

SMART CARD SEBAGAI PENGAMAN SEPEDA MOTOR BERBASIS MIKROKONTROLER

Yohanes Chrisostomus Purba¹⁾, Antonius Wibowo²⁾
E-mail: ycp_hans@yahoo.com, lazeid@yahoo.com

ABSTRAK

Pengaman sepeda motor yang berbentuk smart card terdiri dari bagian yang memuat chip EEPROM dan bagian pengaman smart card. Pembacaan smart card didesain dan ditempatkan pada sepeda motor. Untuk pembacaan data chip SLE 4442, dan pembacaan data pengaman smart card digunakan mikrokontroler AT89S51. Untuk pengaturan aktif maupun tidak aktifnya CDI, dan starter juga digunakan mikrokontroler AT89S51.

Sistem pengaman sepeda motor akan menjadi aktif bila sepeda motor dalam keadaan mati. Smart card dengan chip, dan dengan pengaman smart card (sensor) digunakan pada saat sepeda motor akan dinyalakan, dan apabila smart card benar, maka sepeda motor sudah siap dipakai, dan smart card bisa dilepas dari sistem pengaman sepeda motor. Sistem pengaman sepeda motor harus off bila sepeda motor akan berjalan.

Sumber tenaga listrik untuk menggerakkan alat pengaman sepeda motor ini disuplai oleh accu yang mampu menyediakan tenaga selama 6 sampai dengan 8 jam. Arus listrik yang dikonsumsi oleh alat pengaman sepeda motor sekitar 10mA saat standby. Pada saat alat pengaman sepeda motor ini bekerja, arus yang dikonsumsi sekitar 2A. Alat pengaman sepeda motor ini melakukan fungsi pengamanan dengan baik.

Kata kunci: smart card, pengaman, sepeda motor, mikrokontroler

PENDAHULUAN

Semakin banyak sepeda motor digunakan oleh orang sebagai alat transportasi, semakin banyak pencurian motor terjadi. Oleh karena itu orang memikirkan bagaimana membuat pengaman sepeda motor.

Pada rancangan terdahulu pengaman sepeda motor menggunakan pengaman dengan menggunakan saklar (lebih tepatnya menggunakan saklar *toggle*). Dalam perkembangan teknologi yang semakin pesat, maka dalam penelitian ini telah dirancang alat pengaman sepeda motor dengan menggunakan *card*.

Dalam pengamanan sepeda motor dibutuhkan *smart card*. Muncul pertanyaan, mengapa dipakai *smart card*? Karena *smart card* mempunyai 4 bit sekuriti, di mana 4 bit sekuriti ini berupa *password* yang akan di-*setting* sendiri. *Password* tersebut tidak di-*input*-kan, melainkan di-*setting* dari awal pada mikrokontrolernya. Jadi mikrokontroler langsung meminta kartu, dan membandingkan *password*-nya pada saat kunci kontak sepeda motor *on*. Cara kerja dari alat ini sangat simpel, dan mudah dioperasikan.

TINJAUAN PUSTAKA

Smart Card SLE4442

Memory Card SLE4442 terdiri dari 256 × 8 bit *EEPROM* memori utama dengan 32 bit memori proteksi dengan fungsi *PROM*. Memori utama dihapus, dan ditulis per bit. Ketika dihapus, keseluruhan 8 bit dari sebuah bit data

diberi logika 1. Ketika ditulis, informasi dalam sel *EEPROM* diubah bit per bit menjadi berlogika 0 (logika *AND* antara data lama dan baru dalam *EEPROM*). Biasanya sebuah perubahan data terdiri dari prosedur menghapus, dan menulis. Akan tetapi hal ini tergantung pada isi bit data dalam memori utama, dan bit data baru apakah *EEPROM* benar-benar dihapus, dan atau ditulis. Jika tidak ada salah satu dari 8 bit yang membutuhkan transisi 0 ke 1, penghapusan data tidak akan dilakukan. Sebaliknya, penulisan tidak akan dilakukan bila tidak ada transisi 1 ke 0 yang diperlukan. Operasi tulis maupun baca memerlukan waktu sekitar 2,5 ms^[1].

Selain itu, 32 bit pertama diproteksi dari perubahan data dengan menulis bit yang berhubungan dengannya pada memori proteksi. Setiap bit data dalam *range* alamat ini dihubungkan dengan 1 bit memori proteksi yang mempunyai alamat yang sama dengan memori utama yang dihubungkan. Sekali ditulis, bit proteksi tidak dapat dihapus.

Memory card SLE4442 menyediakan sebuah logika kode sekuriti yang mengatur akses penulisan atau penghapusan memori. Untuk keperluan ini, *SLE4442* mempunyai 4 bit sekuriti memori dengan sebuah *Error Counter/EC* (bit 0 sampai bit 2), dan 3 bit data referensi. Keseluruhan dari 3 bit disebut *Programmable Security Code (PSC)*. Setelah diberi *power*, data pada memori dapat dibaca kecuali data referensi. Hanya setelah perbandingan data verifikasi yang benar dengan data referensi,

¹⁾ Mahasiswa di Fakultas Teknik Jurusan Teknik Elektro Universitas Katolik Widya Mandala Surabaya

²⁾ Staf Pengajar di Fakultas Teknik Jurusan Teknik Elektro Universitas Katolik Widya Mandala Surabaya

memori dapat diakses penuh sampai *power* dimatikan. Setelah 3 perbandingan gagal, *Error Counter* akan menghentikan akses tulis, dan hapus lebih lanjut.

Konfigurasi Pin Memory Card SLE4442

Memory card SLE4442 memiliki konfigurasi pin sebagaimana disajikan pada Gambar 1. Lima dari delapan pin yang ada harus terhubung sesuai dengan fungsinya masing-masing agar *memory card SLE4442* dapat diakses.

VCC	C1	C5	GND
RST	C2	C6	N.C.
CLK	C3	C7	I/O
N.C.	C4	C8	N.C.

Gambar 1. Konfigurasi Pin *Memory Card SLE4442*

Pada Tabel 1 akan dijelaskan fungsi-fungsi setiap pin dalam *memory card SLE4442*. Untuk pin *VCC*, *RST*, *CLK*, *I/O*, dan *GND* akan dihubungkan dengan mikrokontroler *AT89S51*.

