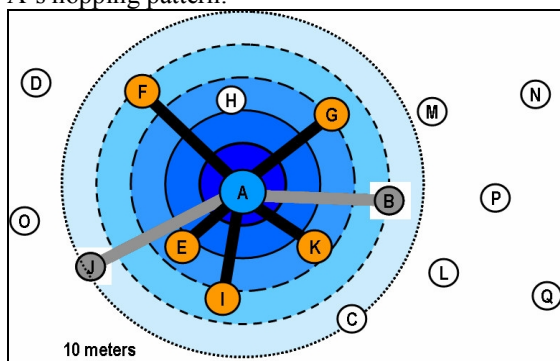
Setelah proses inquiry, akan dilakukan proses paging, dimana pada proses ini akan dibangun hubungan antar device (antar master sebagai pemula dengan sebuah slave. Hubungan master/slave pada *bluetooth* dikenal dengan sebutan piconet.

Untuk menciptakan piconet, device A melakukan broadcasts Page command dengan explicit device ID darislave target (pada gambar di atas adalah B) yang telahsiapa. Semua

bluetooth devices kecuali B akan mengabaikan perintah ini karena tidak ditujukan padanya.

Ketika device B membalas, device A akan mengirim sebuah FHS packet kembali dan menentukannya sebagai Active Member Address pada Piconet. Sebagai active slave, device B mulai memonitor secara terus-menerus perintah dari selanjutnya dari device A.

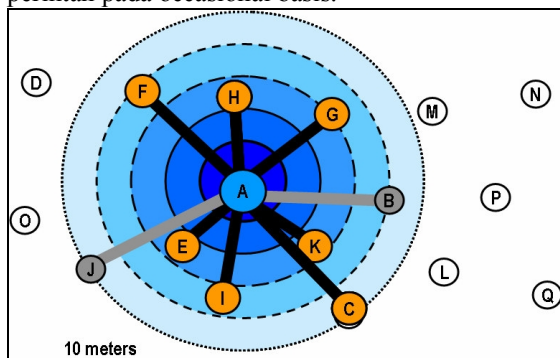
Sebuah *bluetooth* master dapat melakukan proses paging ini dengan maksimum 7 active slaves. 7 merupakan batas atas karena hanya disediakan 3 bits pada *Bluetooth* untuk Active Member Address (AMA) dengan 000 disediakan untuk master dan sisanya untuk slaves. Sekali lagi, semua active slaves ke A akan memonitor secara terus-menerus untuk perintah yang ditujukan kepada mereka dalam sinkronisasi dengan device A's hopping pattern.



Gambar6. Proses Parking

Parking merupakan mekanisme yang memungkinkan *Bluetooth* Master untuk berhubungan dengan 256 devices tambahan. 256 adalah batas atas karena disediakan 8 bits pada *Bluetooth* untuk Parked Member Address (PMA).

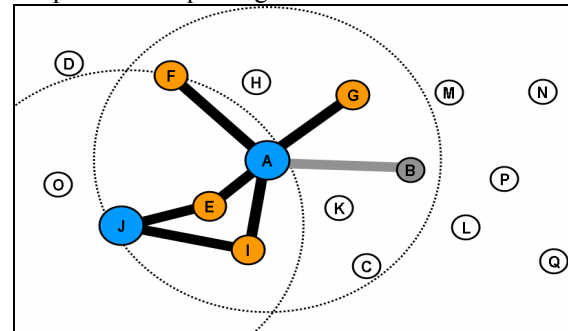
Untuk memarkirkan sebuah device, *Bluetooth* Master mengeluarkan Park command terhadap sebuah active slave dan menentukannya sebagai PMA. Slave ini kemudian memasuki mode parked dan menyerahkan AMA-nya. Sebagai sebuah parked slave, device akan berubah ke dalam mode passive dan hanya memonitor perintah-perintah pada occasional basis.



Gambar7. Proses Mengembangkan Piconet

Dengan adanya Active Member Addresses yang dilepaskan oleh sebuah active slaves, *Bluetooth* Master dapat melakukan proses paging dengan device lain untuk menjadi Active Slaves.

Pada Gambar 7, device A menambahkan H dan C ke dalam piconet-nya dengan AMAs yang dilepaskan oleh parking nodes B dan J.

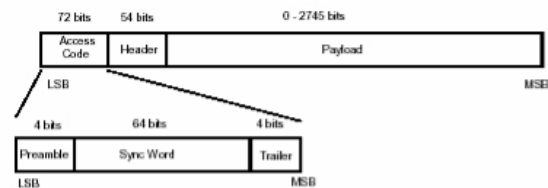


Gambar8 Scatternet

Tiap *bluetooth* node dapat menjadi bagian dari beberapa piconets sekaligus dalam satu waktu. Hal ini membuat beberapa piconets dapat bergabung membentuk sebuah struktur yang disebut scatternet. Pada Gambar 8, dua piconets bergabung menjadi sebuah scatternet melalui slaves E dan I.

