

BAB III

METODE PENELITIAN DAN PERANCANGAN PEMBUATAN ALAT

Penyusunan skripsi ini didasarkan pada masalah yang bersifat aplikatif, yaitu perencanaan dan perealisasiian alat agar dapat bekerja sesuai dengan yang direncanakan dengan mengacu pada rumusan masalah. Langkah-langkah yang perlu dilakukan untuk merealisasikan alat yang dirancang adalah penentuan spesifikasi alat, identifikasi, perancangan dan pembuatan alat, pengujian alat, dan pengambilan kesimpulan. Perancangan dan pembuatan alat yang secara garis besar terdiri dari dua bagian yaitu :

- Perancangan perangkat keras (*hardware*)
- Perancangan perangkat lunak (*software*)

Pada perancangan perangkat keras meliputi rangkaian–rangkaiian dan sistem antarmuka pada mikrokontroler yang digunakan pada alat ini. Sedangkan pada perancangan perangkat lunak meliputi diagram alir dan *software* secara umum.

3.1 Penentuan Spesifikasi Alat

Spesifikasi alat secara global ditetapkan terlebih dahulu sebagai acuan dalam perancangan selanjutnya. Spesifikasi alat yang direncanakan yaitu :

- Alat akan bekerja ketika *bluetooth* ON lalu *key tag* dimasukkan dan setelah *key tag* ON data diolah menuju mikrokontroler.
- Mikrokontroler yang dipakai AVR ATmega 8, yang berfungsi sebagai pemroses masukan *key tag*.
- *Buzzer* digunakan sebagai indikator jika terjadi kesalahan.
- Modul *bluetooth* yang dipakai adalah HC-05 yang sudah dilengkapi *PCB Extendednya*.
- *Relay* akan menyala ketika semua kondisi syarat terpenuhi.

Identifikasi perancangan alat dilakukan untuk mempelajari teori penunjang sistem yang dibutuhkan dalam perencanaan dan pembuatan alat. Teori yang diperlukan antara lain berkaitan dengan rangkaian modul *Bluetooth* HC-05, *buzzer* dan *relay*.

Perancangan dan pembuatan alat dalam penelitian ini dibagi menjadi dua bagian, yaitu *hardware* dan *software*.

- **Perancangan Alat**

1. **Perancangan *Hardware***

Perancangan perangkat keras dalam sistem ini meliputi perancangan rangkaian *Bluetooth*, mikrokontroler, *buzzer*, *relay* dan perancangan *prototype* alat.

2. **Perancangan Pembuatan Perangkat Lunak (*software*)**

Perancangan perangkat lunak berupa *flowchart* (diagram alir) bahasa pemrograman yang dipakai untuk ATmega8. Bahasa pemrograman yang digunakan adalah bahasa C yang nantinya dibuat dan di *compile* menggunakan *software* codevision AVR.

Pembuatan alat ini meliputi pembuatan perangkat keras (*hardware*) dan pembuatan perangkat lunak (*software*) sebagai berikut :

- **Pembuatan Alat**

1. **Pembuatan perangkat keras (*hardware*)**

Pembuatan perangkat keras ini direalisasikan berdasarkan perancangan yang telah dilakukan dari masing – masing blok diagram.

2. **Pembuatan perangkat lunak (*software*)**

Pembuatan perangkat lunak ini dilakukan berdasarkan diagram alir yang dibuat dan direalisasikan pada *source code* menggunakan *Code Vision* AVR.

3.2 Pengujian Alat

Untuk menganalisis kinerja alat apakah sesuai dengan yang direncanakan maka dilakukan pengujian sistem. Pengujian dilakukan pada masing-masing blok pada perancangan *hardware* serta pengujian keseluruhan untuk mengetahui *software* dapat berjalan atau tidak. Ada tiga pengujian alat yang dilakukan, yaitu :

1. Pengujian Rangkaian *Bluetooth*

Pengujian modul ini dilakukan dengan cara menyambungkan pin-pin *bluetooth* pada regulator dan mengaktifkannya kemudian mentransferkan data dari *master* modul pada *slave* sesuai dengan *input* data pada program.