Tabel 1. Fungsi Pin pada *Memory Card SLE4442*

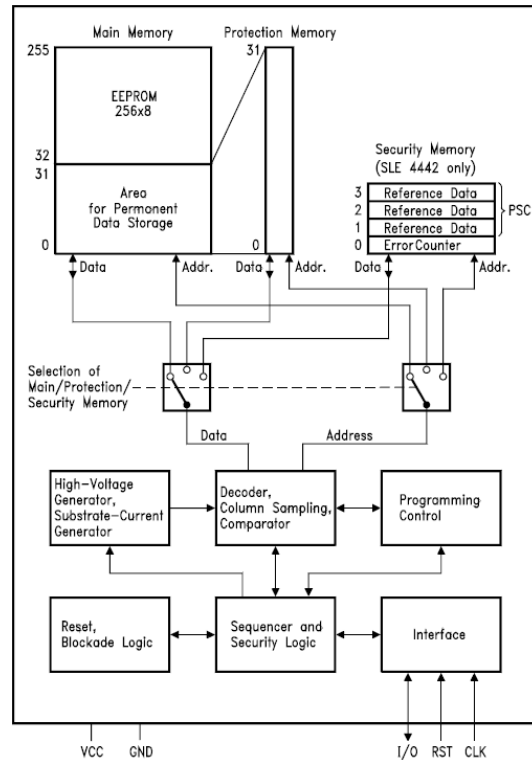
Kontak	Simbol	Fungsi
C1	VCC	Sumber tegangan
C2	RST	Jalur reset
C3	CLK	Jalur clock
C4	N.C.	Tidak terhubung
C5	GND	Ground
C6	N.C.	Tidak terhubung
C7	I/O	Jalur data
C8	N.C.	Tidak terhubung

Dalam pembuatan perangkat keras (*hardware SLE4442*), digunakan sebuah *socket amphenol* untuk menghubungkan pin-pin yang ada pada kartu dengan *header* yang terletak pada *socket amphenol* agar memudahkan dalam menghubungkannya dengan mikrokontroler *AT89S51*^[2,3].

Diagram Blok Memory Card SLE4442

Di dalam kartu *SLE4442* terdapat teknologi yang kompleks. Pada Gambar 2 disajikan diagram blok bagian-bagian fungsi yang terdapat pada kartu *SLE4442*.

Ada tiga macam *memory* di dalam kartu *SLE4442*. Yang pertama adalah *main memory*. *Main memory* merupakan tempat penyimpanan data dengan kapasitas 256 bit. Yang kedua adalah *protection memory* yang merupakan 32



Gambar 2. Gambaran Fungsi Kartu *SLE4442* Secara Umum

bit data yang menentukan apakah lokasi pada *main memory* yang berhubungan dengannya dapat diubah atau tidak. Yang ketiga adalah *security memory* yang merupakan lokasi yang berisi 3 bit data referensi dengan sebuah *error counter*. *Selection of Main/Protection/Security Memory* digunakan oleh kartu *SLE4442* untuk memilih pengaksesan *memory* dalam *SLE4442*.

Kartu *SLE4442* ini berhubungan dengan *device* lain lewat 3 buah pin yaitu: pin *I/O*, pin *RST*, dan pin *CLK*. Ketiga pin ini digunakan untuk mengakses kartu *SLE4442*.

Protokol Transmisi

Pengaksesan kartu *SLE4442* dilakukan dengan protokol hubungan 2 kabel antara *IFD*, dan *IC*. Segala perubahan data diawali dengan sisi turun pada *clock*. Protokol transmisi yang ada pada kartu *SLE4442* dibagi menjadi 4 *mode* yaitu:

- *Reset*, dan *Answer-to-Reset*;
- *Command Mode*;
- *Outgoing Data Mode*;
- *Processing Mode*.

Answer-to-Reset dilakukan berdasarkan pada *ISO standard 7816-3 (ATR)*. *Reset* dapat diberikan pada waktu kapanpun selama operasi. Proses ini dimulai dengan transisi *low* ke *high* pada *RST*, yang diikuti dengan sebuah pulsa pada jalur *clock*. Proses ini diakhiri dengan

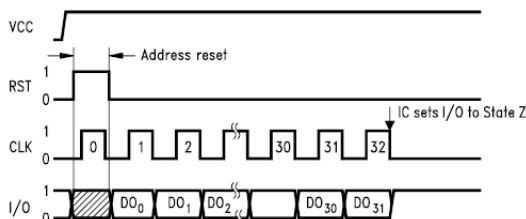
transisi *high* ke *low* pada *RST*. Setelah proses ini dilakukan, 32 buah pulsa *clock* dibutuhkan untuk mengeluarkan 32 bit isi alamat *EEPROM* yang pertama. Empat (4) bit data ini merupakan pesan balasan dari *SLE4442*. *Clock* ke-33 akan membuat *I/O* menjadi berimpedansi tinggi, dan sekaligus mengakhiri prosedur *ATR*.

Tabel 2 merupakan urutan penerimaan dalam *ATR*. Tabel tersebut menunjukkan bahwa *SLE4442* akan mengirimkan *least significant* bit terlebih dahulu.

Tabel 2. Urutan Data *ATR*

Bit 0	DO7...DO0
Bit 1	DO15...DO8
Bit 2	DO23...DO16
Bit 3	DO31...DO24

Gambar 3 menunjukkan *timeline* dari proses *reset*, dan *answer to reset*.



Gambar 3. *Timeline Answer-to-reset*

Selain *Reset*, dan *Answer-to-reset*, terdapat tiga buah mode operasi lainnya pada *SLE4442*, yaitu *command mode*, *outgoing data mode*, dan *processing mode*.

Pada *command mode*, *chip* akan menerima perintah. Perintah ini diawali dengan kondisi *start*, diikuti dengan 3 bit perintah, dan diakhiri dengan kondisi *stop*. Kondisi *start* adalah kondisi saat sisi turun pada *I/O* selama *CLK* pada level *H*. Sedangkan kondisi *stop* adalah kondisi saat sisi naik pada *I/O* selama *CLK* pada level *H*.

Setelah menerima perintah dengan benar, ada 2 kemungkinan mode yang akan terjadi, yaitu *outgoing data mode*, dan *processing mode* untuk penulisan, dan penghapusan.

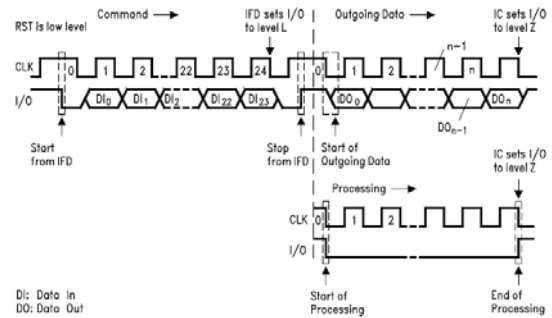
Pada *outgoing data mode*, *IC* mengirimkan data ke *IFD*. Bit pertama akan valid pada *I/O* setelah sisi turun pertama pada *CLK*. Setelah bit data terakhir, sebuah pulsa *clock* tambahan diperlukan untuk membuat *I/O* menjadi berimpedansi tinggi, dan menyiapkan *IC* untuk perintah selanjutnya. Selama mode ini, kondisi *start*, dan *stop* dihilangkan.

Pada *processing mode*, *IC* memproses secara internal. *IC* harus diberi sinyal *clock* secara terus-menerus sampai *I/O* yang kondisinya berada pada level *L* setelah sisi

turun pertama pada *CLK*, di-*set* kembali ke impedansi tinggi. Kondisi *start*, dan *stop* dihilangkan dalam mode ini.

Pada *command mode*, *outgoing data mode*, dan *processing mode*, jalur *RST* berlogika *low*. Jika *RST* di-*set* *high*, operasi dibatalkan, dan *I/O* di-*set* ke impedansi tinggi (*break*).

Pada Gambar 4 disajikan *timeline* dari *command mode*, *outgoing data mode*, dan *processing mode*.



