

# Simulasi Kendali Pintu Perlintasan dan Pemberitahuan Kedatangan Kereta Api Otomatis Menggunakan Sensor *Optocoupler* dan SMS Gateway pada Stasiun Kereta Api Kebayoran

**Achmad Solichin<sup>1</sup>, Achmad Ardiansyah<sup>2</sup>**

Fakultas Teknologi Informasi, Universitas Budi Luhur

Jl. Ciledug Raya, Petukangan Utara, 12260, Jakarta

Telp. 021 5853753 ext 305, HP. 0856 8198436

Email: achmad.solichin@budiluhur.ac.id<sup>1</sup>

ancahmaricah@gmail.com<sup>2</sup>

## **Abstrak**

*Kebayoran railway station is located in South Jakarta and flanked by two stations, Pondok Ranji and Palmerah. The problems that occurred in there is process of arriving and departing trains that not optimal. It still rely on telephone communication with other station personnel. For the passengers, there is no notification about the arrival of the train. A predefined schedule often does not match the actual arrival of the train, so they had to wait quite a long time at the station. Therefore, in this study, we designed an applications that control the door rail and provide arrival notification that works automatically and computerized. The result of this study is an application that's monitor the arrival of the railway, will close or open the door rail automatically and turn on the siren. It will also tell train passengers over the loudspeakers at the time of arrival and departure of trains in the Railway Station. In terms of communication with other stations, the application also comes with a SMS Gateway which will notify the departure of the train to the next station via SMS. This application used optocoupler sensor that is placed before and after the door crossings on railway sleepers. Passing train will activate the sensor and then processed into a number of instructions by the application. This application is expected to reduce the accident at the door of railway crossings, improving the quality of services and can help train officers to work more efficiently, especially at the Kebayoran Railway Station*

**Kata kunci:** railway station, optocoupler, sms gateway, interface.

## **1. PENDAHULUAN**

Salah satu sarana transportasi darat yang banyak digunakan oleh masyarakat adalah kereta api. Menurut data dari Direktorat Jenderal Perkeretaapian, untuk wilayah Jawa dan Sumatra jumlah pengguna jasa angkutan kereta api mencapai 201.930.000 penumpang (Perkeretaapian, 2010). Salah satu fasilitas yang tidak terpisahkan dalam penyelenggaraan transportasi kereta api adalah stasiun kereta api. Stasiun kereta api merupakan tempat dimana penumpang dapat naik dan turun dalam memakai transportasi kereta api. Stasiun

Kereta Api Kebayoran merupakan salah satu stasiun kereta api yang ada di wilayah Jakarta Selatan.

Stasiun kereta api merupakan sarana yang penting, akan tetapi layanan yang diberikan bagi para penumpang kereta api masih belum optimal. Saat ini masih terdapat beberapa kelemahan antara lain proses kedatangan dan keberangkatan kereta api yang masih mengandalkan komunikasi telepon dengan petugas stasiun lainnya. Pemberitahuan kedatangan kereta api akan disampaikan melalui telepon oleh petugas di stasiun sebelumnya kepada stasiun tujuan. Saat

menerima pemberitahuan kedatangan, petugas stasiun akan menutup palang pintu kereta api. Proses tersebut dapat memiliki kelemahan yaitu jika terjadi kelalaian petugas dapat menyebabkan kecelakaan. Selain itu, bagi penumpang kereta api yang akan naik di stasiun kereta api Kebayoran juga tidak terdapat pemberitahuan pasti terkait kedatangan kereta api. Jadwal yang sudah ditetapkan seringkali tidak sesuai dengan kedatangan kereta api sehingga penumpang harus menunggu cukup lama di stasiun.

Oleh karena itu, dalam penelitian ini dirancang suatu aplikasi kendali pintu perlintasan dan pemberitahuan kedatangan kereta api yang bekerja secara otomatis dan terkomputerisasi. Dalam penelitian ini dihasilkan aplikasi yang men-simulasikan kondisi sebenarnya. Aplikasi tersebut berfungsi untuk memantau kedatangan kereta api, selanjutnya akan menutup atau membuka palang pintu perlintasan dan menyalakan sirene secara otomatis. Aplikasi ini juga akan memberitahu penumpang kereta api melalui pengeras suara pada saat kedatangan dan keberangkatan kereta api di Stasiun Kereta Api Kebayoran.

Dalam hal komunikasi dengan stasiun lainnya, aplikasi juga dilengkapi dengan *SMS Gateway* yang akan memberitahukan keberangkatan kereta api ke stasiun selanjutnya melalui *SMS (Short Messaging Service)*. *Interface* yang digunakan dalam aplikasi ini menggunakan sensor *optocoupler* yang diletakkan sebelum dan sesudah pintu perlintasan pada bantalan rel kereta api. Kereta api yang lewat akan mengaktifkan sensor dan selanjutnya diolah menjadi sejumlah instruksi oleh aplikasi. Aplikasi dibuat dengan bahasa pemrograman Java dan untuk menghubungkan antara sensor dengan komputer melalui *parallel port* menggunakan kabel data DB-25.

Dengan adanya aplikasi ini diharapkan dapat mengurangi tingkat

kecelakaan pada pintu perlintasan kereta api, meningkatkan mutu pelayanan kepada penumpang kereta api dan dapat membantu pekerjaan petugas kereta api untuk menjalankan tugasnya secara lebih efisien terutama pada Stasiun Kereta Api Kebayoran.

## **2. LANDASAN TEORI**

### **2.1 *Parallel Port* pada Komputer**

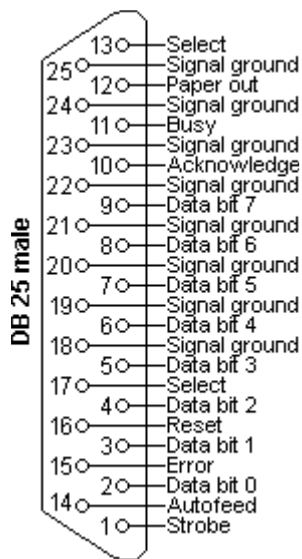
*Parallel port* merupakan antarmuka suatu dari sistem komputer yang berfungsi mentransfer data secara *parallel*. *Parallel port* biasanya digunakan untuk menghubungkan komputer dengan *printer*. *Port* ini membolehkan kita memiliki masukan hingga 8 bit atau keluaran hingga 12 bit pada saat yang bersamaan, dengan hanya menambahkan rangkaian eksternal sederhana. *Parallel port* merupakan salah satu jenis *socket* pada komputer untuk berkomunikasi dengan peralatan luar untuk mengirim data digital.