2. Pengujian Rangkaian *Buzzer*

Pengujian ini bertujuan untuk mengetahui *output* dari *buzzer* berfungsi dengan baik. Pengujian dilakukan dengan melihat *output buzzer* pada rangkaian dengan memberi logika *high* atau *low*.

3. Pengujian Rangkaian *Relay*

Pengujian rangkaian *relay* dengan cara dicatu melalui pin *out* dari mikrokontroler untuk mengetahui apakah *relay* dapat aktif atau tidak.

4. Pengujian Keseluruhan Sistem

Pengujian keseluruhan sistem ini dengan menyambungkan semua *hardware* yang dibuat berdasarkan blok diagram dan memasukkan program berupa *software* yang bekerja untuk mengendalikan *hardware* yang telah dibuat. Sistem bekerja sesuai *flowchart* yang telah direncanakan dan sesuai dengan *input* yang dimasukkan.

Kesimpulan didapat berdasarkan hasil perealisasi sistem Penggunaan *Bluetooth* sebagai Pengaman Ganda pada Sepeda Motor berbasis Mikrokontroler dalam Bentuk *Prototype*. Beberapa hal hasil pengujian disampaikan dalam kesimpulan disertai realita yang disusun secara berurutan.

3.3 Perancangan Pembuatan Alat

3.3.1 Diagram Blok Sistem

Keseluruhan bagian *hardware* atau biasa disebut dengan diagram blok *hardware*. Pada gambar 3.1 merupakan gambar blok diagram dari *hardware*.



Gambar 3.1 Diagram Blok Sistem dari *Hardware*

Penjelasan masing – masing blok diagram :

a. *Smart Phone*

Merupakan modul yang berfungsi sebagai pengirim data menuju ke *bluetooth*.

b. *ATMega8*

Merupakan mikrokontroler yang berfungsi sebagai pemroses dari masukan modul *bluetooth* yang ada di *smart phone*, serta menghasilkan *output* untuk mengaktifkan *buzzer* dan *driver*.

c. *Bluetooth Receiver*

Merupakan modul yang berfungsi sebagai penerima data dari *bluetooth* yang ada di *smart phone*.

d. *Key Tag*

Merupakan bentuk fisik kontak sepeda motor.

e. *Buzzer*

Merupakan komponen elektronika yang berfungsi untuk mengubah getaran listrik menjadi getaran suara.

3.3.2 Prinsip Kerja Alat

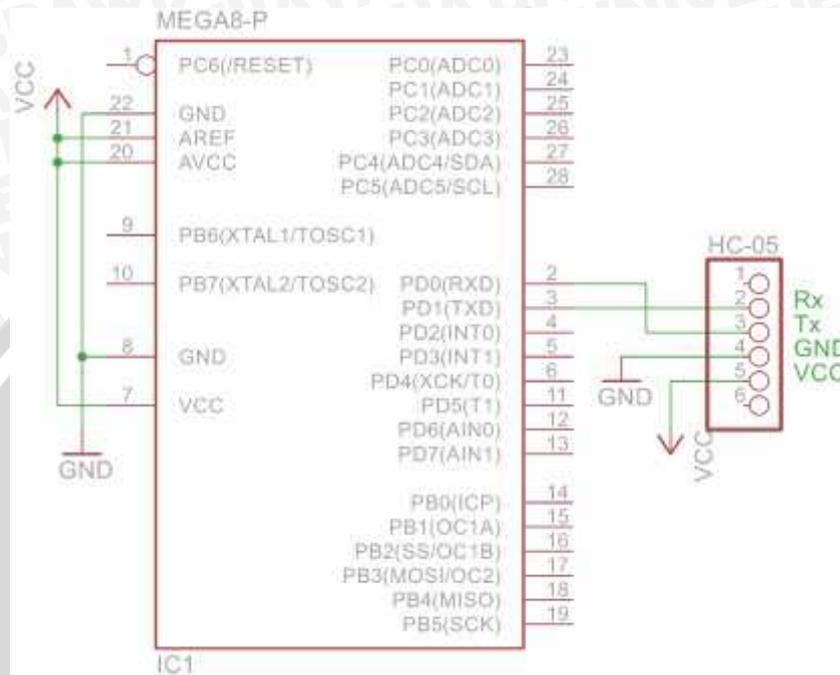
Prinsip kerja alat berdasarkan blok diagram pada gambar 3.1. Pada saat alat kondisi *idle* (diam) maka alat akan menunggu masukkan dari *bluetooth* yang ada di *smart phone* untuk diolah. Setelah *bluetooth* terkoneksi (*pairing*) dan memasukkan *password* maka data akan diolah menuju mikrokontroler. Langkah selanjutnya memasukkan *key tag* (kontak) menjadi ON. Ketika semua telah sesuai dengan prosedur maka *prototype* dapat dijalankan. Dan *buzzer* akan baru menyala jika pelaksanaan perintah tidak sesuai dengan prosedur.

