

SISTEM KEAMANAN MOTOR MENGGUNAKAN TELEPON SELULAR BERBASIS KOMUNIKASI DUA ARAH

Robby Saleh¹; Ho ping²; Achmad.S³; Aulia.U⁴

^{1, 2, 3, 4}Jurusan Sistem Komputer, Fakultas Ilmu Komputer, Universitas Bina Nusantara,
Jl. K.H. Syahdan No. 9, Kemanggis/Palmerah, Jakarta Barat 11480
¹robby@binus.ac.id

ABSTRACT

Cellular phone nowadays can be used for security system on vehicle. One of the researches that has been done was using MMS/SMS to tell the owner of the vehicle that the vehicle was being stolen. The research purpose was to make security system on vehicle using AVR as controller and cellular phone to communicate, and the cellular phone that was on the vehicle was dialed directly to the owner informing that was being stolen. The used research method was literature and experimental method. The research result indicated that the security system has response time ± 20 seconds more faster than the system before (using MMS/SMS) that needs time 2-6 minutes.

Keywords: cellular phone, security, vehicle, two direction communication

ABSTRAK

Telepon selular (ponsel) saat ini dapat digunakan untuk sistem keamanan pada kendaraan. Salah satu penelitian yang pernah dilakukan adalah menggunakan MMS/SMS untuk memberitahu pemilik kendaraan jika pencurian terjadi. Penelitian ini bertujuan membuat sistem keamanan pada kendaraan menggunakan AVR sebagai controller dan ponsel untuk komunikasi, dan ponsel yang berada di kendaraan men-dial langsung pemilik kendaraan jika terjadi pencurian. Metode penelitian menggunakan studi literatur dan eksperimental. Hasil penelitian adalah sistem keamanan memiliki response time sebesar ± 20 detik lebih cepat dari sistem sebelumnya (menggunakan MMS/SMS) yang membutuhkan waktu 2-6 menit.

Kata kunci: telepon selular, keamanan, kendaraan, komunikasi dua arah

PENDAHULUAN

Maraknya pencurian kendaraan bermotor (curanmor) di ibukota beberapa tahun belakangan telah menyebabkan keresahan dan menimbulkan kerugian sangat besar yang dialami masyarakat, terutama pemilik kendaraan. Pencuri tidak segan mencuri kendaraan yang berada di halaman parkir depan rumah, bahkan tidak jarang mereka mencuri kendaraan yang berada dalam garasi rumah. Hal ini dapat terjadi karena beberapa aspek penting yang dilupakan oleh pemilik kendaraan bermotor, di antaranya adalah kurangnya kewaspadaan terhadap kejadian yang mungkin terjadi pada kendaraan mereka dan kurangnya pengamanan kendaraan yang memungkinkan pencuri leluasa mencuri kendaraan korban.

Sistem keamanan pada motor dapat dikategorikan dalam dua jenis, yaitu sistem keamanan yang bersifat pasif dan sistem keamanan yang

bersifat aktif. Sistem keamanan pasif banyak digunakan oleh pemilik motor karena harganya terjangkau, juga sistem ini dapat melindungi motor agar tidak dicuri. Sistem yang lain adalah sistem keamanan yang bersifat aktif, yaitu menggunakan alarm dan semua sistem keamanan yang terbuat dari elektronik. Alarm mempunyai kelebihan dibandingkan dengan peralatan kunci konvensional seperti kunci stang ataupun gembok cakram karena alarm dapat memberikan suatu peringatan dini pada pemilik kendaraan dan orang yang berada di sekitar lokasi motor.

Sistem keamanan kendaraan menggunakan telepon selular (ponsel) (Arief, Bambang, Ignassius, pada tahun 2005) telah dikembangkan dalam lingkungan Universitas Bina Nusantara. Pada penelitian sebelumnya hasil yang diperoleh belum sempurna, seperti sumber daya yang terbatas, yaitu menggunakan baterai ponsel dan menggunakan MCS-52 yang tidak mempunyai EEPROM. Penelitian tersebut menggunakan media MMS

(*Multimedia Message Service*) untuk mengirimkan gambar foto kondisi di dalam mobil/kendaraan untuk menghindari hal yang tidak diinginkan. Media MMS mempunyai respons sistem yang cukup lama antara 2-6 menit, tergantung dari lancarnya komunikasi dan waktu pengiriman. Pada penelitian ini, ponsel tidak lagi mengirimkan media SMS (*Short Messaging Service*)/MMS melainkan langsung *dial* ke pemilik motor dan pemilik motor juga dapat melakukan *missed call* balik ke ponsel motor. Dengan metode ini diharapkan sistem dapat merespons lebih cepat dari 2-6 menit, yaitu sekitar 20 detik sehingga pemilik kendaraan dapat dengan segera mengetahui apa yang terjadi.