2.1.2 Struktur Frame Data *Bluetooth*

Struktur frame data dari *Bluetooth* dapat dilihat pada gambar berikut.



Gambar9. Struktur Frame Data *Bluetooth*

Channel Access Code (CAC): mengidentifikasi sebuah piconet, kode ini digunakan dengan semua traffic exchanged pada sebuah piconet.

Device Access Code (DAC): Digunakan untuk signaling, seperti paging dan respon terhadap paging.

Inquiry Access Code (IAC):

- ❖ General Inquiry Access Code (GIAC), umum untuk semua *bluetooth* devices.
- ❖ Dedicated Inquiry Access Code (DIAC), umum untuk sebuah kelas dari *Bluetooth* devices.
- ❖ Inquiry process "finds" BT devices dalam range.

Packet Header

AM_ADDR: 3 bit alamat member menunjukkan active members dari sebuah piconet.

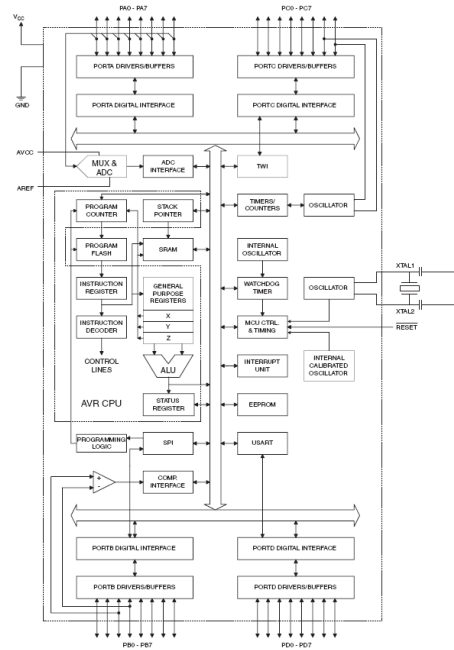
Data Type: Menunjukkan bermacam-macam tipe paket dan panjangnya. Memperbolehkan non-addressed slaves untuk menentukan kapan mereka dapat transmit.

Flow Control

Acknowledgement: ACK/NAK field HEC: header error check, jika error ditemukan, keseluruhan paket dibuang.

3.2 Mikrokontroler AVR ATmega8535

Atmel, salah satu vendor yang bergerak di bidang mikroelektronika, telah mengembangkan AVR (Alf and Vegard's Risc prosesor) sekitar tahun 1997. Berbeda dengan MCS 51, AVR menggunakan arsitektur RISC (Reduced Instruction Set Computing) yang mempunyai lebar bus data 8 bit. Perbedaan ini bisa dilihat dari frekuensi kerjanya. MCS 51 memiliki frekuensi kerja seperduabelas kali frekuensi osilator. Jadi dengan frekuensi osilator yang sama, kecepatan AVR dua belas kali lebih cepat dibanding kecepatan MCS 51. Secara umum AVR dapat dikelompokkan menjadi 4 kelas, yaitu keluarga ATtiny, keluarga AT90Sxx, keluarga ATmega dan AT86RFxx. Perbedaan antartipe AVR terletak pada fitur-fitur yang ditawarkan, sementara dari segi arsitektur dan set instruksi yang digunakan hampir sama[8].



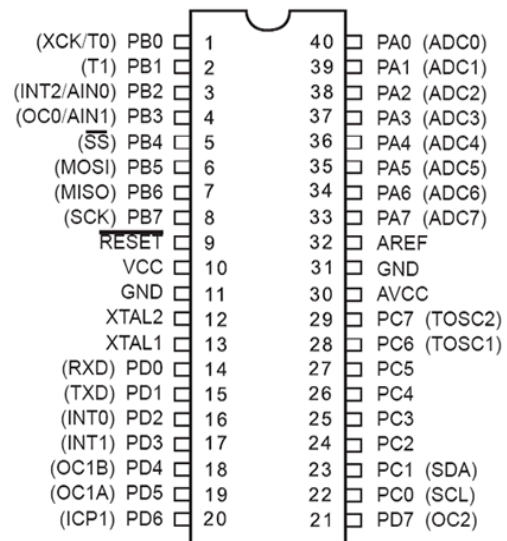
Gambar 10 Arsitektur ATmega8535

2.2.1 Arsitektur ATmega8535

Fitur

1. 8 bit AVR berbasis RISC dengan performa tinggi dan konsumsi daya rendah
2. Kecepatan maksimal 16 MHz
3. Memori
 - a. 8 Kb Flash
 - b. 512 byte SRAM
 - c. 512 byte EEPROM
4. Timer/Counter
 - a. 2 buah 8 bit Timer/Counter
 - b. 1 buah 16 bit Timer/Counter
 - c. 4 kanal PWM
5. 8 kanal 10/8 ADC
6. Programable serial USART
7. Komparator analog
8. 6 pilihan sleep mode untuk penghematan daya listrik
9. 32 jalur I/O yang bisa diprogram