3.3.3 Perancangan Perangkat Keras (*Hardware*)

3.3.3.1 Perancangan Rangkaian Antarmuka *Bluetooth*

Modul *bluetooth* ini digunakan untuk menerima data dari *smart phone*. Agar modul ini dapat bekerja maka *supply* modul diberi tegangan 5V. Rangkaian koneksi antara mikrokontroler dengan *bluetooth* ditunjukkan pada gambar 3.2. Pin-pin yang harus dihubungkan dengan mikrokontroler dalam perancangan ini antara lain :

- Pin 2 dihubungkan dengan Rx
- Pin 3 dihubungkan dengan Tx
- Pin 4 dihubungkan dengan GND
- Pin 5 dihubungkan dengan Vcc 5V

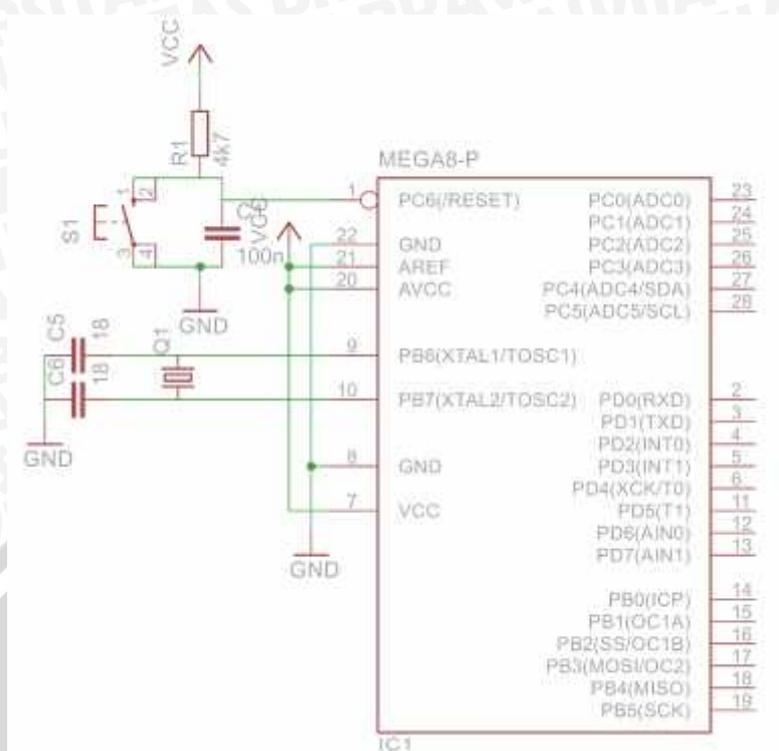


Gambar 3.2 Rangkaian Antarmuka *Bluetooth* dengan Mikrokontroler ATmega8

3.3.3.2 Perancangan Rangkaian Minimum Sistem Mikrokontroler ATmega8

Mikrokontroler yang digunakan pada rangkaian ini adalah ATmega8 yang termasuk dalam seri AVR. IC mikrokontroler dapat bekerja dengan baik jika ditambahkan sebuah rangkaian *oscillator* atau pembangkit *clock* dan rangkaian *reset*. Rangkaian *oscillator* terdiri dari dua kapasitor dan xtal (kristal). Sedangkan untuk rangkaian reset terdiri dari *push button*, resistor, dan kapasitor.