*Parallel port* distandarisasi oleh standar *IEEE (Institute of Electrical and Electronics Engineers)* 1284 pada tahun 1994 (Wikipedia). Tujuan standarisasi ini adalah untuk mendesain *driver* dan peralatan yang baru sehingga kompatibel dengan peralatan lainnya dan standar *parallel port* sebelumnya. Standar ini membedakan model operasi sebagai berikut:

- *Compatible mode*
- *Nibble mode*
- *Byte mode*
- *EPP mode*
- *ECP mode*

*Compatible mode*, *nibble mode*, dan *byte mode* digunakan sebagai standar perangkat keras yang tersedia di *parallel port original*. Sedangkan untuk *EPP mode* dan *ECP mode* membutuhkan tambahan perangkat keras sehingga mampu bekerja dengan kecepatan tinggi. *Compatible mode* hanya dapat mengirim data dari komputer yang mempunyai

kecepatan berkisar antara 80 Kbyte per detik. Agar komputer dapat menerima suatu data, *nibble mode* atau *byte mode* harus diterapkan. *Nibble mode* dapat menerima data sepanjang 1 *nibble* (4 bit), sementara *Byte mode* dapat menerima data sepanjang 1 byte (8 bit). Gambar 1 adalah gambar konektor parallel port DB-25 yang banyak digunakan pada komputer.



Gambar 1: Parallel DB25 pinout (Bies, 2010)

Secara umum LPT port terbagi atas tiga saluran yaitu saluran data (pin 2-9) yang bersifat output, saluran status (pin 15, 13, 12, 11 dan 10) yang bersifat input dan saluran kendali (pin 1, 14, 16 dan 17). Untuk melakukan *interfacing*, maka terlebih dahulu harus diketahui alamat dari ketiga saluran tersebut. *Interfacing* merupakan suatu cara kita menghubungkan komputer dengan piranti lain diluar komputer. Komputer dimanfaatkan sebagai pusat kendali untuk mengendalikan piranti luarnya. Misalnya: mengendalikan motor, *led*, dan lampu. Komputer juga dapat dimanfaatkan sebagai masukan dari sensor-sensor, saklar dan piranti input lain (Widodo, 2007). Masing-masing pin dari *parallel port* mempunyai fungsi seperti pada tabel 1. Kolom *SPP Signal* pada tabel 1 yang menggunakan huruf 'n' di depan nama sinyal menunjukkan bahwa sinyal

tersebut bersifat aktif rendah (*active low*).

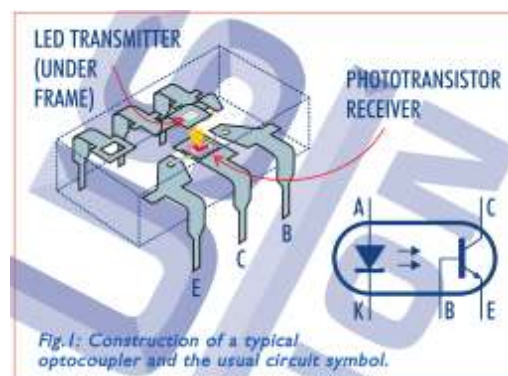
Tabel 1: Daftar Pin Parallel Port (DB-25)

Pin No (D-Type 25)	Pin No (Centronics)	SPP Signal	Direction In/Out	Register Control	Hardware Inverted
1	1	nStrobe	In/Out	Control	Yes
2	2	Data 0	Out	Data	
3	3	Data 1	Out	Data	
4	4	Data 2	Out	Data	
5	5	Data 3	Out	Data	
6	6	Data 4	Out	Data	
7	7	Data 5	Out	Data	
8	8	Data 6	Out	Data	
9	9	Data 7	Out	Data	
10	10	nAck	In	Status	
11	11	nBusy	In	Status	Yes
12	12	Paper-Out / Paper-End	In	Status	
13	13	Select	In	Status	
14	14	nAuto-Linefeed	In/Out	Control	Yes
15	32	nError / nFault	In	Status	
16	31	nInitialize	In/Out	Control	
17	36	nSelect-Printer / nSelect-In	In/Out	Control	Yes
18-25	19-30	Ground	Grnd		

(Peacock, 1998)

## 2.2 Sensor Optocoupler

*Optocoupler* merupakan piranti elektronika yang berfungsi sebagai pemisah antara rangkaian *power* dengan rangkaian *control*. *Optocoupler* merupakan salah satu jenis komponen yang memanfaatkan sinar sebagai pemicu ON/OFF-nya. Opto berarti *optic* dan *coupler* berarti pemicu. Sehingga bisa diartikan bahwa *optocoupler* merupakan suatu komponen yang bekerja berdasarkan picu cahaya *optic*. *Optocoupler* termasuk dalam sensor, dimana terdiri dari dua bagian yaitu *transmitter* dan *receiver*.



Gambar 2: Sensor Optocoupler (Distribution, 2001)

## 2.3 Bahasa Pemrograman Java

Java adalah sebuah platform

teknologi pemrograman yang dikembangkan oleh Sun Microsystem. Java diciptakan oleh suatu tim yang dipimpin oleh Patrick Naughton dan James Gosling dalam suatu proyek dari Sun Microsystem. Mereka menyebut bahasa pemrograman yang dihasilkan dengan nama Oak lalu diganti dengan nama Java.

Java adalah bahasa pemrograman berorientasi objek yang serba guna. Bahasa ini menyediakan sejumlah perluasan yang mendukung pengembangan aplikasi GUI, dan juga pengembangan aplikasi klien server terhadap jaringan lokal (LAN) atau jaringan luas (WAN). Java meminjam model ekstensibilitas sewaktu pengeksekusian, manajemen memori dinamis, dan beberapa pengeksekusian secara bersamaan (*multi-threading*) (Kadir, 1996).

Java dibuat untuk membuat bahasa pemrograman yang *portable* dan *independent* terhadap lingkungan (*platform independent*). Java dapat membuat perangkat lunak yang dapat ditanamkan (*embedded*) pada berbagai mesin dan peralatan seperti *handphone*, *remote control*, *microwave* dan lain-lain.