PEMBAHASAN

Mikrokontroler AVR

Mikrokontroler merupakan salah satu bagian dasar dari suatu sistem komputer. Sebuah mikrokontroler dibangun dari elemen dasar yang sama dengan sebuah sistem komputer, namun mikrokontroler memiliki bentuk yang jauh lebih kecil. Pada penelitian ini, mikrokontroler yang digunakan adalah AVR ATmega8535.

AVR adalah keluarga mikrokontroler tipe RISC (*Reduced Instruction Set Computing*) produksi Atmel yang dikembangkan oleh dua orang mahasiswa bernama Alf-Egil Bogen dan Vegard Wollan di Norwegian Institute of Technology yang kemudian diteruskan oleh Atmel Norwegia. Nama AVR telah disebut-sebut sebagai singkatan dari *Advanced Virtual RISC*, namun ada juga yang mengatakan bahwa nama tersebut berdasarkan nama para penemunya, yaitu Alf dan Vegard. Perkembangan terbaru dari Atmel, termasuk keluarga AVR32 yang merupakan prosesor RISC 32 bit dan mempunyai instruksi SIMD dan DSP. Selain itu, AVR32 juga mempunyai fitur tambahan untuk audio dan video *processing* untuk dapat bersaing dengan prosesor ARM yang telah mendominasi pasaran. Adapun prosesor yang akan dipergunakan pada penelitian ini merupakan sebuah prosesor AVR 8 bit.

AT Command

Pada komunikasi dengan *AT Command*, ponsel akan berfungsi sebagai alat penghubung antara kendaraan dengan pemilik. Ponsel itu dapat untuk bertukar data, gambar, jadwal, dan nada dering melalui koneksi *Bluetooth* dan *irDA*. Dalam penelitian ini yang digunakan adalah koneksi kabel

RS-232 untuk komunikasi serial. Pada saat ponsel dihubungkan dengan modul serial maka ponsel beroperasi sebagai *Data Circuit-terminating Equipment* (DCE) dan AVR beroperasi sebagai *Data Terminal Equipment* (DTE).

Dalam penelitian ini, *AT Command* yang digunakan adalah *AT Command* untuk Sony Ericsson T68i. *Command-command* yang digunakan, antara lain:

ATD → untuk mendial nomor ponsel pemilik kendaraan

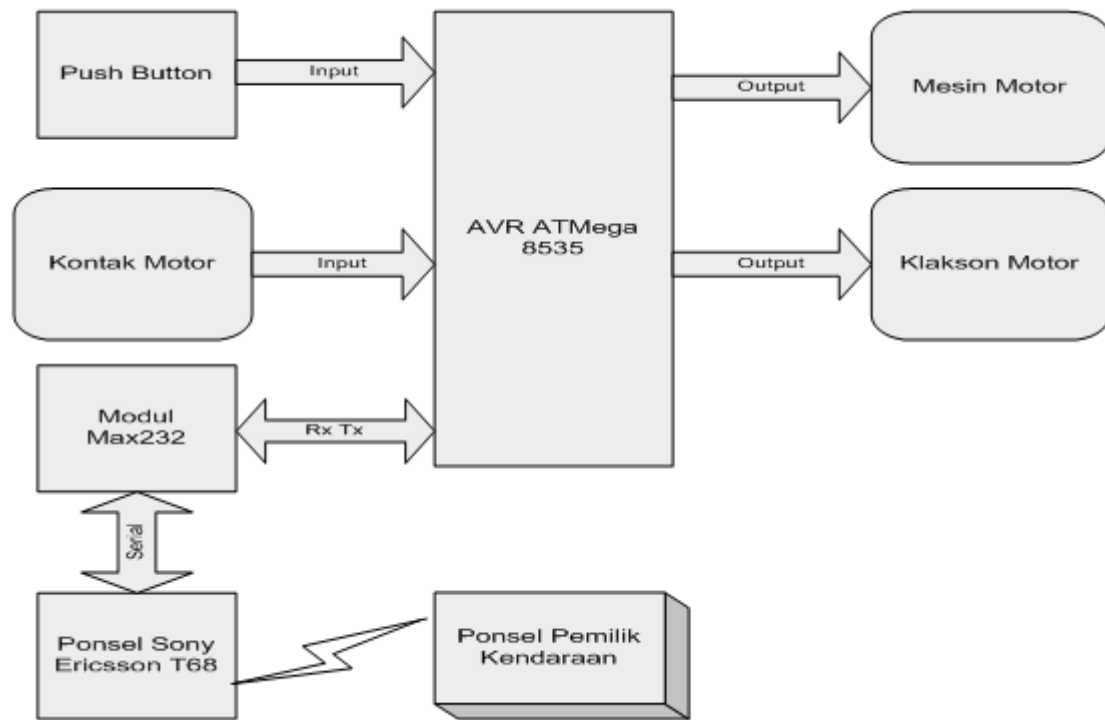
ATH → untuk menutup panggilan yang sedang berlangsung