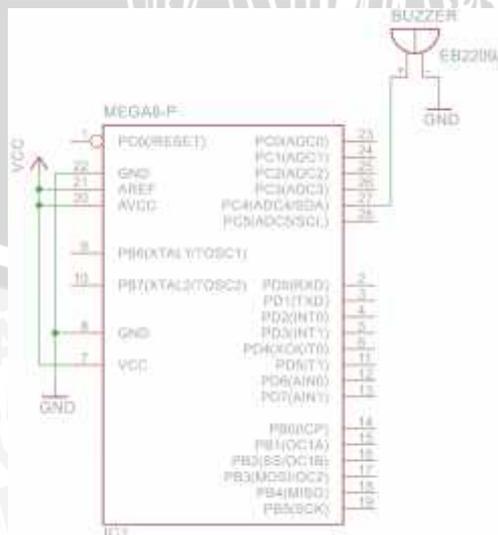
Pada mikrokontroler ATmega8 nilai XTAL yang diizinkan berkisar antara 0.4 MHz sampai 16 MHz. Perancangan ini menggunakan kristal sebesar 11,0592 MHz. Besar nilai C_1 dan C_2 disesuaikan dengan apa yang tertera dalam *datasheet*, yaitu antara 12 pF sampai 22 pF. Perancangan minimum sistem yang digunakan dalam alat ini menggunakan kapasitor 22 pF. Rangkaian minimum sistem ATmega8 ditunjukkan dalam gambar 3.3.



Gambar 3.3 Rangkaian Minimum Sistem Mikrokontroler ATmega8

3.3.3.3 Perancangan Rangkaian Antarmuka Buzzer

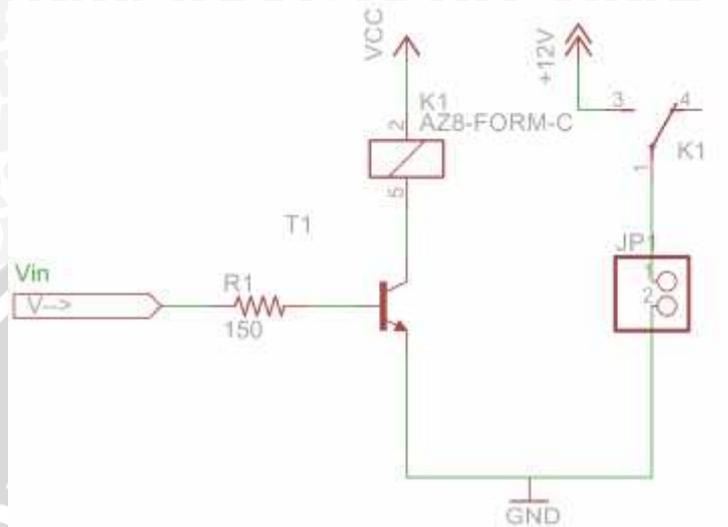
Perancangan antarmuka *buzzer* ini, logika yang digunakan untuk mengaktifkan *buzzer* adalah aktif *high*, *buzzer* akan aktif ketika *output* dari mikrokontroler berlogika 1. Berdasarkan *datasheet*, I_{in-max} yang dapat diterima *buzzer* adalah 40 mA maka *buzzer* tidak membutuhkan resistor untuk mengurangi arus yang memasuki *buzzer* karena *output* mikrokontroler hanya berkisar ± 20 mA. Rangkaian antarmuka *buzzer* dengan mikrokontroler ditunjukkan dalam gambar 3.4.



Gambar 3.4 Rangkaian Antarmuka Buzzer dengan Rangkaian Mikrokontroler

3.3.3.4 Perancangan Rangkaian Relay

Perancangan rangkaian *relay* bertujuan untuk mengaktifkan dan menonaktifkan koil pada sepeda motor dengan pemicuan dari pin keluaran mikrokontroler. Pada perancangan ini menggunakan *relay* dengan tipe SPDT (*Single Pole Dual Throw*). Rangkaian *relay* ditunjukkan pada gambar 3.5.