Sintaks Java merupakan pengembangan dari bahasa C/C++. Berikut adalah beberapa karakteristik pemrograman Java:

- *Object Oriented*. Pemrograman berorientasi objek memodelkan dunia nyata kedalam objek dan melakukan interaksi antar objek-objek tersebut.
- *Portable*. Java tidak terdapat aspek yang tergantung pada lingkungan implementasi. *Source code* maupun program Java dapat dengan mudah dijalankan pada *platform* yang berbeda-beda tanpa harus dikompilasi ulang.
- *Architecture Neutral*. Program Java merupakan *platform independent*. Program cukup mempunyai satu buah versi yang dapat dijalankan pada

*platform* berbeda dengan *Java Virtual Machine*.

- *Simple*. Java mudah untuk dipelajari terutama untuk orang yang sudah mengenal pemrograman, tapi belum terlalu terikat pada paradigma pemrograman prosedural.
- *Distributed*. Java dibuat untuk membuat aplikasi terdistribusi secara mudah dengan adanya *library networking* yang terintegrasi pada Java. Java memiliki *library* yang lengkap untuk dirangkai pada protocol TCP/IP, seperti HTTP dan FTP.
- *Secure*. Sebagai bahasa pemrograman untuk aplikasi internet dan terdistribusi, Java memiliki beberapa mekanisme keamanan untuk menjaga aplikasi tidak digunakan untuk merusak sistem komputer yang menjalankan aplikasi tersebut.
- *Multi-threaded*. Java mempunyai kemampuan untuk membuat suatu program yang dapat melakukan beberapa pekerjaan secara sekaligus dan simultan.
- *Dynamic*. Dalam beberapa hal, Java merupakan bahasa yang lebih dinamis dari C atau C ++. Dirancang untuk beradaptasi dengan lingkungan yang berkembang.
- *Robust*. Java mempunyai reliabilitas yang tinggi. *Compiler* pada Java mempunyai kemampuan mendeteksi *error* secara lebih teliti dibandingkan bahasa pemrograman lain. Java mempunyai *runtime exception handling* untuk membantu mengatasi *error* pada pemrograman.

## 2.4 Database MySQL

MySQL adalah *Relational Database Management System (RDMS) open source* yang saat ini cukup banyak digunakan pada berbagai aplikasi. Keandalannya dalam mengolah *database* ditunjang kecepatannya dalam mengakses perintah *query* serta banyaknya fitur-fitur

yang dimiliki menjadikannya sebagai database yang cukup banyak digunakan saat ini.

Untuk dapat mengakses data dalam database relational, digunakan sebuah bahasa standar yang disebut SQL (*Structure Query Language*). Hampir semua *database server* yang ada saat ini mendukung bahasa SQL untuk melakukan manajemen datanya.

Secara umum, SQL terdiri dari dua bahasa, yaitu *Data Definition Language (DDL)* dan *Data Manipulation Language (DML)*. Implementasi DDL dan DML berbeda untuk tiap sistem manajemen *database*, namun secara umum implementasi tiap bahasa ini memiliki bentuk standar yang ditetapkan ANSI (*American National Standards Institute*).

## 2.5 Konsep Dasar SMS

### 2.5.1 Pengertian SMS

SMS merupakan salah satu fitur GSM yang dikembangkan dan distandarisi oleh *ETSI (European Telecommunication Standard Institute)* yaitu badan standar telekomunikasi eropa yang menstandarisasikan fitur-fitur GSM (ESTI, 1996). SMS atau *Short Message Service* merupakan sebuah layanan yang banyak diaplikasikan pada sistem komunikasi tanpa kabel, memungkinkan dilakukannya pengiriman pesan antara *terminal* pelanggan dengan sistem eksternal seperti *email*, *voice mail* dan lain-lain. Pelayanan SMS menggunakan *SMS Center (SMSC)* yang bertindak sebagai sistem simpan dan diteruskan bagi pesan pendek tersebut.

Pada proses pengiriman SMS dari *handphone*, SMS tersebut tidak langsung dikirim ke *handphone* tujuan, tetapi terlebih dahulu dikirim ke *SMS Center*, kemudian dengan sistem *store and forward* SMS tersebut dikirimkan ke *handphone* tujuan. Jadi dengan keberadaan SMSC kita dapat mengetahui

status SMS, apakah sudah terkirim atau gagal terkirim. Jika *handphone* tujuan dalam keadaan aktif dan berada dalam jangkauan, maka *handphone* tersebut mengirim konfirmasi ke SMSC bahwa SMS telah diterima. Sebaliknya jika *handphone* tujuan mati atau di luar jangkauan, maka SMS akan disimpan di SMSC sampai periode validitas terpenuhi. Jika sampai periode validitas terpenuhi, SMSC akan mengirimkan pesan ke *handphone* pengirim bahwa SMS gagal dikirimkan.



Gambar 3: Cara Kerja SMS

### 2.5.2 Protocol Data Unit (PDU)

Dalam pengiriman atau penerimaan SMS ada dua mode yakni *mode PDU (Protocol Data Unit)* dan *mode teks*. *Mode PDU* merupakan format pesan dalam bentuk *oktet heksadesimal* dan *oktet semidesimal* dengan panjang mencapai 140 karakter (7 bit) atau 160 karakter (8 bit). *Mode teks* adalah format pesan dalam bentuk teks asli yang dituliskan pada saat akan mengirim pesan. Mode teks ini adalah hasil pengkodean dari mode teks PDU. Di Indonesia, tidak semua operator GSM mendukung *mode teks*, sehingga *mode* yang digunakan adalah PDU. Pada pengiriman SMS ada 2 (dua) sisi, yaitu *mobile terminated (handphone penerima)* dan *mobile originated (handphone pengirim)*.

Pada proses pengiriman, SMS yang dikirim oleh *terminal* masih dalam bentuk teks, sedangkan pada pengiriman ke SMSC harus dalam bentuk PDU. Oleh karena itu, *terminal handphone* mengubah format teks ke PDU (*encodec*). Sedangkan pada proses penerimaan SMS, format PDU yang

diterima oleh *handphone* harus diubah menjadi teks agar dapat dibaca kembali (*decodec*).