Gambar 4. Mode Operasi

Perintah Pengoperasian

Untuk mengakses kartu *SLE4442* terdapat berbagai macam perintah yang telah disediakan. Format pengiriman perintah disajikan pada pada Tabel 3.

Tabel 3. Urutan Pengiriman Perintah

MSB	Control				LSB	MSB	Address				LSB	MSB	Data				LSB						
B7	B6	B5	B4	B3	B2	B1	B0	A7	A6	A5	A4	A3	A2	A1	A0	D7	D6	D5	D4	D3	D2	D1	D0

Setiap kali perintah dikirimkan, terdapat 24 bit data yang harus dikirimkan, dan dibagi dalam 3 bagian. Bagian pertama (B7 sampai dengan B0) menunjukkan perintah yang akan dilakukan. Bagian kedua (A7 sampai dengan A0) adalah alamat yang akan digunakan (bila ada). Dan bagian ketiga (D7 sampai dengan D0) adalah data yang akan digunakan (bila ada). Perintah dalam *memory card SLE4442* disajikan pada Tabel 4.

Tabel 4. Perintah Dalam *Memory Card SLE4442*

Nama perintah	Kode	Alamat	Data
<i>Read Main Memory</i>	30H	00H...FFH	Tidak ada efek
<i>Read Protection Memory</i>	34H	Tidak ada efek	Tidak ada efek
<i>Update Main Memory</i>	38H	00H...FFH	00H.FFH
<i>Write Protection Memory</i>	3CH	00H...1FH	00H
<i>Read Security Memory</i>	31H	Tidak ada efek	Tidak ada efek
<i>Update Security Memory</i>	39H	00H...03H	00H...FFH
<i>Compare Verification Data</i>	33H	01H...03H	00H...FFH

Pada Tabel 4 ditunjukkan seluruh perintah yang dapat digunakan dalam *memory card SLE4442*. *Memory card SLE4442* mempunyai 7 buah perintah yang dapat dibagi menjadi beberapa bagian. Perintah *read main memory*, dan *read protection memory* adalah dua buah perintah yang dapat dilakukan kapanpun diperlukan. Perintah *update main memory*, *write protection memory*, dan *update security memory* pada alamat 1 sampai dengan 3 akan dapat dilakukan setelah perintah *compare verification data* selesai dilakukan dengan benar. Perintah *read security memory* akan mengeluarkan 4 buah bit data jika perintah *compare verification data* sebelumnya telah dilakukan dengan benar. Jika tidak, perintah *read security memory* hanya akan mengeluarkan data berupa *error counter* yang menunjukkan berapa kali kesempatan memasukkan pin dapat dilakukan.

Perintah *read main memory* adalah perintah untuk membaca isi dari memori utama pada alamat tertentu ($N = 0, \dots, 255$) sampai akhir dari *memory*. Setelah perintah dimasukkan, *IFD* harus memberikan pulsa *clock* yang cukup. Jumlah *clock* yang harus diberikan adalah $m = (256 - N) \times 8 + 1$. Akses pembacaan ke *main memory* selalu dimungkinkan.

Perintah *read protection memory* adalah perintah untuk membaca daerah-daerah terproteksi dalam memori. Akses pembacaan ke *protection memory* selalu dimungkinkan.

Perintah *update main memory* adalah perintah untuk menulis sebuah bit data ke dalam *main memory*. Jumlah *clock* yang diperlukan tergantung pada data lama, dan data baru. Untuk menghapus dan menulis dibutuhkan waktu sekitar 5 ms atau dengan 255 buah pulsa. Untuk menghapus atau menulis saja memerlukan waktu 2,5 ms, atau dengan pulsa sebanyak 124 buah pulsa *clock* dengan asumsi semuanya dikerjakan dengan kecepatan *clock* 50 kHz.

Jika alamat yang akan diubah ternyata diproteksi, maka *I/O* akan menjadi berimpedansi tinggi setelah *clock* kedua dalam *mode processing*.

Perintah *write protection memory* adalah perintah untuk memproteksi 32 bit pertama alamat *main memory* dari perubahan. Untuk *clock* pulsa yang dibutuhkan sama dengan perintah *update main memory*. Setiap bit pada *protection memory* mewakili 1 bit alamat dalam *main memory*. Sedangkan alamat *main memory*

berkisar antara 32 sampai dengan 255 tidak mempunyai bit proteksi.

Perintah *read security memory* adalah perintah untuk membaca 4 bit *security memory*. Jumlah *clock* selama *outgoing data mode* adalah sebanyak 32 buah *clock*. Sebelum dilakukan *Programmable Security Code (PSC)* dengan benar, 3 bit data referensi akan bernilai *low*.

Perintah *update security memory* adalah perintah untuk menulis *security memory*. Kalau *PSC* belum dilakukan dengan benar, maka hanya bit-bit dalam alamat 0 saja yang dapat diubah dari “1” ke “0”. Banyaknya *clock* yang diperlukan sama dengan pulsa *clock* yang dijelaskan dalam *update main memory*.

Perintah *compare verification data* adalah perintah yang hanya dapat digunakan dalam kombinasinya dengan prosedur *update error counter* dalam *PSC*. Perintah ini membandingkan 1 bit data verifikasi yang dimasukkan dengan data bit pada referensi yang bersangkutan. Untuk prosedur ini, pulsa *clock* diperlukan selama *processing mode*.

Prosedur PSC

Prosedur *PSC* disajikan pada Tabel 5 sebagai berikut.

Tabel 5. Prosedur *PSC*

Nama perintah	Kode	Alamat	Data	Keterangan
	B7...B0	A7...A0	D7...D0	
<i>Read Security Memory</i>	31H	Tidak ada efek	Tidak ada efek	Melihat <i>error counter</i>
<i>Update Security Memory</i>	39H	0H	Data	Menulis sebuah bit yang bernilai 1 menjadi 0
<i>Compare Verification data</i>	33H	01H	Data	Data referensi bit pertama
<i>Compare Verification data</i>	33H	02H	Data	Data referensi bit kedua
<i>Compare Verification data</i>	33H	03H	Data	Data referensi bit ketiga
<i>Update Security Memory</i>	39H	00H	FFH	Menghapus <i>error counter</i>
<i>Read Security Memory</i>	31H	Tidak ada efek	Tidak ada efek	Melihat <i>error counter</i>

Untuk dapat mengakses *memory card SLE4442*, diperlukan prosedur verifikasi *Programmable Security Code (PSC)* yang benar. Variasi apapun pada prosedur ini akan membuatnya gagal, sehingga akses penulisan atau penghapusan tidak akan tercapai. Selama

prosedur tersebut tidak dilakukan dengan benar, bit-bit *error counter* hanya dapat diubah dari "1" ke "0", tetapi tidak bisa dihapus.

Pertama-tama, sebuah bit *error counter* harus ditulis menjadi 0 dengan perintah *update security memory*, dan diikuti dengan tiga perintah *compare verification data* yang dimulai dengan bit pertama dari data referensi. Sebagai tanda bahwa seluruh prosedur sudah dilakukan dengan benar adalah dapat dihapusnya *error counter* yang tidak terhapus secara otomatis. Setelah itu, akses penulisan, dan penghapusan pada seluruh area memori dapat dilakukan selama tegangan listrik masih diberikan. Bila terdapat kesalahan, seluruh prosedur di atas dapat diulang asalkan bit *error counter* yang akan dihapus masih ada. Setelah berhasil, data referensi juga diijinkan untuk diubah.