Gambar 3.5 Rangkaian Relay

Agar rangkaian bekerja sesuai dengan keinginan, *relay* perlu ditrigger terlebih dahulu oleh transistor dikarenakan tegangan pada mikrokontroler tidak bisa langsung mengaktifkan *relay*. Agar tidak terjadi arus balik yang masuk ke mikrokontroler saat pengaktifan, pin keluaran mikrokontroler melewati dioda dalam bentuk LED terlebih dahulu sebelum masuk ke kaki basis transistor. LED juga berfungsi sebagai indikator jika ada trigger dari mikrokontroler. Pemicuan dalam rangkaian ini menggunakan pemicuan logika tinggi pada pin mikrokontroler. Pada perancangan ini menggunakan *relay* 5 V dengan tipe SRD-05VDC.

- *Coil Voltage* V_{dc} = 5 V
- *Nominal Current* = 71,4 mA
- *Coil Resistance* = 70

Berdasarkan parameter di atas dapat dicari arus pada koil dengan $V_{CE(sat)}$ sebesar 0,6 V melalui persamaan :

$$V_S - V_{relay} - V_{CE(sat)} = 0 \quad (4-1)$$

$$V_S - I_{relay} \times R_{relay} - V_{CE(sat)} = 0 \quad (4-2)$$

Berdasarkan persamaan (4-2) didapat perhitungan :

$$\begin{aligned} 5 \text{ V} - 70I_{\text{relay}} - 0,6 \text{ V} &= 0 \\ 4,4 \text{ V} - 70I_{\text{relay}} &= 0 \\ I_{\text{relay}} &= \frac{4,4}{70} \\ &= 62,85 \text{ mA} \end{aligned}$$

Jadi 62,85 mA merupakan arus minimal agar *relay* dapat aktif.

Transistor yang digunakan dalam rangkaian *relay* ini adalah transistor NPN dengan tipe S8050 dikarenakan memiliki spesifikasi berdasarkan *datasheet* sebagai berikut :

- $V_{\text{CE(sat)}} = 0,6 \text{ V}$
- $h_{fe} = 120$
- $V_{\text{BE(sat)}} = 1,2 \text{ V}$

Arus basis pada transistor dapat dihitung melalui persamaan :

$$\begin{aligned} I_{\text{B}} &= \frac{62,85}{120} \\ &= 0,5 \text{ mA} \end{aligned}$$

Basis pada transistor memerlukan arus 10 mA untuk transistor dapat aktif. Setelah mendapatkan besarnya arus yang dibutuhkan kemudian dapat dicari besarnya nilai resistor pada kaki basis. Besarnya nilai resistor yang digunakan pada kaki basis transistor dapat dicari melalui persamaan :

$$V_{\text{S}} - V_{\text{R1}} - V_{\text{BE(sat)}} = 0 \quad (4-3)$$

$$V_{\text{S}} - I_{\text{R1}} \times R_{\text{R1}} - V_{\text{BE(sat)}} = 0 \quad (4-4)$$

Sehingga dari persamaan (4-4) dapat dihitung nilai R_{B} sebagai berikut :