### **2.5.3 AT Command**

Dibalik teks SMS yang diterima dan dikirim pada sebuah telepon seluler sebenarnya adalah berupa perintah *AT Command* yang bertugas mengirim atau menerima data dari dan ke *SMS Center*. Perintah *AT Command* tiap-tiap perangkat dapat berbeda-beda, setiap *vendor* biasanya memberikan referensi tentang daftar perintah AT yang tersedia. *AT Command* digunakan untuk berkomunikasi dengan terminal melalui *port serial* pada komputer. Dengan menggunakan perintah AT, dapat diketahui kekuatan sinyal dari *terminal*, mengirim pesan, menambahkan *item* pada buku alamat, mematikan *terminal* dan banyak fungsi lainnya.

## **3. ANALISIS DAN PERANCANGAN APLIKASI**

Berdasarkan permasalahan yang dihadapi oleh Stasiun Kereta Api Kebayoran, selanjutnya dianalisis strategi pemecahan masalah yang paling tepat beserta rancangan aplikasinya.

### **3.1 Strategi Pemecahan Masalah**

Untuk mengatasi masalah tersebut, maka diperlukan suatu aplikasi otomatis yang menggunakan komputer. Aplikasi ini berfungsi untuk kendali pintu perlintasan dan pemberitahuan kedatangan kepada penumpang kereta api yang dilakukan secara otomatis. Dengan demikian, aplikasi ini akan sangat membantu petugas palang pintu dan penumpang kereta api pada Stasiun Kereta Api Kebayoran dalam melaksanakan tugasnya.

Jadwal keberangkatan dan kedatangan kereta api sudah ditetapkan, namun dalam kenyataannya masih terjadi keterlambatan baik karena kendala teknis

maupun non teknis. Keterlambatan akan mengakibatkan penumpukan penumpang kereta api di stasiun. Melalui aplikasi ini, waktu kedatangan dan keberangkatan kereta api akan diberitahukan melalui pengeras suara, sehingga calon penumpang kereta api dapat bersiap-siap. Pemberitahuan akan aktif saat sensor *optocoupler* yang dipasang di bantalan kereta api aktif. Sensor dapat dipasang dengan jarak tertentu untuk mengatur waktu pemberitahuan.

Saat kereta api berangkat dari stasiun Kebayoran, petugas harus memberikan informasi ke stasiun kereta api selanjutnya menggunakan telepon. Hal tersebut dapat mengakibatkan permasalahan jika petugas kurang disiplin atau lalai dalam menjalankan tugasnya. Aplikasi ini dirancang untuk mengirimkan pemberitahuan secara otomatis melalui SMS ke stasiun kereta api selanjutnya ketika kereta api berangkat dari Stasiun Kereta Api Kebayoran. SMS akan terkirim secara otomatis saat kereta api akan berangkat.

Petugas Stasiun Kereta Api Kebayoran dan petugas palang pintu harus siaga apabila kereta api akan melewati pintu perlintasan pada Stasiun Kereta Api Kebayoran. Sehingga apabila petugas PPKA dan petugas palang pintu melakukan kelalaian dapat mengakibatkan kecelakaan kereta api dengan kendaraan umum di pintu perlintasan kereta api. Oleh sebab itu, aplikasi ini juga dirancang untuk menutup pintu perlintasan kereta api dan menyalakan *sirene* yang dilakukan secara otomatis saat kereta api akan melintas. Dengan adanya *sirene*, pengendara kendaraan umum yang akan melintas juga dapat menghentikan kendaraanya ketika *sirene* berbunyi.

Dalam pembuatan aplikasi ini dibutuhkan beberapa komponen, yaitu :

- Sebuah *interface* yang sudah dirangkai untuk aplikasi kendali pintu perlintasan dan pemberitahuan

kedatangan kereta api otomatis.

- Suatu program aplikasi yang akan melakukan kendali terhadap *interface* melalui komputer.
- Kabel data DB-25 yang dibutuhkan untuk menghubungkan antara *interface* dengan komputer melalui *parallel port*.
- Sebuah *handphone* yang berfungsi sebagai *SMS Gateway Server* dan terhubung ke komputer.

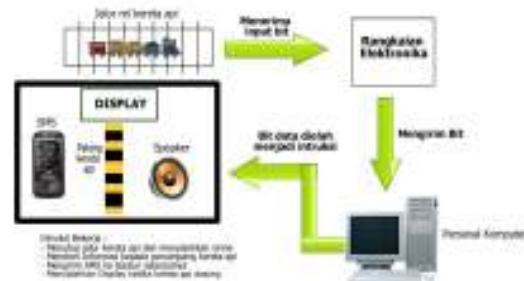
### 3.2 Rancangan Aplikasi

Pada aplikasi ini terdapat *login* sebagai salah satu pengamanan aplikasi, sehingga pengguna aplikasi dan terkontrol. *User* dibagi menjadi 2 (dua) *level* yaitu *Administrator* dan *General User*. *Administrator* dapat mengakses semua menu, sedangkan *General User* hanya dapat mengakses menu tertentu.

Aplikasi pada komputer terhubung dengan *interface* melalui *parallel port* dengan menggunakan kabel data DB-25. *Parallel port* untuk mengirimkan data yang ditransmisikan memiliki kecepatan yang tinggi. *Interface* ini menggunakan sensor *optocoupler* dimana sensor ini akan aktif apabila cahaya inframerah terputus. Agar cahaya inframerah terputus dalam sensor ini dibutuhkan tekanan sehingga sensor ini bekerja. Sensor *optocoupler* dipasang pada bantalan rel kereta api dan menghindari dari jangkauan manusia, sehingga hanya kereta api yang bisa menekan sensor *optocoupler*.

Sensor *optocoupler* dipasang pada bantalan rel kereta api arah Palmerah dan arah Pondok Ranji sebelum pintu perlintasan. Apabila kereta api mengenai sensor *optocoupler* tersebut, maka menutup pintu perlintasan kereta api, menyalakan *sirene* dan memberitahukan kedatangan kereta api kepada penumpang melalui pengeras suara. Sensor *optocoupler* juga dipasang pada bantalan rel di Stasiun Kereta Api Kebayoran agar ketika kereta api tiba di Stasiun Kereta

Api Kebayoran akan menampilkan *display* dan jika kereta api berangkat dari Stasiun Kereta Api Kebayoran akan mengirimkan SMS ke stasiun kereta api selanjutnya.



Gambar 4: Rancangan Aplikasi

Dalam prosesnya, bit-bit yang dikirim oleh *interface* diolah pada aplikasi menjadi instruksi-instruksi. Instruksi bekerja untuk menutup palang pintu, menyalakan *sirene*, memberikan informasi waktu ketika kereta api akan tiba, menyalakan *display* dan mengirimkan SMS ke stasiun kereta api selanjutnya.