Perancangan Perangkat Keras

Blok-Diagram Sistem

Secara garis besar, sistem ini dibuat selalu menyala atau *standby* agar dapat menerima *input* dari kontak motor pada saat kapan pun sehingga jika ada yang melakukan kontak pada motor maka *interrupt* pada AVR akan mendapat *input*. Selanjutnya, *interrupt* akan mengaktifkan *register timer* agar mulai menghitung dari 0 sampai 10 detik. Jika tidak ada *input* yang valid dari *push button* maka AVR akan *men-dial* nomor ponsel pemilik selama 30 detik, mesin motor akan mati kemudian klakson pada motor akan berbunyi. Jika pemilik kendaraan mendapat panggilan dari ponsel kendaraan, pemilik harus merespons dengan menelpon ke ponsel yang ada di motor agar dapat mematikan fungsi *dial* dari program. Setelah melakukan telepon ke ponsel yang ada di motor pemilik harus memasukkan kode pengaman yang valid ke modul *push button* agar validasi dapat dilakukan sehingga motor akan kembali dalam keadaan normal.

Modul AVR digunakan sebagai modul utama dari sistem ini. Modul serial digunakan sebagai komunikasi antara AVR dengan ponsel. Modul serial digunakan agar AVR dapat memberi instruksi kepada ponsel untuk menghubungi pemilik kendaraan, modul serial menggunakan IC MAX 232. Modul *relay* digunakan agar AVR dapat mematikan mesin dan menyalakan klakson pada motor. Modul *relay* dibuat dengan IC ULN 2003. Tiap *relay* mengatur satu bagian dari motor, yaitu mesin dan klakson. *Push button* digunakan untuk pemilik memasukkan *input* ke AVR. *Push button* yang digunakan sebanyak 4 buah sehingga bentuknya dapat disesuaikan. Kontak motor agar dapat diterima sebagai *interupsi* oleh AVR dikonversi menggunakan LM 7805 (Lihat gambar 1).

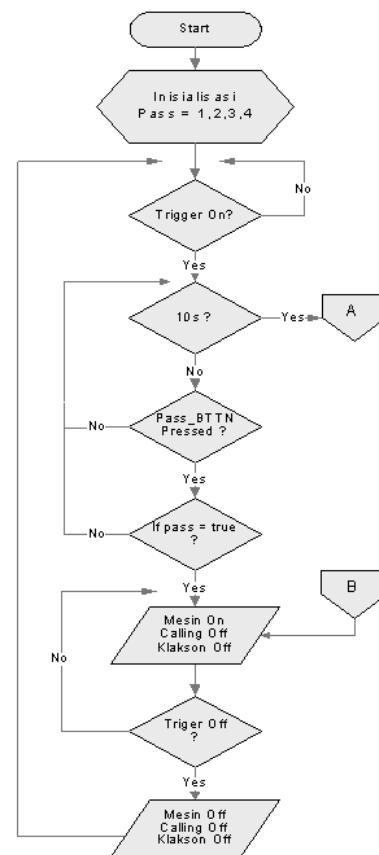


Gambar 1 Blok Diagram Sistem

Diagram Alir Kode Pengaman Benar

Perancangan Perangkat Lunak

Tahap perancangan perangkat lunak atau program pada AVR menggunakan bahasa C dan menggunakan register dari AVR. Bahasa C sebagai program utama sistem ini C digunakan untuk merancang fungsi pada AVR yang akan berperan sebagai pengatur sebagian besar fungsi sistem. Perintah antara AVR dengan ponsel dalam penelitian ini menggunakan *AT Command* yang dikirim secara serial dengan AVR. Algoritma yang digunakan pada sistem ini adalah apabila ada yang melakukan kontak pada motor lalu menghidupkan mesin tanpa memasukkan kode pengaman maka dalam 10 detik mesin motor akan mati lalu klakson pada motor akan menyala. Ponsel pada motor akan *men-dial* pemilik yang kemudian akan ditanggapi oleh pemilik dengan melakukan *missed call* ke ponsel kendaraan. Setelah melakukan *missed call* pemilik harus memasukkan kode pengaman dengan benar maka sistem keamanan ini akan berjalan normal kembali. Dalam perancangannya, sistem ini dibagi dua validasi, yaitu validasi benar dan validasi salah. Pada tiap validasi ada sedikit perbedaan cara untuk memenuhi nilai validasinya agar suatu *state* menjadi benar atau salah.