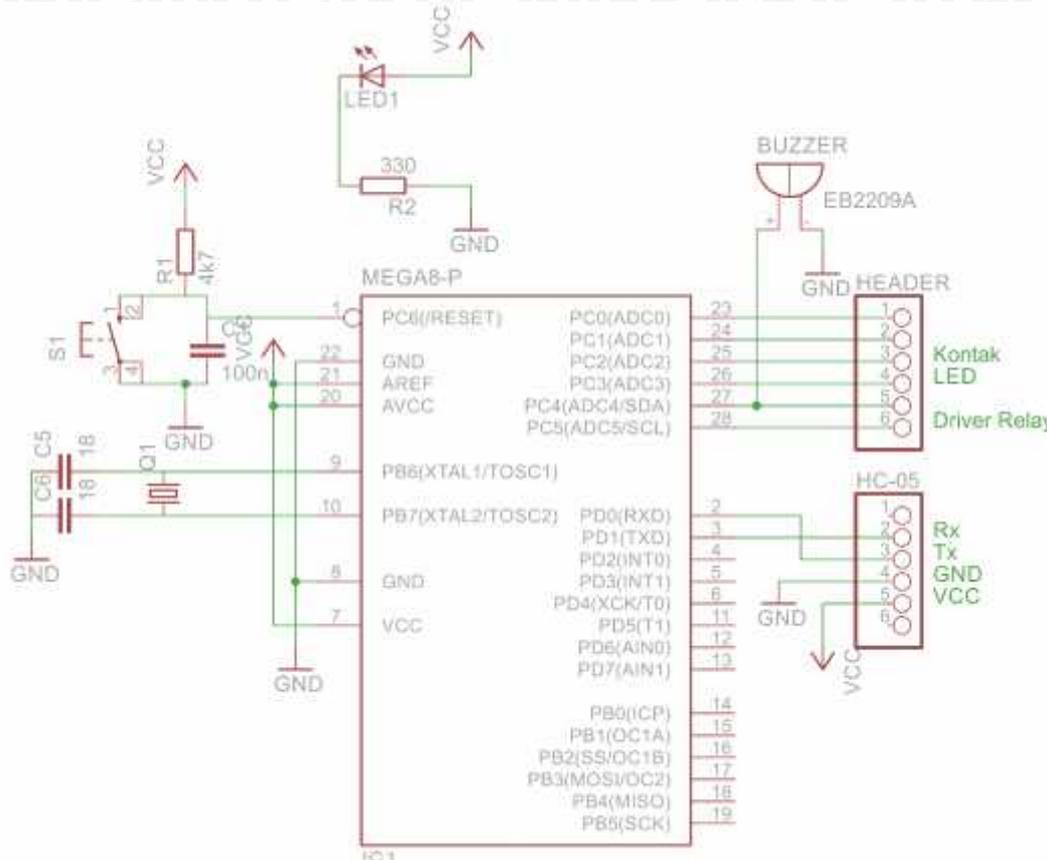
$$\begin{aligned} 4,2 \text{ V} - 0,5R_1 - 1,2 \text{ V} &= 0 \\ 3 \text{ V} - 0,5R_1 &= 0 \\ R_1 &= \frac{3}{0,5} \\ &= 6 \text{ k} \end{aligned}$$

Berdasarkan hasil perhitungan nilai resistansi maksimal agar transistor aktif adalah 6 k . Pada gambar 3.5 digunakan resistor R_1 sebesar 150 sehingga rangkaian masih akan tetap bekerja dengan arus sebesar 20 mA berdasarkan perhitungan dibawah ini :

$$\begin{aligned} 4,2 \text{ V} - 0,15I_{\text{R1}} - 1,2 \text{ V} &= 0 \\ 3 \text{ V} - 0,15I_{\text{R1}} &= 0 \\ I_{\text{R1}} &= \frac{3}{0,15} \\ &= 20 \text{ mA} \end{aligned}$$