Pada Tabel 5 di atas ditunjukkan langkah-langkah yang harus dilakukan dalam menyelesaikan prosedur *PSC*. Bagian abu-abu menunjukkan bagian yang penting dalam prosedur tersebut.

Prosedur ATR

ATR (answer-to-reset) merupakan sebuah bagian pemrograman *memory card SLE4442* yang mempunyai banyak informasi di dalamnya. Selain untuk memulai pengaksesan *memory card SLE4442*, *ATR* juga dapat memberikan identitas *memory card* tersebut. Pada Tabel 6 disajikan isi dari *ATR memory card SLE4442*.

Tabel 6. Urutan Data *ATR*

Byte 1							
b8	b7	b6	b5	b4	b3	b2	b1
1	0	1	0	0	0	1	0
Byte 2							
b8	b7	b6	b5	b4	b3	b2	b1
0	0	0	1	0	0	1	1
Byte 3							
b8	b7	b6	b5	b4	b3	b2	b1
0	0	0	1	0	0	0	0
Byte 4							
b8	b7	b6	b5	b4	b3	b2	b1
1	0	0	1	0	0	0	1

Isi dari *ATR* berjumlah 32 bit. Setelah seluruh 32 bit diterima, dan dimasukkan dalam 4 buah variabel, maksud dari isi *ATR* itu dapat dilihat. Pada bit pertama, 4 bit pertama (b8,..., b5) berisi tipe protokolnya. Dalam alat pengaman sepeda motor ini menggunakan kartu

SLE4442, sehingga isinya adalah "1010" yang artinya *2 wire bus protocol*. Satu (1) bit selanjutnya (b4) adalah *RFU* yang berarti disiapkan untuk penggunaan lainnya yang belum ditentukan. Tiga (3) bit terakhir (b3, b2, dan b1) berisi identitas struktur. Dalam hal ini berisi "010" yang artinya *general purpose*, atau kartu *SLE4442* ini untuk penggunaan umum.

Pada bit kedua, 1 bit pertama (b8) menandakan apakah proses baca akan dilakukan sampai akhir, atau dengan batasan panjang tertentu. Dalam hal ini nilainya "0" yang berarti proses pembacaan akan dilakukan sampai akhir. Empat (4) bit selanjutnya (b7, ..., b4) menandakan jumlah unit data. Dalam hal ini berisi "0010" yang berarti 256 buah unit data yang menunjukkan besarnya *main memory*. Tiga (3) bit terakhir (b3, b2, dan b1) menunjukkan panjang unit data. Dalam hal ini nilainya "011" yang berarti panjang bit per data adalah dua pangkat tiga yaitu delapan bit.

Pada bit ketiga, 8 bit data (b8, ..., b1) menunjukkan kategorinya menurut *ISO 7816-4*. Dalam hal ini berisi "00010000". Pada bit keempat, 1 bit pertama (b8) digunakan untuk menentukan 7 bit selanjutnya (b7, ..., b1) apakah isinya referensi data *DIR* atau *RFU*.

Mikrokontroler AT89S51

Mikrokontroler *AT89S51* mempunyai fitur utama 128 bit *RAM*, 4 kbit *ROM*, 2 *timer*, 1 *port* serial, 4 *port I/O* (masing-masing *port*

Tabel 7. Tegangan, Arus Minimum, dan Maksimum pada Pin Mikrokontroler

Symbol	Parameter	Condition	Min	Max	Units
V _{IL}	Input Low Voltage	Except EA	-0.5	0.1	V
V _{IL1}	Input Low Voltage (EA)		-0.5	0.3	V
V _{IH}	Input High Voltage	Except XTAL1, RST	0.9	0.5	V
V _{IH1}	Input High Voltage	XTAL1, RST	0.7	0.5	V
V _{OL}	Output Low Voltage	Ports 1,2,3, I _{OL} = 1.6mA		0.45	V
V _{OL1}	Output Low Voltage	Port 0, ALE, PSEN, I _{OL} = 3.2mA		0.45	V
V _{OH}	Output High Voltage Ports 1,2,3	I _{OH} = -60uA	2.4		V
		I _{OH} = -25uA	0.75		V
	ALE, PSEN	I _{OH} = -10uA	0.9		V
V _{OH1}	Output High Voltage Port 0	I _{OH} = -800uA	2.4		V
		I _{OH} = -300uA	0.75		V
	External Bus Mode	I _{OH} = -80uA	0.9		V
I _{IL}	Logical 0, Input Current Ports 1,2,3	V _{IN} = 0.45 V		-50	uA
I _{TL}	Logical 1 to 0 Transition Current Ports 1,2,3	V _{IN} = 2 V		-650	uA
I _{LI}	Input Leakage Current Port 0, EA	0.45 < V _{IN} < V _{CC}		±10	uA
I _{CC}	Power Supply Current	Active Mode		20	mA
		Idle Mode		5	mA
	Power Down Mode	V _{CC} = 6V		100	uA
		V _{CC} = 3V		40	uA

berisi 8 bit), dan 6 *interrupt source* dalam sebuah *chip* tunggal. Tegangan, arus minimum, dan maksimum pada pin mikrokontroler disajikan pada Tabel 7.

Tegangan pada suhu -40 sampai 80°C sebesar 4 sampai 5,5 V. Dari Tabel 7 dapat diketahui I_{OL} maksimum per *port* pin adalah 10 mA. I_{OL} maksimum per 8-bit *port* pada *port* 0 adalah 26 mA, dan pada *port* 1,2,3 adalah 15 mA. Total I_{OL} maksimum untuk semua *output* pin adalah 71 mA.

Clock

Mikrokontroler AT89S51 mempunyai osilator internal (*on chip oscillator*) yang dapat digunakan sebagai *clock* (pewaktu). Osilator internal akan menentukan *state* yang membentuk sebuah *machine cycle*. *Machine cycle* terbagi atas enam *state* yang masing-masing kondisi panjangnya dua periode osilator. Dengan demikian satu *machine cycle* dikerjakan selama dua belas periode osilator. Berikut bentuk matematis mengenai perhitungan waktu untuk satu *machine cycle*:

$$1 \text{ machine cycle} = 12 \times \text{periode oscillator} \quad (1)$$

Oleh karena $T = \frac{1}{f}$, maka:

$$T_{\text{machine cycle}} = 12 \times \frac{1}{f_{\text{kristal}}} \quad (2)$$

Timer

Mikrokontroler AT89S51 mempunyai 2 buah *timer* utama, yaitu:

- o Register TMOD (*Timer Mode*).

Timer 0, dan *timer* 1 menggunakan register TMOD untuk menentukan mode operasi. TMOD mempunyai register 8 bit di mana 4 bit *lower* untuk *Timer* 0 dan 4 bit *upper* untuk *Timer* 1 sebagaimana disajikan pada Gambar 5.