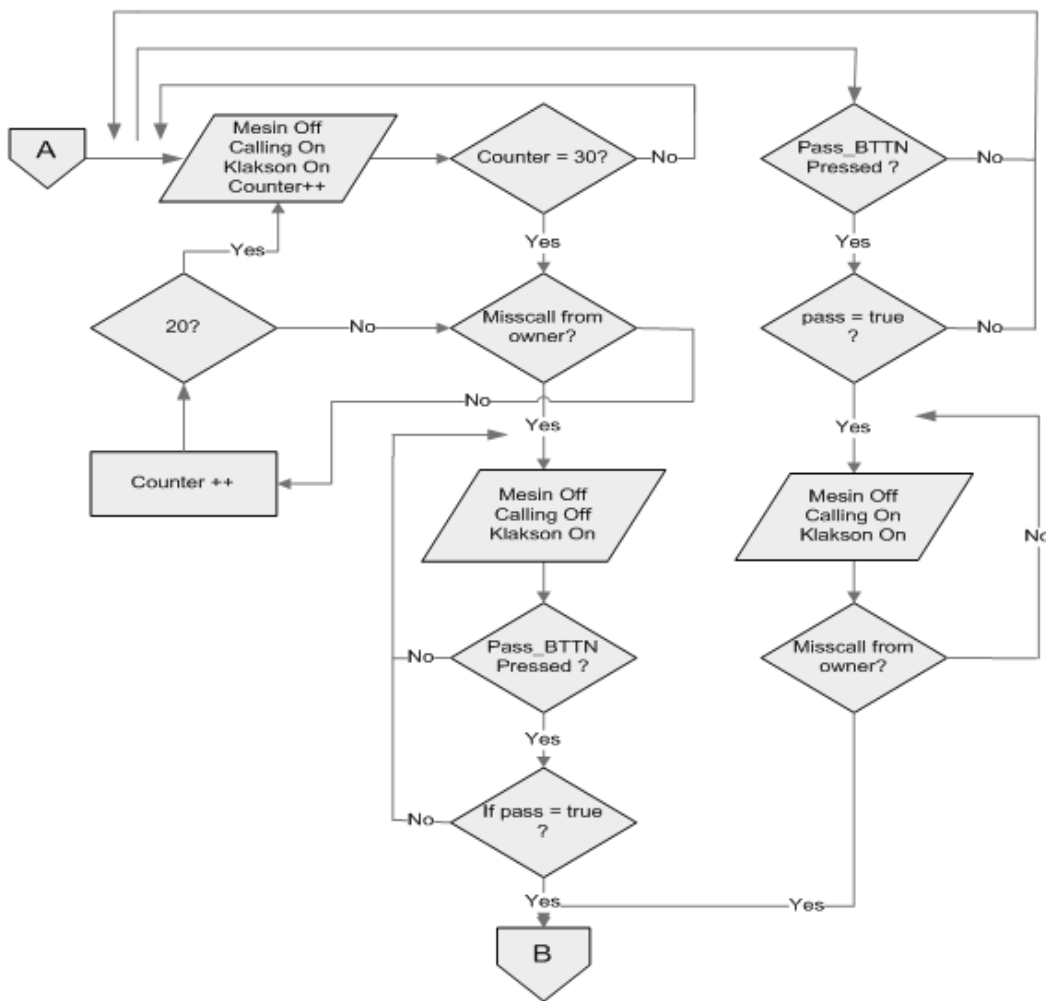
Gambar 2 FlowChart Validasi Benar

Tahap awal sistem ini bekerja adalah menginisialisasi kode pengaman, yaitu 1,2,3,4. Kode pengaman dapat disesuaikan tergantung dari pemilik menginginkan berapa banyak kode yang diinginkan menjadi kode pengaman. Kode pengaman akan disimpan di dalam *eeeprom* AVR sehingga saat di-*reset* nilai dari kode pengaman tidak akan berubah, kecuali diprogram ulang. Setelah tahap inisialisasi program menunggu *input* dari kontak motor, dalam program AVR jika mendapat *input* berupa nilai *logic* '1' maka *timer* pada AVR akan mulai menghitung dari 0 sampai 10 detik. Logika *input* yang digunakan adalah *rising high* artinya dari *logic* yang awalnya '0' di ubah menjadi '1' akan meng-*interrupt* program AVR untuk mulai menghitung dan memberikan status 1 agar kontak tidak mulai dari awal jika diberi *trigger* lagi nantinya. Nilai status awal sistem ini jika di-*reset* maka akan bernilai 0, akan bernilai 1 jika mendapat *trigger* dari kontak. Setelah dapat *trigger* dari kontak