3.3.3.5 Perancangan Keseluruhan Sistem

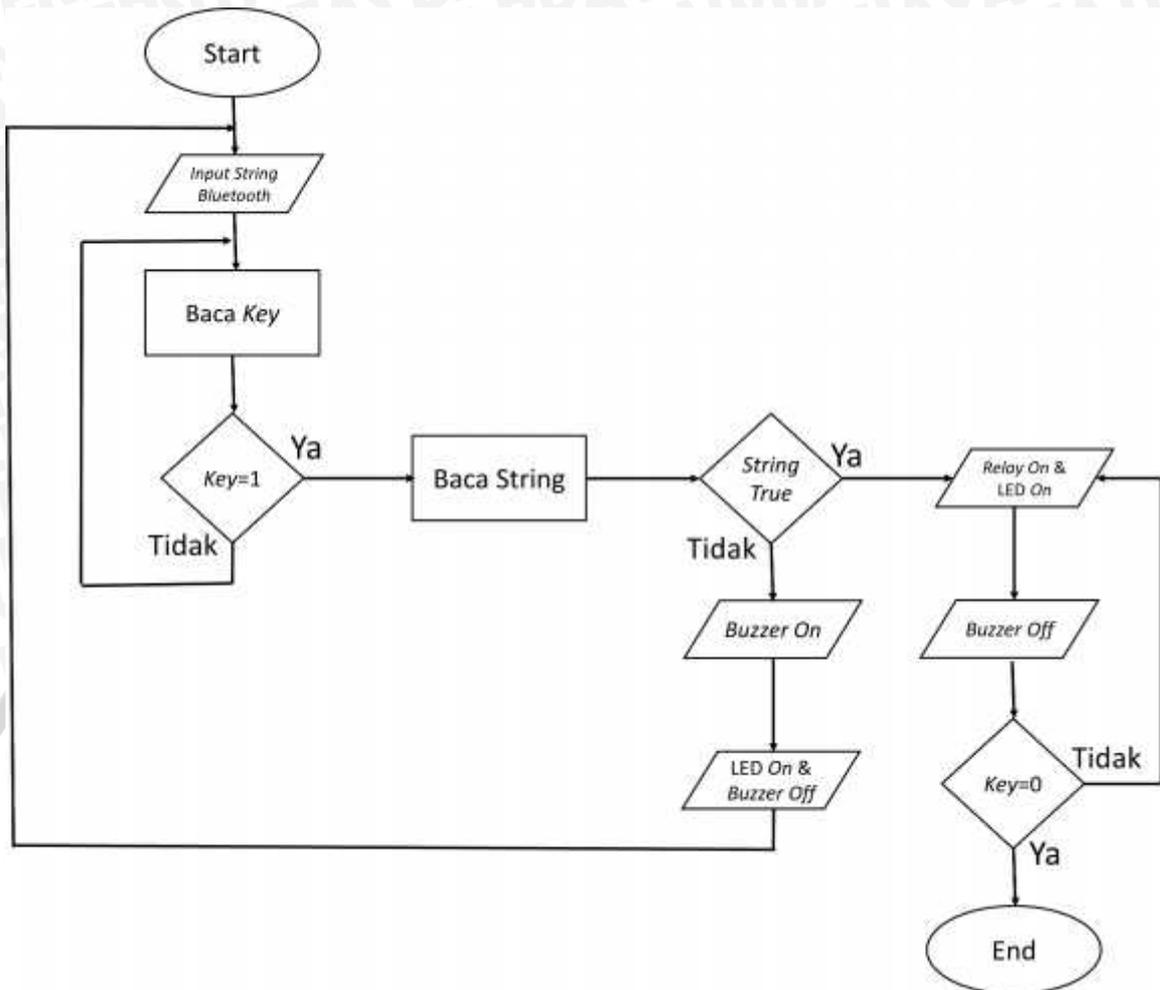
Perancangan ini dilakukan dengan menggabungkan semua perancangan yang telah dilakukan dari masing-masing blok diagram yang telah ditentukan. Sistem keseluruhan dari alat ini ditunjukkan dalam gambar 3.6.



Gambar 3.6 Rangkaian Keseluruhan Sistem

3.3.4 Perancangan Perangkat Lunak (Software)

Perancangan perangkat lunak pada mikrokontroler berupa diagram alir atau *flowchart*. *Flowchart* ini berfungsi sebagai alur kerja dari rangkaian *hardware* yang telah dirancang sebelumnya. Bahasa yang digunakan dalam membuat program untuk alat ini adalah bahasa C, sedangkan *compiler* yang digunakan yaitu *Code Vision AVR*. *Flowchart* dari alat ini ditunjukkan dalam gambar 3.7.



Gambar 3.7 Flowchart Prototype Alat

Pada gambar di atas dijelaskan tentang *flowchart prototype* alat. Pertama melakukan *input* data dari *smart phone* yaitu berupa *char* (data) yang telah di *setting* sebelumnya. Kemudian data akan disimpan dalam mikrokontroler. Setelah itu *key tag* (kontak) dinyalakan guna menstimulasi langkah berikutnya. Terdapat tiga kondisi yang akan terjadi yaitu : kondisi pertama, jika data dari input *smart phone* benar maka driver & LED akan ON. Kondisi kedua jika data yang dimasukkan dari *smart phone* tidak benar maka *buzzer* akan ON dan berbunyi terus-menerus, dimana *buzzer* tersebut akan OFF jika telah memasukkan data yang benar dari *smart phone*. Kondisi ketiga, jika memulai langsung dengan inisialisasi *key tag* (kontak) tanpa menggunakan data dari *smart phone* maka secara otomatis *buzzer* akan ON terus-menerus dan akan OFF jika sudah diberi data masukkan yang sesuai dengan data pada mikrokontroler dari *smart phone*.