Secara singkat, proses pengolahan bit-bit tersebut mengikuti aturan sebagai berikut:

- 1) Jika kereta api mengenai sensor dari arah Palmerah pada jalur 1, maka sensor *optocoupler* akan mengirim bit 1. Setelah itu, data diolah komputer untuk pintu perlintasan kereta api tertutup dari arah Palmerah, pintu perlintasan kereta api dari arah Pondok Ranji terbuka, dan *display* mati. Kondisi tersebut akan mengirim bit data keluar 65.
- 2) Jika kereta api mengenai sensor dari arah Palmerah pada jalur 1 dan berhenti di Stasiun Kebayoran, sensor tersebut akan mengirim bit 69. Setelah itu data diolah oleh komputer untuk pintu perlintasan kereta api terbuka dari arah Palmerah dan Pondok Ranji, dan *display* menyala. Kondisi tersebut akan mengirim bit data keluar 4.
- 3) Jika kereta api berangkat dari Stasiun Kebayoran pada jalur 1, akan

mengirim bit 148. Setelah itu data diolah oleh komputer untuk lampu hijau menyala, pintu perlintasan Kereta Api tertutup dari arah Pondok Ranji, pintu perlintasan Kereta Api terbuka dari arah Palmerah dan *display* mati. Kondisi tersebut akan mengirim bit data keluar 0.

- 4) Jika kereta api mengenai sensor dari arah Pondok Ranji pada jalur 2, maka sensor *optocoupler* akan mengirim bit. Sensor tersebut akan mengirim bit 2. Setelah itu, data diolah komputer untuk pintu perlintasan kereta api tertutup dari arah Pondok Ranji, pintu perlintasan kereta api dari arah Palmerah terbuka dan *display* mati. Kondisi tersebut akan mengirim bit data keluar 130.
- 5) Jika kereta api mengenai sensor dari arah Pondok Ranji pada jalur 2 dan berhenti di Stasiun Kebayoran, sensor tersebut akan mengirim bit 134. Setelah itu data diolah oleh komputer untuk pintu perlintasan kereta api terbuka dari arah Palmerah dan Pondok Ranji, dan *display* menyala. Kondisi tersebut akan mengirim bit data keluar 4.
- 6) Jika kereta api berangkat dari Stasiun Kebayoran pada jalur 2, akan mengirim bit 148. Setelah itu data diolah oleh komputer untuk lampu hijau menyala, pintu perlintasan kereta api tertutup dari arah Pondok Ranji, pintu perlintasan kereta api terbuka dari arah Palmerah dan *display* mati. Kondisi tersebut akan mengirim bit data keluar 0.

### 3.3 Spesifikasi Basis Data

Basis data diperlukan dalam aplikasi ini untuk menyimpan pengguna aplikasi dan untuk mencatat pesan-pesan yang dikirimkan melalui *SMS Gateway*. Pesan-pesan tersebut cukup penting untuk disimpan agar memudahkan proses identifikasi jika terjadi kesalahan informasi atau kegagalan sistem.

Berikut ini spesifikasi basis data yang digunakan dalam aplikasi ini:

Tabel 2: Tabel *user*

No	Nama Field	Tipe Data	Panjang	Keterangan
1	Username	Varchar	25	Usernama pengguna
2	NamaLengkap	Varchar	30	Nama lengkap
3	Password	Varchar	25	Password pengguna
4	Level	Varchar	15	Level pengguna

Tabel 3: Tabel pesan

No	Nama Field	Tipe Data	Panjang	Keterangan
1	NoPesan	Int	5	Nomor urut pesan
2	NoHP	Varchar	15	Nomor handphone
3	IsiPesan	Varchar	160	IsiPesan
4	WktTglPesan	Timestamp	19	Waktu pengiriman pesan
5	JenisPesan	Varchar	11	JenisPesan

### 3.4 Rancangan Perangkat Keras

Aplikasi ini memerlukan beberapa perangkat keras dan rangkaian elektronika. Berikut ini penjelasan terkait rancangan perangkat keras tersebut:

#### 3.4.1 *Parallel Port*

Untuk dapat menggunakan *parallel port*, harus terlebih dahulu diketahui alamatnya. BIOS yang akan mengatur pengalamatannya. Alamat basis (*base address*) untuk LPT1 adalah 888 (378h) dan LPT2 adalah 632 (278h).

*Parallel port* pada perancangan perangkat keras digunakan sebagai pengirim sinyal data dan penerima data ke rangkaian dari rangkaian antarmuka. Pada *parallel port* terdapat 25 pin pada konektor DB25, 17 pin berfungsi sebagai saluran pembawa informasi. Saluran pembawa tersebut terbagi menjadi 3 bagian, yaitu: 8 bit data, 5 bit status dan 4 bit kontrol. Sedangkan 8 pin berfungsi sebagai *ground*.

Pin yang digunakan pada aplikasi kendali pintu perlintasan dan pemberitahuan kedatangan kereta api otomatis menggunakan 8 pin, 4 pin berfungsi sebagai input yaitu 1 pin untuk sensor *optocoupler* dari palmerah, 1 pin untuk sensor *optocoupler* dari Pondok Ranji, 1 pin untuk sensor *optocoupler* untuk Kereta Api Listrik (KRL) dan 1 pin



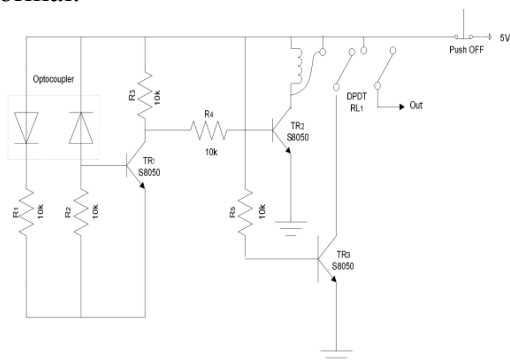
untuk sensor *optocoupler* berhenti di Stasiun Kereta Api Kebayoran dan *display*.

Sedangkan 4 pin berfungsi sebagai *output* yaitu 2 pin untuk sensor *optocoupler* untuk kereta api berhenti pada jalur 1 dan jalur 2 di Stasiun Kereta Api Kebayoran, 2 pin untuk menyalakan sirene dan pintu perlintasan arah palmerah dan pintu perlintasan arah pondok ranji.