maka saat *timer* berjalan, program akan menunggu kode pengaman dimasukkan. Validasi kode pengaman dilakukan pada saat *push button* sudah memasukkan 4 kali *input* untuk divalidasi apakah nilai dari *input* yang diberikan sama dengan kode pengaman yang disimpan dalam *eeeprom*. Jika hasilnya sama maka proses validasi kode pengaman sudah selesai dilakukan, hasilnya *timer* akan berhenti berhitung lalu nilai *timer* akan dijadikan 0, status dijadikan 0 kembali dan klakson akan berbunyi setengah detik untuk menandakan bahwa hasil validasi benar. Motor akan dalam keadaan normal jika validasi benar ini dilakukan maka mesin motor tidak akan mati.

Apabila *trigger off* atau motor akan dimatikan maka keadaannya akan kembali pada saat awal tombol *reset* ditekan. Keadaan itu akan membuat sistem kembali *stanby* menunggu *interrupt* selanjutnya dilakukan.

Diagram Alir Kode Pengaman Salah atau Lewat 10 Detik



Gambar 3 FlowChart Validasi Salah

Flow chart ini menjelaskan apabila *timer* sudah lewat 10 detik atau pemilik memasukkan kode pengaman salah dan lewat dari 10 detik maka program akan masuk ke dalam validasi salah. Dalam validasi ini mesin motor akan mati, klakson motor menyala, dan ponsel yang pada kendaraan akan menelepon pemilik kendaraan. AVR akan men-*dial* pemilik kendaraan selama 30 detik, setelah 30 detik kemudian AVR akan menutup panggilan yang sedang dilakukan. Pada saat ini, program akan menunggu *missed call* yang dilakukan pemilik ke ponsel yang ada pada kendaraan. Lama menunggunya diberi waktu selama 20 detik, jika lewat dari waktu yang diberikan tidak ada *missed call* yang masuk maka AVR akan kembali men-*dial* pemilik kendaraan. *Missed call* yang masuk akan diberi nilai 1 untuk memberi *flag* kepada program agar tidak men-*dial* ulang ke pemilik kendaraan. Setelah dapat *missed call*, program akan menunggu kode pengaman dimasukkan dengan menekan *push button*. *Input* akan divalidasi jika sudah ditekan 4 kali, jika nilainya sama dengan yang disimpan pada *eprom* maka akan diberi nilai 1 oleh program. Proses selanjutnya adalah melakukan validasi dua *input* yang diberikan dengan logika '*and (&&)*'. Apabila sesuai dengan logika yang ada maka keadaannya mesin motor akan dapat dinyalakan kembali, klakson motor akan normal, dan status akan diberi nilai 0.

Cara lainnya adalah dengan mendahulukan kode pengaman dimasukkan kemudian melakukan *misscall*. Kode pengaman dimasukkan dengan cara menekan *push button* sebanyak 4 buah sebagai *input*. Program akan menyocokkan dengan nilai yang diberikan sebagai *input* dengan yang nilai yang ada pada *eprom* pada AVR. Apabila hasilnya sama maka akan diberi nilai 1 oleh program. Dalam validasi ini memasukkan kode pengaman tidak akan membuat AVR berhenti mendial pemilik kendaraan. Artinya, program akan terus menerus men-*dial* ke pemilik kendaraan untuk meminta untuk segera di-*missed call*. Jika *missed call* sudah didapat maka program akan memberi nilai 1 untuk nantinya akan dilakukan validasi dua *input*. Validasi dua *input* dilakukan dengan operator logika '*and (&&)*' sehingga jika salah satu masih bernilai 0 maka nilai dari logika itu belum valid. Apabila dua *input* sudah memenuhi validasi '*and*' maka mesin akan dapat dinyalakan kembali, klakson akan kembali normal, dan status akan kembali menjadi 0. *Reset* juga menjadi salah satu validasi *input* apabila pemilik lupa dengan kode pengaman, dengan menekan tombol *reset* program akan memulai dari awal sehingga keadaan motor menjadi normal.

Pertimbangan untuk menggunakan dua validasi pada saat salah dan lewat dari 10 detik adalah untuk menghindari aksi coba-coba dari orang yang bukan pemilik kendaraan. Selain itu, satu validasi sangat lemah dalam hal keamanan maka digunakan dua validasi dalam sistem keamanan ini.