Gambar 5. Register TMOD

Keterangan:

GATE : menghubungkan *timer/counter* dengan pin $INTx$

C/T : menentukan fungsi sebagai *timer* atau *counter*

Keterangan tentang register TMOD disajikan pada Tabel 8.

Tabel 8. Keterangan Register TMOD

M1	M0	Mode	Operating Mode
0	0	0	13-bit <i>Timer</i> (8048 compatible)
0	1	1	16-bit <i>Timer / Counter</i>
1	0	2	8-bit <i>Auto-Reload Timer / Counter</i>
1	1	3	(<i>Timer</i> 0) <i>TL0</i> is an 8-bit <i>Timer / Counter</i> controlled by the standard <i>Timer</i> 0 control bits. <i>TH0</i> is an 8-bit <i>Timer</i> and is controlled by <i>Timer</i> 1 control bits.
1	1	3	(<i>Timer</i> 1) <i>Timer / Counter</i> 1 stopped

Register Timer

AT89S51 mempunyai dua buah *timer*, yaitu *Timer* 0, dan *Timer* 1. Masing-masing *timer* mempunyai register 16 bit. Register 16 bit tersebut diakses sebagai *low byte*, dan *high byte* yaitu *TL0* untuk *Timer* 0 *low byte*, *TH0* untuk *Timer* 0 *high byte*, *TL1* untuk *Timer* 1 *low byte*, dan *TH1* untuk *Timer* 1 *high byte*.

Reset

Reset pada mikrokontroler adalah suatu kondisi yang akan mengembalikan register ke keadaan awal sebagaimana disajikan pada Tabel 9, dan akan menjalankan program dari alamat 0000H. Kondisi reset akan terjadi apabila pin *RST* mendapat logika ‘1’ selama minimal dua *machine cycle*. Setelah kondisi pin *RST* kembali berlogika ‘0’, mikrokontroler akan mulai menjalankan program dari alamat 0000H.

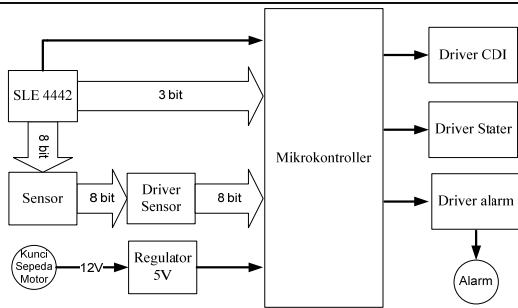
Tabel 9. Isi Register Setelah Kondisi Reset

Register	Isi Register (heksa)
PC	0000
DPTR	0000
ACC	00
PSW	00
SP	07
B	00
P0-P3	FF

METODE PENELITIAN

Perancangan Sistem

Dalam bagian ini akan dibahas perencanaan perangkat keras, dan perangkat lunak. Perangkat keras yang akan didesain meliputi *power supply* 12 V sampai 5 V, sensor *driver*, *relay driver*, mikrokontroler, pembaca kartu SLE4442, dan *smart card* SLE4442. Sedangkan perangkat lunak yang akan didesain adalah program bahasa C dengan *software Raisonance Kit* 6.1 sebagai pembaca *password* SLE4442 dan pengontrol *CDI*, alarm, dan starter. Diagram alir alat pengaman sepeda motor disajikan pada Gambar 6.



Gambar 6. Diagram Blok Alat Pengaman Sepeda Motor

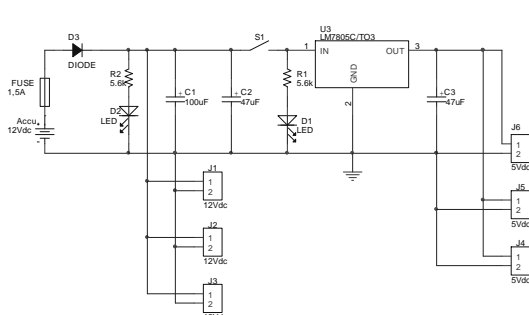
Penjelasan dari diagram alir alat pengaman sepeda motor pada Gambar 6 adalah sebagai berikut: sistem pengaman sepeda motor ini dalam keadaan aktif. Pada saat sepeda motor dinyalakan, hal yang harus dilakukan pertama kali adalah memutar kunci kontak sepeda motor sampai pada ON. Setelah ON, maka hal kedua adalah pengemudi memasukkan smart card tersebut ke dalam kotak pengaman pada sepeda motor. Jika smart card benar, maka secara otomatis semua pengaman (pengaman starter, dan CDI) akan tersambung, dan smart card pun bisa diambil.

Perancangan Elektronik

Perancangan elektronik terdiri dari perancangan rangkaian power supply 12 V sampai 5 V, sensor driver, relay driver, pembaca kartu SLE4442, dan smart card SLE4442.

Perancangan Power Supply dari Accu

Power supply dari accu, sebagaimana rangkaiannya disajikan pada Gambar 7, digunakan untuk menyuplai tegangan pada rangkaian pengaman sepeda motor^[4]. Accu yang digunakan adalah accu 12 V, dan 5 Ah.

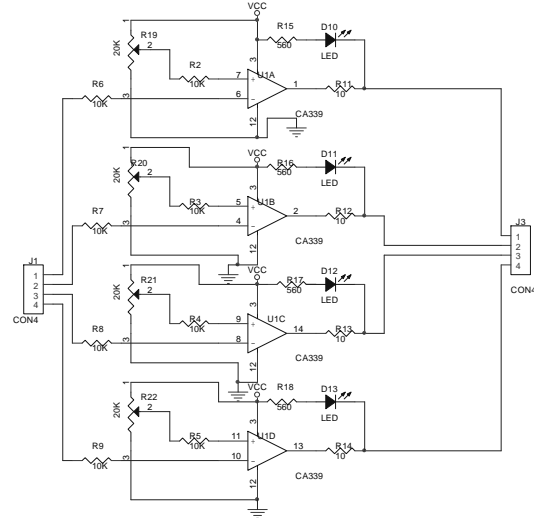


Gambar 7. Rangkaian Power Supply dari Accu

Pemakaian arus listrik terbesar dalam alat ini adalah alarm di mana arus maksimum alarm pada saat berbunyi adalah sebesar 500mA – 1A. Cara mengatasi bila terjadi hubungan arus pendek, pada rangkaian ini diberi fuse sebesar 1,5 A.

Perancangan Sensor Driver^[5]

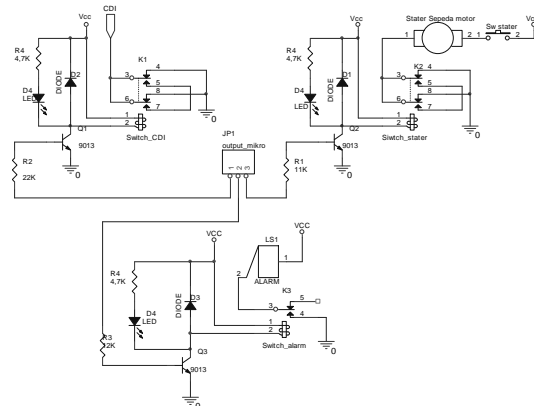
Sensor driver, sebagaimana disajikan pada Gambar 8 menggunakan komparator IC LM 339. Sensor driver digunakan untuk membandingkan tegangan dari sensor cahaya terhadap tegangan referensi, dan akhirnya memberikan hasil output high 3,5 V DC dan low 0,3 V DC,



Gambar 8. Rangkaian Sensor Driver

Perancangan Relay Driver

Rangkaian lengkap relay driver sebagaimana disajikan pada Gambar 9 digunakan untuk mengendalikan on/off dari relay lewat pengontrolan output mikrokontroler. Dalam hardware alat pengaman sepeda motor ini, dibutuhkan 3 relay sebagai saklar otomatis dari CDI, starter, dan alarm. Input dari relay driver ini dikendalikan oleh 3 bit output mikrokontroler yaitu P0.0, P0.1, dan P0.2.