### 3.4.2 Rangkaian Sensor Optocoupler

Rangkaian sensor (*optocoupler*) merupakan inti utama dalam aplikasi ini. Pada rangkaian ini terdiri dari 5 buah resistor dengan hambatan 10K ohm, 3 buah transistor dengan tipe s8050, 1 buah relay DPDT, 1 buah Switch Push Off dan 1 buah sensor *optocoupler*.

Rangkaian ini bekerja apabila kereta api melintasi rel lalu transistor (TR2) mengaktifkan *relay* (RL1). Setelah *relay* (RL1) aktif, *relay* (RL1) akan bekerja untuk memberi tegangan supaya mengaktifkan transistor (TR3). Transistor (TR3) akan bekerja untuk mengaktifkan *relay* tetap hidup. *Switch Push Off* untuk me-reset *relay* supaya kondisi menjadi normal.



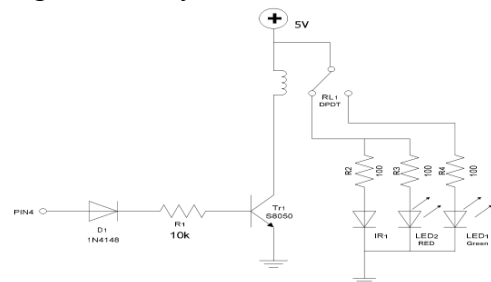
Gambar 5: Rangkaian *optocoupler*

### 3.4.3 Rangkaian Pemberhentian Kereta Api

Rangkaian pemberhentian kereta api berfungsi untuk ketika kereta api berhenti di Stasiun Kereta Api Kebayoran. Rangkaian ini terdiri dari 2 buah resistor

yang membutuhkan hambatan 10K ohm, 1 buah dioda dengan tipe IN4148, 1 buah transistor dengan tipe s8050, 2 buah LED dan satu infrared.

Rangkaian ini bekerja mendapat tegangan dari pin 4. Inputan dari pin 4 akan mengaktifkan transistor (Tr1), transistor (Tr1) bekerja untuk mengaktifkan *infrared*.

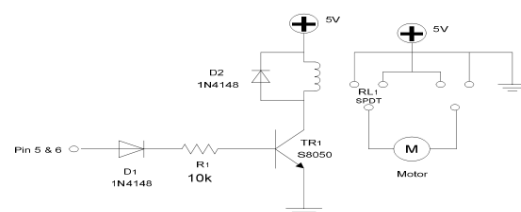


Gambar 6: Rangkaian Pemberhentian KA

### 3.4.4 Rangkaian Pintu Perlintasan

Rangkaian pintu perlintasan berfungsi untuk menutup palang pintu ketika kereta api akan datang pada Stasiun Kereta Api Kebayoran. Rangkaian ini terdiri dari 2 buah resistor dengan hambatan 10K ohm, 1 buah relay SPDT, 1 buah dioda dengan tipe IN4148, 1 buah transistor dengan tipe s8050 dan 1 buah motor DC sebagai mengerakkan palang pintu.

Rangkaian ini bekerja mendapat tegangan dari pin 5 dan pin 6. Inputan dari pin 5 dan pin 6 akan mengaktifkan transistor. *Transistor* mendapat tegangan untuk menghidupkan *relay*. *Relay* bekerja untuk menggerakkan motor kearah tertentu atau menutup palang pintu perlintasan kereta api.



Gambar 7: Rangkaian Pintu Perlintasan

### 3.4.5 Rangkaian *Display*

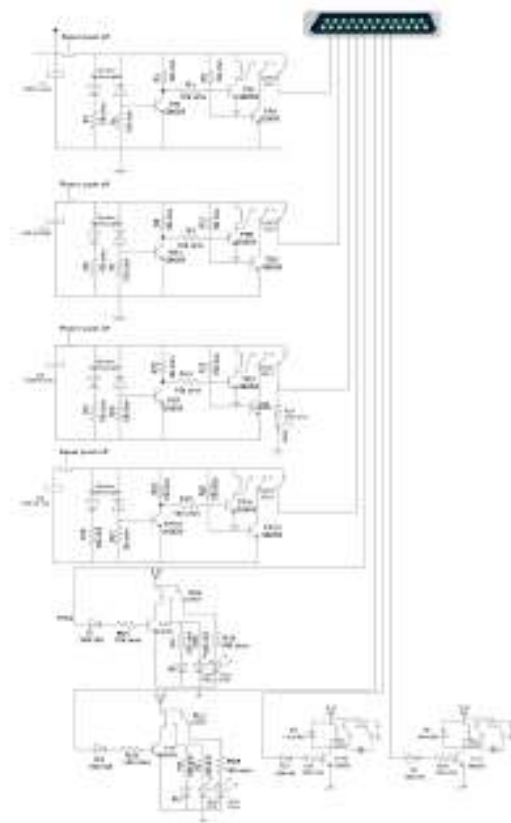
Rangkaian *display* berfungsi untuk menyalakan *display* ketika kereta api datang pada Stasiun Kereta Api Kebayoran. Pada rangkaian ini terdiri dari 5 buah resistor hambatan 10K ohm dan 1 resistor hambatan 100 ohm, 3 buah transistor dengan tipe s8050, 1 buah relay DPDT, 1 buah Switch Push Off, 1 buah LED dan 1 buah sensor *optocoupler*.

Rangkaian ini bekerja apabila kereta api sudah tiba di Stasiun Kereta Api Kebayoran lalu transistor (TR2) mengaktifkan relay (RL1). Setelah relay (RL1) aktif, relay (RL1) akan bekerja untuk memberi tegangan supaya mengaktifkan transistor (TR3). Transistor (TR3) akan bekerja untuk mengaktifkan relay tetap hidup dan mengaktifkan LED supaya menyala. *Switch Push Off* untuk mereset relay supaya kondisi menjadi normal.

### 3.4.6 Rangkaian Gabungan

Rangkaian keseluruhan antarmuka mencakup semua komponen elektronika pada aplikasi ini. Rangkaian ini terdiri dari beberapa komponen, diantaranya adalah 4 sensor *optocoupler*, 25 resistor dengan nilai 10k ohm, 6 resistor dengan nilai 100 ohm, 5 dioda dengan tipe 1N4148, 5 LED, 16 transistor dengan tipe s8050, 4 kapasitor dengan tipe 100/16 volt, 6 relay DPDT dan 2 relay SPDT.