Implementasi dan Evaluasi

Setelah melalui beberapa tahapan sebelumnya, yaitu mengadakan penelitian melalui metode penelitian kepustakaan, penelitian laboratorium, perancangan algoritma sistem keamanan beserta perangkat lunak dan perancangan perangkat keras, tahap selanjutnya adalah pengimplementasian sistem. Data hasil pengukuran didapat dengan melakukan suatu pengujian, meliputi waktu yang dibutuhkan untuk 1 siklus pengoperasian dan jalannya sistem secara keseluruhan. Pengujian yang dilakukan didasarkan pada apakah sistem yang dirancang dapat berjalan sesuai dengan yang diinginkan.

Pengujian pada Modul *Push Button*

Dalam pengujian dengan durasi 1-2 detik atau cepat, pemilik sedikit mengalami kesulitan dalam memasukkan kode pengaman. Hal itu disebabkan adanya beberapa *push button* yang tidak bereaksi ketika ditekan oleh pemilik sehingga rata-rata data keberhasilan menjadi 80%. Faktor lain adalah pemilik terlalu buru-buru memasukkan kode pengaman dan kecilnya modul menjadi salah satu kendala kegagalan memasukkan kode pengaman.

Pada saat pengujian dengan durasi 3-4 detik atau sedang, pemilik kendaraan dapat memasukkan kode pengaman dengan persentase keberhasilan hingga 100%. Hal itu karena pemilik kendaraan memiliki cukup waktu untuk memasukkan kode pengaman dengan benar.

Pengujian yang berikutnya adalah pengujian dengan durasi 4-5 detik atau lambat. Pemilik kendaraan juga dapat memasukkan kode pengaman dengan keberhasilan 100% tanpa mengalami kegagalan satupun. Faktor keberhasilan ini juga dapat disimpulkan bahwa pemilik kendaraan memiliki cukup waktu untuk memasukkan kode pengaman. Dengan pengujian ini dapat disimpulkan bahwa dengan durasi 3-4 detik sistem ini sudah dapat mengambil *input* yang kemudian diproses oleh AVR, pengujian tersebut bisa dilihat pada Tabel 1.

Tabel 1 Pengujian *Push Button*

Pengujian	Cepat 1-2 detik	Sedang 3-4 detik	Lambat 4-5 detik
1	Gagal	Berhasil	Berhasil
2	Berhasil	Berhasil	Berhasil
3	Berhasil	Berhasil	Berhasil
4	Gagal	Berhasil	Berhasil
5	Berhasil	Berhasil	Berhasil
6	Berhasil	Berhasil	Berhasil
7	Berhasil	Berhasil	Berhasil
8	Berhasil	Berhasil	Berhasil
9	Berhasil	Berhasil	Berhasil
10	Berhasil	Berhasil	Berhasil
Berhasil	80%	100%	100%

Percobaan Ponsel Men-*dial* ke Ponsel Pemilik Kendaraan

Pada percobaan ini, pengujian dilakukan dengan tiga waktu *peak time* yang berbeda, yaitu pagi, siang, malam. Hal itu karena dalam satu hari ada berbagai macam kendala tiap waktunya, jadi dibagi menjadi tiga percobaan. *Peak time* diasumsikan karena dalam satu hari ada waktu-waktu tertentu jaringan seluler berada dalam keadaan sibuk.

Dari percobaan tersebut, didapat waktu dari tiap-tiap *peak time* yang berbeda. Pada percobaan pagi hari, didapat rata-rata dari *dial* yang dilakukan sebesar 21,28 detik. Pada percobaan siang hari, didapat rata-rata dari *dial* yang dilakukan sebesar 20,73 detik. Pada percobaan malam hari, didapat rata-rata dari *dial* yang dilakukan sebesar 20,83 detik. Dari hasil percobaan *dial* di atas waktu terbaik dari ketiganya adalah pada siang hari dengan 20,73 detik (Lihat tabel 2).

Pengujian Respons Sistem

Pengujian ini dilakukan untuk mengetahui pada saat pemilik kendaraan mendapat panggilan telepon masuk dari ponsel yang ada pada kendaraan, lalu di-*reject* oleh pemilik yang kemudian langsung melakukan *missed call* ke ponsel yang ada pada

kendaraan. Pengujian ini dilakukan apakah *missed call* yang dilakukan oleh pemilik kendaraan diterima dengan sukses atau tidak (Lihat Tabel 3).