2.2.2 Konfigurasi Pin ATmega8535



Gambar 11 Konfigurasi pin atmega8535

Dari gambar konfigurasi pin ATmega tersebut, dapat dijelaskan secara fungsional konfigurasi pin ATmega8535 sebagai berikut:

1. VCC merupakan pin yang berfungsi sebagai pin masukan catu daya.
2. GND merupakan pin ground
3. Port A (PA0..PA7) merupakan pin I/O dua arah dan pin masukan ADC.
4. Port B (PB0..PB7) merupakan pin I/O dua arah dan pin fungsi khusus, yaitu Timer/Counter, komparator analog, dan SPI.

5. Port C (PC0..PC7) merupakan pin I/O dua arah dan pin fungsi khusus, yaitu TWI, komparator analog dan Timer Oscillator.
6. Port D (PD0..PD7) merupakan pin I/O dua arah dan pin fungsi khusus, yaitu komparator analog, interupsi eksternal, dan komunikasi serial.
7. RESET merupakan pin yang digunakan untuk me-reset mikrokontroler.
8. XTAL1 dan XTAL2 merupakan pin masukan clock eksternal.
9. AVCC merupakan pin masukan tegangan untuk ADC.
10. AREF merupakan pin masukan tegangan referensi ADC.

3.3 Relay

Dalam dunia elektronika, relay dikenal sebagai komponen yang dapat mengimplementasikan logika switching. Sebelum tahun 70an, relay merupakan "otak" dari rangkaian pengendali. Baru setelah itu muncul PLC yang mulai menggantikan posisi relay. Relay yang paling sederhana ialah relay elektromekanis yang memberikan pergerakan mekanis saat mendapatkan energi listrik. Secara sederhana relay elektromekanis ini didefinisikan sebagai berikut :

- ❖ Alat yang menggunakan gaya elektromagnetik untuk menutup (atau membuka) kontak saklar.
- ❖ Saklar yang digerakkan (secara mekanis) oleh daya/energi listrik[7].

3.4 Bahasa C

Akar dari bahasa C adalah bahasa BCPL yang dikembangkan oleh Martin Richards pada tahun 1967. Bahasa ini memberikan ide kepada Ken Thompson yang kemudian mengembangkan bahasa yang disebut dengan B pada tahun 1970. Perkembangan selanjutnya dari bahasa B adalah bahasa C oleh Dennis Ritchie sekitar tahun 1970-an di Bell Telephone Laboratories Inc. (sekarang adalah AT&T Bell Laboratories). Bahasa C pertama kali digunakan pada komputer Digital Equipment Corporation PDP-11 yang menggunakan sistem operasi UNIX[9].

Standar bahasa C yang asli adalah standar dari UNIX. Sistem operasi, kompilasi C dan seluruh program aplikasi UNIX yang esensial ditulis dalam bahasa C. Kepopuleran bahasa C membuat versi-versi dari bahasa ini banyak dibuat untuk komputer mikro. Untuk membuat versi-versi tersebut menjadi standar, ANSI (American National Standards Institute) membentuk suatu komite (ANSI committee X3J11) pada tahun 1983 yang kemudian menetapkan standar ANSI untuk bahasa C. Standar ANSI ini didasarkan kepada standar UNIX yang diperluas.

Struktur penulisan bahasa C secara umum terdiri atas empat blok, yaitu[9]:

1. Header,
2. Deklarasi konstanta global dan atau variable,
3. Fungsi dan atau prosedur,
4. Program utama.

3.5 CodeVision AVR

Code vision AVR merupakan software untuk membuat code program microcontroller AVR. Software ini bisa kita download versi demonya di www.hpinfotech.com kebanyakan programmer memakai software ini karena fasilitas-fasilitas yang disediakan CodeVision AVR memudahkan programmer dalam membuat code.

Code vision AVR C Compiler (CVAVR) merupakan kompilasi bahasa C untuk AVR. Kompilasi ini cukup memadai untuk belajar AVR, karena selain mudah penggunaannya juga didukung berbagai fitur yang sangat membantu dalam pembuatan software untuk keperluan pemrograman AVR.