Berikut ini rangkaian keseluruhan antarmuka:



Gambar 8: Rancangan Antarmuka Gabungan

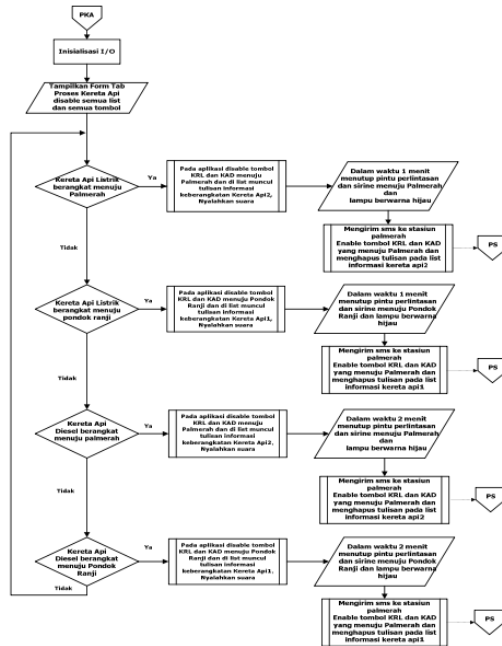
### 3.5 *Flowchart* Aplikasi

Untuk menggambarkan proses yang dilakukan oleh program dalam aplikasi ini, berikut ini beberapa *flowchart* utama dari aplikasi ini.

#### 3.5.1 *Flowchart* Proses Keberangkatan Kereta Api

Berikut ini adalah *flowchart* kereta api keberangkatan pada proses kereta api. Sebelum kereta api berangkat harus memilih panel kereta api listrik atau panel kereta api diesel. Setelah memilih panel kereta api tersebut lalu memilih keberangkatan menuju palmerah atau pondok ranji. Keberangkatan kereta api meliputi pemberitahuan keberangkatan lewat speaker, menutup pintu perlintasan, dan mengirim informasi keberangkatan ke stasiun kereta api selanjutnya melalui SMS. Urutan proses yang akan dilalui pada kereta api keberangkatan

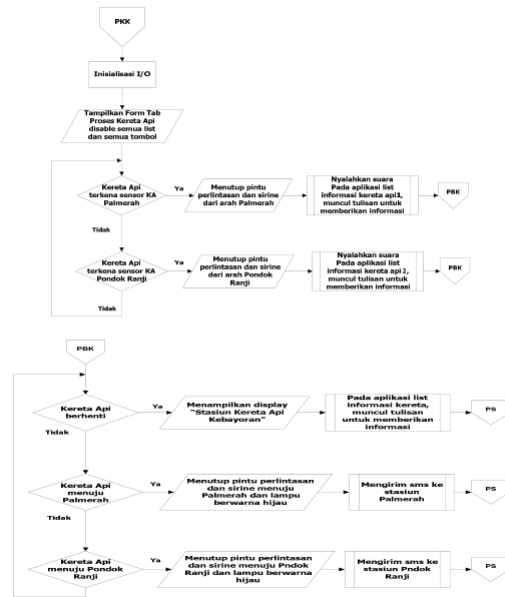
digambarkan dengan *flowchart* pada gambar berikut ini:



Gambar 9: *Flowchart* Keberangkatan Kereta Api

### 3.5.2 *Flowchart* Proses Kedatangan dan Kereta Api Berhenti

Berikut ini adalah *flowchart* kereta api kedatangan dan kereta api berhenti pada proses kereta api. Kereta api yang datang ke Stasiun Kereta Api Kebayoran akan menutup pintu perlintasan, menyalakan sirene, menyalakan suara dan pemberitahuan informasi pada list informasi kereta api. Kereta api yang berhenti pada Stasiun Kereta Api Kebayoran akan menyalakan *display* sedangkan kereta api yang tidak berhenti langsung menuju arah tujuannya. Urutan proses yang akan dilalui pada kereta api kedatangan dan kereta api berhenti digambarkan dengan *flowchart* pada gambar berikut ini:



Gambar 10: *Flowchart* Proses Kedatangan KA

## 4. IMPLEMENTASI DAN UJI COBA APLIKASI

Program kendali pintu perlintasan dan pemberitahuan kedatangan kereta api otomatis pada Stasiun Kereta Api Kebayoran menggunakan sensor *optocoupler* dibuat dengan tujuan memudahkan pegawai kereta api dalam melakukan pekerjaannya untuk menutup pintu perlintasan saat kereta api melewati jalur perlintasan pada Stasiun Kereta Api Kebayoran secara otomatis dan memberikan informasi kepada penumpang bahwa kereta api akan datang ke Stasiun Kebayoran. Dengan adanya aplikasi ini, menghindari kecelakaan di pintu perlintasan dan pemberitahuan informasi yang dilakukan secara otomatis.

### 4.1 Spesifikasi Perangkat Keras dan Perangkat Lunak

Program pemberitahuan kedatangan keberangkatan kepada penumpang kereta api dan pintu perlintasan otomatis ini akan berjalan jika perangkat keras dan perangkat lunak yang dibutuhkan

memenuhi kriteria untuk dapat menjalankan program. Berikut kriteria perangkat keras dan perangkat lunak yang dibutuhkan.

#### **4.1.1 Perangkat Keras**

Untuk dapat menjalankan program serta menghubungkannya ke alat diperlukan perangkat keras yang mendukung. Di bawah ini merupakan spesifikasi perangkat keras yang dianjurkan untuk menjalankan alat:

- PC Prosesor Intel Pentium III
- Hard Disk Seagate 40 Gb
- RAM SD-RAM Visipro 512 MB
- Monitor
- Kabel data parallel (LPT1 / DB-25 )
- Motherboard ABIT KT7A
- Speaker
- Keyboard
- Mouse
- Handphone Sony Ericsson w850i
- Kabel Data handphone Sony Ericsson w850i atau Bluetooth Dongle Billionton.

#### **4.1.2 Perangkat Lunak**

Dibutuhkan perangkat lunak untuk menghubungkan program dengan *interface*. Di bawah ini merupakan spesifikasi perangkat lunak untuk menjalankan aplikasi kendali pintu perlintasan dan pemberitahuan kedatangan kereta api otomatis :

- Sistem Operasi Windows XP SP 2
- J2SDK dan JVM versi 1.6+
- Jnpout32, merupakan *library* untuk menghubungkan *parallel port* dengan Java.
- *Commapi*, merupakan *library* untuk untuk berkomunikasi port dengan java pada *SMS Server*.
- Java Media Framework (JMF 2.1.1e)
- Database Mysql 5+.
- Program aplikasi kendali pintu perlintasan dan pemberitahuan kedatangan kereta api otomatis atau aplikasi kereta api 1.0.