Tabel 4 menunjukkan respons sistem jika pemilik melakukan *missed call* balik pada detik ke berapa yang efektif dilakukan. Pada tabel *missed call* balik pada saat panggilan berlangsung, detik ke-5 dan 10 jika panggilan dari ponsel kendaraan di-*reject* oleh pemilik dan dilakukan panggilan beberapa detik kemudian ke ponsel yang ada pada kendaraan. Dihasilkan jika pada detik ke-5 *reject* dilakukan lalu telepon pada detik ke-10 hasilnya nada sibuk. *Reject* pada detik ke-10 lalu dilakukan *missed call* pada detik ke-15 yang dihasilkan tetap nada sibuk. Berikutnya pada detik ke-15 dilakukan *reject* lalu dilakukan *missed call* balik pada detik ke 20 hasil yang didapatkan diterima oleh sistem. Tabel *missed call* pada saat panggilan selesai dilakukan tersebut menunjukkan saat panggilan selesai dilakukan oleh sistem lalu dilakukan *missed call* pada detik ke-5 maka hasil yang didapatkan adalah diterima oleh sistem. Lalu pada detik ke-10 setelah panggilan dari sistem selesai dilakukan *missed call* ke ponsel kendaraan, hasil yang didapat adalah diterima oleh sistem. Percobaan selanjutnya dilakukan pada detik ke-15 setelah panggilan berakhir coba dilakukan *missed call* ke ponsel yang ada pada kendaraan, hasil

yang didapat adalah nada sibuk dari ponsel kendaraan. Terakhir adalah melakukan *missed call* pada detik ke-20 setelah panggilan dari ponsel kendaraan selesai melakukan panggilan, hasilnya adalah nada sibuk.

Sebab dari nada sibuk karena ponsel yang ada pada kendaraan masih melakukan panggilan ke nomor pemilik kendaraan. Jika pemilik kendaraan *me-reject* panggilan dari ponsel yang ada pada kendaraan pada detik ke-5 dan 10, ponsel yang ada pada kendaraan akan masuk ke dalam panggilan *mail box* dari ponsel pemilik kendaraan sehingga hasil yang didapat adalah nada sibuk pada ponsel pemilik yang melakukan *missed call*.

Dari percobaan tersebut dapat dilihat waktu yang efektif untuk melakukan *missed call* ke ponsel yang ada pada kendaraan agar dapat diterima. Hal yang pertama saat panggilan dari ponsel kendaraan berlangsung lalu dilakukan *reject* maka waktu yang efektif adalah pada saat detik ke-15 dan 20. Hal itu karena pada jeda waktu ini, ponsel yang ada pada kendaraan dalam keadaan menunggu *missed call* dari pemilik. Hal yang kedua adalah setelah panggilan selesai dilakukan, pemilik kendaraan melakukan *missed call* ke ponsel kendaraan. Dari data yang didapat *missed call* balik yang efektif dilakukan adalah pada detik ke-5 dan 10 setelah panggilan dari ponsel pada kendaraan selesai dilakukan. Hal itu karena ponsel yang ada pada kendaraan masih dalam keadaan menunggu *missed call* dari pemilik kendaraan.

Pengujian *Timer* terhadap Jarak

Dalam pengujian ini, sistem diuji dengan mencoba menjalankan motor dengan kecepatan berbeda-beda agar mendapat hasil minimum dan maksimum dari jarak yang ditempuh oleh motor. Data yang dihasilkan untuk melihat seberapa jauh motor bergerak dari tempat asal parkir ke tempat motor berhenti setelah 10 detik. Pengujian dilakukan pada tempat parkir Universitas Bina Nusantara Kampus Syahdan.

Dari pengujian jarak tempuh terhadap *timer*, didapat data dari jarak minimal yang ditempuh dari 1 sampai 10 detik dengan kecepatan konstan 10 Km/jam adalah 27,8 meter. Jarak maksimal yang dihasilkan dengan kecepatan 40 Km/jam adalah 111,1 meter. Pengujian *timer* terhadap jarak ini mempunyai tujuan untuk memperkirakan seberapa jauh pencuri dapat membawa lari motor dari tempat yang awal motor di parkir. Dengan tabel pengujian ini diharapkan dapat membantu pemilik kendaraan jika suatu waktu motornya dibawa lari oleh pencuri, pemilik kendaraan akan dapat memperkirakan seberapa jauh motornya akan berada. Data pada Tabel 5 dihitung berdasarkan pergerakan motor dijalan lurus dengan kecepatan yang berbeda-beda.