Gambar 9. Rangkaian Relay Driver

Untuk mengetahui nilai arus listrik yang dibutuhkan oleh relay, maka dilakukan pengukuran besarnya tahanan tiap-tiap relay,

dan memulai perhitungan untuk mencari arus *relay* tersebut.

• **Relay Driver Kuning**

Rangkaian *Relay driver* kuning disajikan pada Gambar 10. Adapun perhitungan untuk mencari arus *relay* tersebut adalah sebagai berikut.

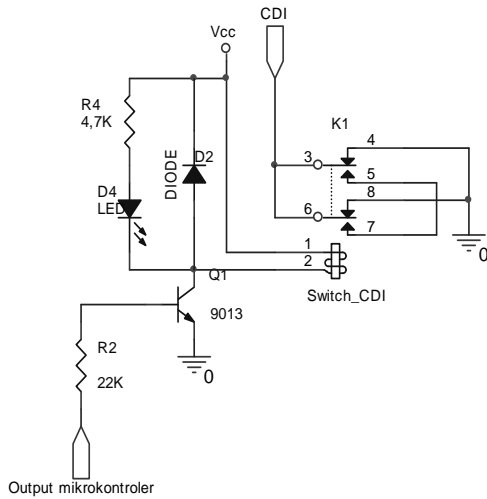
$$R_{\text{relay kuning}} = 698 \Omega$$

$$I_{rk} = \frac{V_{cc}}{R_{rk}} \quad (3)$$

$$I_{rk} = \frac{12}{698} = 0,017A$$

dengan:

I_{rk} = arus listrik dalam *relay* kuning;
 V_{cc} = tegangan listrik suplai 12V DC;
 R_{rk} = tahanan listrik *relay* kuning.



Gambar 10. Rangkaian *Relay Driver* Kuning

Untuk mengetahui tahanan listrik basis (R_b), maka perhitungannya adalah sebagai berikut:

$$I_{rk} = I_{Cmaks} = 0,017A$$

$$\beta_{DC} = 100$$

$$I_{Bmaks} = \frac{I_{Cmaks}}{100} = \frac{0,017}{100} = 0,00017A$$

$$R_{Bmaks} = \frac{V_{BB} - 0,7}{I_{Bmaks}} = \frac{4,5 - 0,7}{0,00017} = 22\,352 \Omega$$

dengan:

$I_{c maks}$ = arus listrik kolektor maksimal;
 β_{DC} = beta DC;
 $I_{B maks}$ = arus listrik basis maksimal;
 V_{BB} = tegangan listrik *input* basis;
 $R_{B maks}$ = tahanan listrik pada basis.

Resistor dengan nilai 22 352 Ω tidak ada di pasaran, dan akhirnya memakai nilai resistor 2,2 k Ω untuk besaran tahanan basis.

• **Relay Driver Hitam**

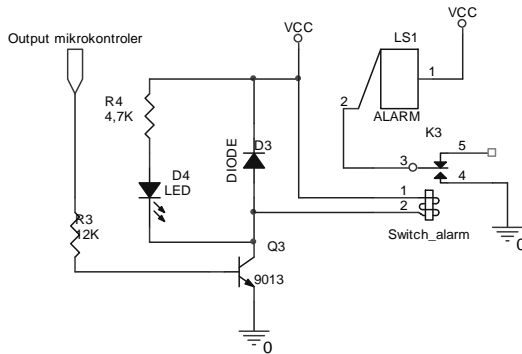
Rangkaian *relay driver* hitam disajikan pada Gambar 11. Adapun persamaan untuk menghitung arus listrik yang mengalir pada *relay driver* tersebut adalah sebagai berikut.

$$I_{rh} = \frac{V_{cc}}{R_{rh}} \quad (4)$$

$$I_{rh} = \frac{12}{400} = 0,03A$$

dengan:

I_{rh} = arus listrik dalam *relay* hitam;
 V_{cc} = tegangan listrik suplai 12 V DC;
 R_{rh} = tahanan listrik *relay* hitam.



Gambar 11. *Relay Driver* Hitam

Tahanan listrik basis (R_B), dihitung sebagai berikut.

$$I_{rh} = I_{Cmaks} = 0,03A$$

$$\beta_{DC} = 100$$

$$I_{Bmaks} = \frac{I_{Cmaks}}{100} = \frac{0,03}{100} = 0,0003A$$

$$R_{Bmaks} = \frac{V_{BB} - 0,7}{I_{Bmaks}} = \frac{4,5 - 0,7}{0,0003} = 12\,666,66 \Omega$$

dengan:

I_{cmaks} = arus listrik kolektor maksimal;
 β_{DC} = beta DC;
 I_{Bmaks} = arus listrik basis maksimal;
 V_{BB} = tegangan listrik *input* basis;
 R_{Bmaks} = tahanan listrik pada basis.

Resistor dengan nilai tahanan listrik 12 666,66 Ω tidak ada di pasaran, dan akhirnya memakai nilai resistor 1,2 k Ω atau 1,5 k Ω untuk besaran tahanan basis.

• **Relay putih**

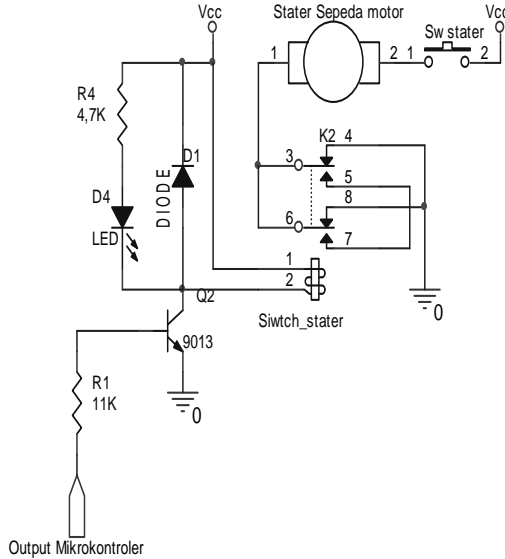
Rangkaian *relay driver* putih disajikan pada Gambar 12. Adapun perhitungan untuk mencari arus listrik yang mengalir dalam *relay* tersebut adalah sebagai berikut.

$$I_{rp} = \frac{V_{cc}}{R_{rp}} \quad (5)$$

$$I_{rp} = \frac{12}{375} = 0,032 A$$

dengan:

I_{rp} = arus listrik dalam *relay* putih;
 V_{cc} = tegangan listrik suplai 12V DC;
 R_{rp} = tahanan listrik dalam *relay* putih.