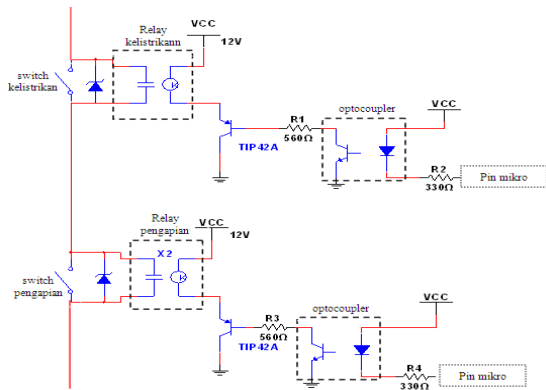
CVAVR ini dapat berjalan di bawah sistem operasi Windows 9x, Me, NT 4, 2000 dan XP. CVAVR ini dapat mengimplementasikan hampir semua instruksi bahasa C yang sesuai dengan arsitektur AVR, bahkan terdapat beberapa keunggulan tambahan untuk memenuhi keunggulan spesifik dari AVR. Hasil kompilasi objek CVAVR bisa digunakan sebagai source debug dengan AVR Studio debugger dari ATMEL[6].

3. Pengujian Sistem

3.1 Rangkaian Driver

Rangkaian catu daya memberikan supply tegangan pada alat pengendali. Rangkaian catu daya memanfaatkan sumber tegangan dari ACCU sepeda motor.

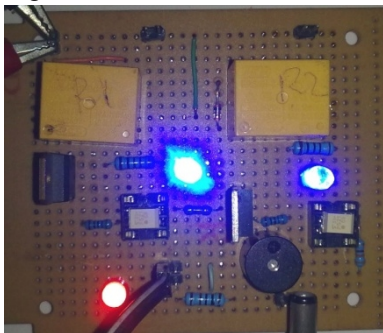
Tegangan dari ACCU sepeda motor ini kemudian diturunkan menjadi 5 VDC saat memasuki mikrokontroler. Keluaran dari port mikrokontroler ini kemudian masuk ke optocoupler yang fungsinya hampir sama dengan photo diode yaitu akan menekan switch (ada dalam perangkat) untuk menyambungkan arus base dari transistor TIP42 ke ground setelah menerima arus/perintah dari port mikrokontroler. Tegangan inilah yang dipakai untuk mengaktifkan lilitan relay sehingga membuatnya bekerja (dari NC/Normaly Close ke NO/Normaly Open). Diode zener yang terdapat pada relay digunakan untuk memblokir serta untuk pembatas tegangan.



Gambar12rangkaian driver relay

3.1 PengujianRangkaian Driver

Pengujian dilakukan dengan memberi logika 1 atau senilai tegangan 5 Volt dari port microcontroller yang digunakan, dari pengujian ini akan diketahui apakah rangkaian driver telah tersusun dengan benar atau tidak.



Gambar13Pengujian Rangkaian Driver

4. Pengujiankoneksi

Table 1 tanpa halangan

Jarak pengujian (meter)	Koneksi	keterangan
1	Oke	Ada respon
2	Oke	Ada respon
4	Oke	Ada respon
6	Oke	Ada respon
8	Oke	Ada respon
10	Oke	Ada respon
12	Oke	Ada respon
14	Oke	Ada respon
16	Gagal	Tidak ada respon

Table 2 dengan halangan

Jarak pengujian (meter)	Koneksi	keterangan
1	Oke	Ada respon
2	Oke	Ada respon
4	Oke	Ada respon
6	Oke	Ada respon
8	Oke	Ada respon
10	Oke	Ada respon
12	Gagal	Tidak ada respon

5. Kesimpulan

- Data yang dikirim dari *easy bluetooth* tidak mencapai batas maksimum dikarenakan pengambilan data dilakukan

dengan PC menggunakan bluetooth dongle.

- Relay pengapian tidak bekerja, bila relay yang dipasang pada sistem kelistrikan tidak bekerja.
- Buzzer pada rangkaian driver relay digunakan sebagai tanda bahwa sistem kelistrikan telah tersambung.

6. DaftarPustaka

- [1]. <http://lecturer.eepis-its.edu/~yuliana/Bluetooth/yamta-bluetooth.pdf>
- [2]. http://p3m.amikom.ac.id/p3m/dasi/maret05/02%20%20STMIK%20AMIKOM%20Yogyakarta%20Makalah%20ANDI%20teknologi%20arsitektur_.pdf
- [3]. www.cert.or.id/...../ali.doc (Sistem Keamanan Pada Bluetooth)
- [4]. www.parallax.com
- [5]. http://iddhien.com/index.php?option=com_content&task=view&id=45&Itemid=111
- [6]. <http://www.avrku.com/2008/11/berkenalan-dengan-codevision-avr.html>
- [7]. <http://learnautomation.files.wordpress.com/2009/08/modul-keseluruhan-automasi-1-1-bab-2.pdf>
- [8]. RikuMettala; Bluetooth Protocol Stack; 1999
- [9]. M.Ary Heryanto, ST & Ir. Wisnu Adi P.; Pemrograman Bahasa C untuk mikrokontroler Atmega8535; ANDI Yogyakarta; 2008