Table 2 Percobaan *Dial*

Percobaan	Waktu yang didapatkan / detik (8.00 Pagi)	Waktu yang didapatkan/ detik (12.00 Siang)	Waktu yang didapatkan/ detik (20.00 Malam)
1	21,35	20,59	20,447
2	21,75	20,78	22,289
3	21,17	20,38	21,218
4	21,45	20,69	20,798
5	21,96	20,18	20,692
6	21,50	20,46	20,54
7	20,74	20,69	20,489
8	20,74	21,16	19,686
9	20,99	21,54	21,176
10	21,24	20,88	20,927
Jumlah	212,89	207,35	208,262
Waktu rata-rata	21,28	20,73	20,82

Tabel 3 Respons *Missed call* Balik saat Panggilan Berlangsung

Telepon Diterima (Jam)	Telepon di <i>Reject</i> (Jam)	<i>User Miss call</i> Balik (Jam)	Tanggapan <i>Miss call</i> (Jam)	Total Waktu (detik)	Hasil Panggilan
16:00:00.00	16:00:05.29	16:00:10.45	16:00:16.22	16.22	Nada Sibuk
16:03:00.00	16:03:10.03	16:03:14.96	16:03:21.69	21.69	Nada Sibuk
16:05:00.00	16:05:15.29	16:05:19.89	16:05:28.65	28.65	Diterima
16:07:00.00	16:07:20.26	16:07:25.29	16:07:32.18	32.18	Diterima

Tabel 4 Respons *Missed call* Balik saat Panggilan Selesai Dilakukan

Telepon Selesai (Jam)	<i>User akan Miss call</i> (Jam)	Diterima Motor (Jam)	Total Waktu (detik)	Hasil Panggilan
16:10:00.00	16:10:05.28	16:10:16.78	16,78	Diterima
16:15:00.00	16:15:10.36	16:15:17.78	17,78	Diterima
16:20:00.00	16:20:15.05	16:20:24.46	24,46	Nada Sibuk
16:25:00.00	16:25:20.41	16:25:28.53	28,53	Nada Sibuk

Tabel 5 Pengujian *Timer* terhadap Jarak

Pengujian	<i>Timer</i>	Kecepatan Motor	Jarak
1	1-10 detik	10 Km/jam	27,8 m
2	1-10 detik	20 Km/jam	55,5 m
3	1-10 detik	30 Km/jam	83,3 m
4	1-10 detik	40 Km/jam	111,1 m

PENUTUP

Berdasarkan hasil evaluasi terhadap sistem, dapat diambil simpulan sebagai berikut Pertama, terjadi peningkatan respons 20 detik jika dibandingkan sistem keamanan metode *dial* dengan metode MMS sehingga pemilik kendaraan dapat mengambil tindakan lebih lanjut. Kedua, sistem keamanan menggunakan ponsel dapat lebih mudah diketahui oleh pemilik kendaraan karena pemilik dapat mengetahui langsung hanya dengan melihat panggilan pada ponsel. Ketiga, pemilik harus menunggu 10 detik setelah mendapat panggilan dari ponsel kendaraan untuk dapat melakukan validasi *missed call*.

DAFTAR PUSTAKA

- Anonymous. 2003. "SE T68I ." Diakses dari http://studiohp.com/phone_detail.php
- _____. 2006. "Hayes command set (AT Command)." Diakses dari http://en.wikipedia.org/w/index.php?title=AT_command&redirect=no
- _____. 2007. "Global System for Mobile Communications." Diakses dari <http://id.wikipedia.org/w/index.php?title=GSM&redirect=no>
- _____. 2007. "Serial Communication." Diakses dari <http://en.wikipedia.org/wiki/LED>
- _____. 2007. "What does GSM mean in a cell phone?." Diakses dari <http://electronics.howstuffworks.com/preempt/ros/electronics/electronics-telecomm/question537.htm>
- Arief, Bambang, Ignasius. 2005. "Sistem Keamanan Kendaraan dengan Menggunakan Telepon Seluler Berbasis MMS." Skripsi S1. Jurusan Sistem Komputer Universitas Bina Nusantara. Jakarta.
- Barnet, Cox and O'Cull. 2003. *Embedded C Programming and The Atmel AVR*. Canada: Thomson Learning.
- Eddy Yuliarso. 1998. "Sistem Telepon Selular Digital GSM." <http://www.elektroindonesia.com/elektro/maj1-8.html>
- Istiyanto, Jazi Eko. 2004. "Rancangan Dan Implementasi Prototipe Sistem Kendali Jarak Jauh Berbasis AT89C52 dan Layanan SMS GSM." Jurusan Fisika FMIPA UGM.
- Kuhnel and Clause. 1998. *AVR RISC Microcontroller Handbook*. Newnes.
- Prayoe, Atharis S. 2005. "Teknologi Komunikasi Tanpa Kabel." <http://dusunlaman.blog.com/304698/#cmts>
- Waradana, Lingga. 2006. *Belajar Sendiri Mikrokontroler AVR seri ATmega 8535 Simulasi, Hardware, dan Aplikasi*. Yogyakarta: Andi.