Gambar 12. Relay Driver Putih

Tahanan basis (R_B) dalam *relay driver* putih dihitung sebagai berikut:

$$I_{rp} = I_{C maks} = 0,32 \text{ mA}$$

$$B_{DC} = 100$$

$$R_{Bmaks} = \frac{V_{BB} - 0,7}{I_{Bmaks}} = \frac{4,5 - 0,7}{0,0003} = 12\ 666,66 \ \Omega$$

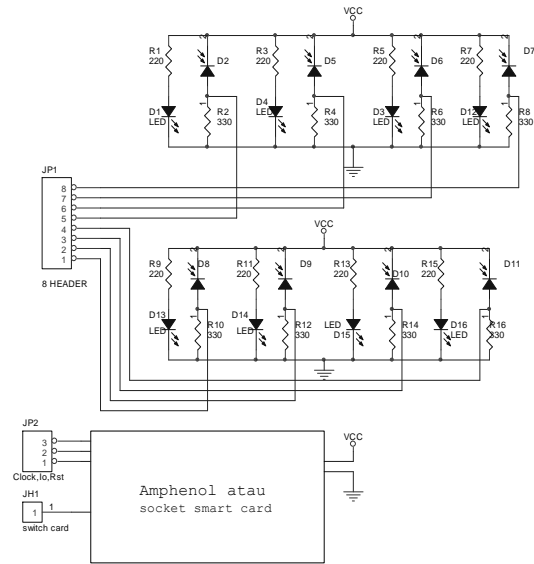
dengan:

$I_{c max}$ = arus listrik kolektor maksimal;
 β_{DC} = beta DC;
 $I_{B maks}$ = arus listrik basis maksimal;
 V_{BB} = tegangan listrik *input* basis;
 $R_{B maks}$ = tahanan listrik pada basis.

Resistor dengan nilai tahanan listrik sebesar 12 666,66 Ω tidak ada di pasaran, dan akhirnya dipakai nilai resistor 1,2 k Ω untuk besaran tahanan basis.

Perancangan Smart Card Reader

Rangkaian *smart card reader* disajikan pada Gambar 13. *Smart card reader* dirancang untuk membaca *chip SLE4442* dan warna hitam pada *smart card*. Ada 2 bagian dalam pembacaan *smart card* sebagaimana dijelaskan sebagai berikut:



Gambar 13. Smart Card Reader

Amphenol atau Socket Smart Card

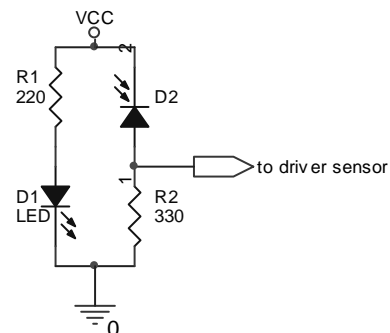
Amphenol/socket *smart card* disajikan pada Gambar 14, yang berfungsi sebagai penghubung antara mikrokontroler, dan *chip SLE4442*.



Gambar 14. Amphenol atau Socket Smart Card.

Fotodiode (pembaca warna hitam)

Sensor ini digunakan untuk membaca gambar warna hitam yang ada pada *smart card*. Sensor pembaca warna hitam disajikan pada Gambar 15.



Gambar 15. Sensor Pembaca Warna Hitam

Smart Card

Smart card disajikan pada Gambar 16 sebagai berikut.

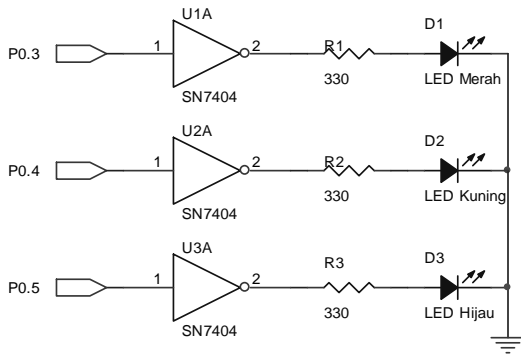


Gambar 16. Smart Card

Smart card yang akan dipakai dalam pembuatan alat pengaman sepeda motor ini adalah smart card SLE4442. Di samping memakai chip, smart card tersebut akan diberi warna hitam untuk pengaman chip itu sendiri.

Indikator LED

Light Emitting Diode atau disingkat LED di dalam alat pengaman sepeda motor digunakan sebagai indikator agar para pemakai mengetahui kondisi lock (warna merah), kondisi card is true (warna kuning), kondisi finish, dan sistem pada saat dimatikan (warna hijau). Gambar 17 di bawah ini adalah rangkaian indikator LED.



Gambar 17. Rangkaian Indikator LED

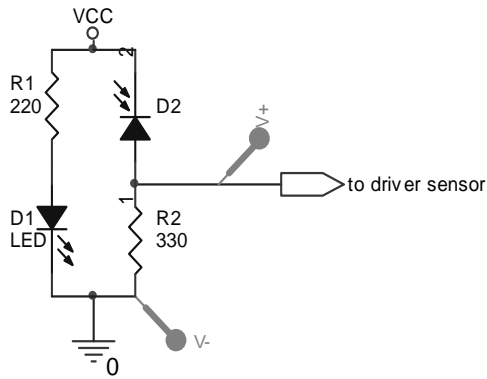
HASIL PENELITIAN DAN PEMBAHASAN

Pengukuran Rangkaian Sensor

Pada Gambar 18 ditunjukkan rangkaian sensor. Hasil pengujian dan pengukuran sensor adalah:

- o Tegangan R_2 pada saat high = 0,78V DC
- o Tegangan R_2 pada saat low = 0,45V DC.

Dari hasil pengukuran di atas, saat sensor tidak menerima cahaya pantulan dari pemancar, maka tegangan output R_2 = 0,45V DC. Pada saat menerima cahaya pantulan dari pemancar, maka tegangan output R_2 = 0,78 V DC. Dengan memakai rangkaian pembagi tegangan, maka sensor dapat bekerja dengan baik, walaupun selisih high, dan low-nya kecil.

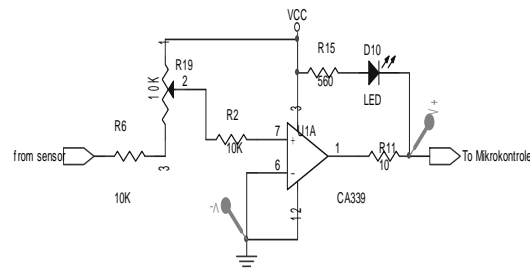


Gambar 18. Rangkaian Sensor

Pengukuran Sensor Driver

Pada Gambar 19 disajikan rangkaian sensor driver. Hasil pengujian dan pengukuran dari sensor driver adalah:

- o Tegangan output pada saat high = 3,56 V DC
- o Tegangan output pada saat low = 0,33 V DC



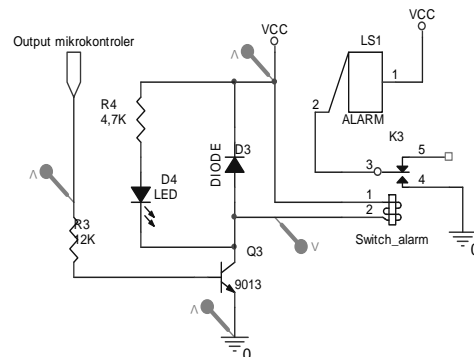
Gambar 19. Rangkaian Sensor Driver

Dari hasil pengukuran di atas, saat rangkaian sensor pada kondisi high, maka output sensor driver adalah low. Pada saat rangkaian sensor low, maka output sensor driver adalah high.

Pengukuran Relay Driver

Ada 3 pengukuran driver relay yang akan diukur. Rangkaian relay driver hitam disajikan pada Gambar 20.

• Relay Hitam



Gambar 20. Rangkaian Relay Driver Hitam

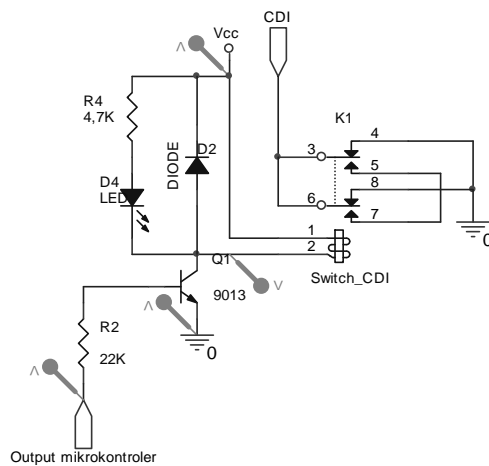
Hasil pengujian, dan pengukuran dari sensor *driver* adalah:

- Pengukuran *output* mikrokontroler pada saat *high* = 3,080 V DC;
- Pengukuran *output* mikrokontroler pada saat *low* = 71,2 mV DC
- Tegangan pada kaki +/- *relay* = 8,85 V DC.

Dari hasil pengukuran di atas, saat *ouput* mikrokontroler memberikan *low*, $V_{relay} = 0$ V, dan *relay* akan *open*. Pada saat *output* mikrokontroler memberikan *high*, $V_{relay} < 12$ V (seharusnya 12 V, hal ini dikarenakan $V_{P0.X}$ mengalami *drop* tegangan, dan β_{DC} 9013 \neq 100) akan tetapi dengan tegangan sebesar 8,85 V DC *relay* sudah berkondisi *close*, sehingga kondisi *relay* ini sesuai dengan yang diharapkan.

• Relay Kuning

Rangkaian *relay driver* kuning disajikan pada Gambar 21.



Gambar 21. Rangkaian Relay Driver Kuning

Hasil pengujian, dan pengukuran dari *relay driver* kuning adalah:

- Pengukuran *output* mikrokontroler pada saat *high* = 3,309 VDC;
- Pengukuran *output* mikrokontroler pada saat *low* = 71 mV;
- Tegangan pada kaki +/- *relay* = 11,54 VDC.

Dari hasil pengukuran di atas, saat *ouput* mikrokontroler memberikan *low*, $V_{relay} = 0$ V, dan *relay* akan *open*. Pada saat *output* mikrokontroler memberikan *high*, $V_{relay} < 12$ V (seharusnya 12 V, hal ini dikarenakan $V_{P0.X}$ mengalami *drop* tegangan dan β_{DC} 9013 \neq 100) akan tetapi dengan tegangan sebesar 11,54 V DC *relay* sudah berkondisi *close*,

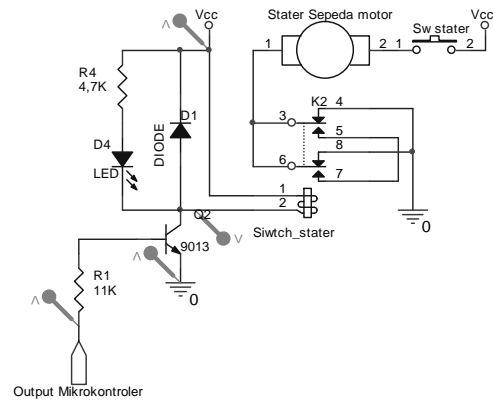
sehingga kondisi *relay* ini sesuai dengan yang diharapkan.

• Relay Putih

Rangkaian *relay driver* putih disajikan pada Gambar 22.

Hasil pengujian dan pengukuran dari *relay driver* putih adalah sebagai berikut:

- Pengukuran *output* mikrokontroler pada saat *high* = 2,882 VDC;
- Pengukuran *output* mikrokontroler pada saat *low* = 76,9 mV;
- Tegangan pada kaki +/- *relay* = 7,81 VDC.



Gambar 22. Rangkaian Relay Driver Putih

Dari hasil pengukuran di atas, saat *ouput* mikrokontroler memberikan *low*, $V_{relay} = 0$ V, dan *relay* akan *open*. Pada saat *output* mikrokontroler memberikan *high*, $V_{relay} < 12$ V (seharusnya 12 V, hal ini dikarenakan $V_{P0.X}$ mengalami *drop* tegangan dan $\beta_{DC} = 9013 \neq 100$), akan tetapi dengan tegangan sebesar 7,81 VDC, *relay* sudah berkondisi *close*, sehingga kondisi *relay* ini sudah sesuai dengan yang diharapkan.

KESIMPULAN

Dari hasil penelitian dan pembahasan yang melibatkan pengujian, dan pengukuran dari alat pengaman sepeda motor, maka dapat diambil kesimpulan yaitu:

1. *Smart card* dapat digunakan dengan baik;
2. Mikrokontroler dapat menjalankan tugasnya antara lain:
 - a. dapat membaca *output* dari sensor *driver*;
 - b. dapat mengecek *password* dalam *smart card*;
 - c. dapat mengontrol 3 *relay* dengan baik;
 - d. dapat memberi sinyal ke indikator LED;
3. Boks pengaman sepeda motor masih kurang efisien untuk ditempatkan di sepeda motor,

karena volumenya yang besar untuk ukuran sepeda motor.

DAFTAR PUSTAKA

- [1] Siemens, *ICs for Chips Cards Intelligent 256-Byte EEPROM SLE4442 datasheet*, <http://flylogic.net/docs/sle4442.pdf>, Diakses 17 Maret 2009
- [2] Mazidi, M. A., Mazidi, J. G., dan McKinlay, R. D., *The 8051 Microcontroller and Embedded Systems Using Assembly and C*, Hlm. 181–260, Pearson Education, New Jersey, 2006
- [3] Atmel, *Microcontroller AT89S51 Datasheet*, <http://www.atmel.com>, Diakses 8 April 2009
- [4] Coughlin, R. F., Frederick, F. D. dan Herman, W. S., *Penguat Operasional dan Rangkaian Terpadu Linear*, Hlm. 53–75, Erlangga, Jakarta, 1994
- [5] Malvino, A. P., dan Barmawi, M., *Prinsip-Prinsip Elektronika, Edisi Ketiga*, Jilid 1, Hlm. 111–140, Erlangga, Jakarta, 1984