#### **4.2 Tampilan Aplikasi**

Berikut ini adalah penjelasan dan gambar mengenai tampilan-tampilan yang ada pada program Aplikasi Kendali Pintu Perlintasan dan Pemberitahuan Kedatangan Kereta Api Otomatis, beserta cara pengoperasian programnya.

##### **4.2.1 Tampilan Utama**

Pada *form* menu utama berisi untuk menampung semua form pada aplikasi ini. Pada menu utama terdapat dua menu utama yaitu menu utama untuk *Administrator* dan *General User*. Sebelum menampilkan form menu utama, terlebih dahulu menampilkan proses loading. Tampilan layar form menu utama seperti pada gambar 11 berikut ini.



Gambar 11: Tampilan Utama

##### **4.2.2 Tampilan Login Pengguna**

Pada form login ini berfungsi untuk membatasi penggunaan aplikasi, sehingga hanya *user* yang terdaftar di database yang dapat menggunakan aplikasi. Sebelum *user* menggunakan aplikasi ini *user* harus menyetik *username* dan *password* terlebih dahulu pada form login. Didalam form login terdapat dua tombol yaitu tombol *Login* dan tombol *Exit* . Tombol login berfungsi untuk mengecek *user name*, *password*, serta hak akses. Apabila *user name* dan *password* benar, selanjutnya mengecek hak akses. Apabila hak akses *Administrator* maka ke menu utama *Administrator* atau apabila hak akses

*General User* maka ke menu utama *General User*. Sedangkan tombol exit berfungsi jika *user* ingin keluar dari form login. Berikut ini tampilan layar form login pada gambar 12.



Gambar 12: Tampilan Form Login

#### 4.2.3 Tampilan Proses Kereta Api

Pada tab proses kereta api ini berfungsi untuk melihat informasi kedatangan dan keberangkatan kereta api, dan memberangkatkan kereta api yang tiba di Stasiun Kereta Api Kebayoran. Ketika kereta api tiba atau berangkat akan memberi informasi pada *list* informasi kereta api 1 dan *list* informasi kereta api 2.

Apabila kereta api listrik berangkat, akan memilih tombol menuju Pondok Ranji dan Palmerah lalu memberitahukan bahwa kereta api akan berangkat sekitar 1 menit lagi kepada penumpang. Sedangkan jika kereta api diesel yang berangkat, maka dapat ditekan tombol menuju Pondok Ranji dan Palmerah lalu memberitahu kepada penumpang, kereta api akan berangkat sekitar 2 menit.

Berikut ini tampilan aplikasi proses kereta api pada gambar 13.



Gambar 13: Tampilan Proses Kereta Api

#### 4.2.3 Tampilan SMS Server

Pada tampilan SMS Server berfungsi untuk mengkonfigurasi komputer supaya bisa berhubungan dengan *handphone*, melihat informasi tentang *handphone*, dan melihat pesan masuk dan pesan keluar. Ada empat tombol pada tab pengaturan dan pesan yaitu tombol konfigurasi *port*, sambung, putus, dan batal. Tombol konfigurasi *port* berfungsi untuk mengatur *port*, *baud rate*, *data bits*, *parity*, *stop bits*, dan *flow control*.

Dengan pengaturan ini komputer dapat berkomunikasi dengan *handphone*. Tombol sambung berfungsi untuk proses pengambilan *input* dan *output stream data*, pengaturan terhadap *serial port* yang telah berhasil terhubung dan melakukan pengaturan beberapa komponen setelah semua proses berhasil terlaksana. Tombol putus berfungsi pemutusan terhadap hubungan serial port dengan terminal. Tombol batal berfungsi untuk membersihkan data pada pesan SMS masuk dan keluar.

Gambar 14 berikut ini tampilan layar tab pengaturan dan pesan.



## 5. KESIMPULAN

Berdasarkan pembahasan di atas, dapat diambil beberapa kesimpulan dan saran untuk pengembangan aplikasi ini.

### 5.1 Kesimpulan

Beberapa kesimpulan dari penelitian ini adalah:

- Bagi petugas kereta api, aplikasi ini sangat membantu dalam mengontrol kedatangan dan keberangkatan kereta api di Stasiun Kebayoran.
- Bagi penumpang kereta api, aplikasi ini akan memberikan informasi kedatangan dan keberangkatan kereta api, sehingga penumpang akan merasa lebih nyaman.
- Bagi pengguna kendaraan yang melewati pintu perlintasan kereta api, otomatisasi palang pintu kereta api dan sirene akan membantu mengurangi resiko terjadinya kecelakaan.
- Dengan menggunakan sensor *optocoupler* yang peka terhadap cahaya inframerah, aplikasi ini menjadi lebih aman dan terjamin.
- Penerapan aplikasi ini dapat meringankan biaya dan tidak membuang-buang waktu dalam melakukan pengoperasiannya.

### 5.1 Saran

Berikut ini beberapa saran pengembangan aplikasi:

- Pada pengembangan berikutnya aplikasi ini dapat digunakan apabila jalur rel kereta api terjadi kerusakan.
- Tanpa perawatan dan pengawasan dari pihak yang bertanggung jawab dalam pemeliharaan aplikasi, maka aplikasi tidak akan berjalan dengan baik dan waktu yang panjang.
- Pada pengembangan berikutnya untuk pemberitahuan kedatangan kereta api adanya tujuan kereta api yang tiba di Stasiun Kebayoran.
- Pada pengembangan berikutnya

supaya bisa memberitahu informasi apabila terjadi kerusakan pada sensor *optocoupler*.

- Dengan pemberitahuan informasi menggunakan SMS, diperlukan kerjasama antara Stasiun Kereta Api Kebayoran dengan provider SMS, supaya tidak terdapat permasalahan dalam pengiriman SMS.

## DAFTAR PUSTAKA

1. Bies, L. (2010). *Parallel cables pinout and port info*. Retrieved May 10, 2011, from Lammert Bies - Computer Interfacing: <http://www.lammertbies.nl/comm/cable/parallel.html>
2. Distribution, E. (2001). *Optocouplers: When & How To Use Them*.
3. ESTI. (1996, July). *GSM Technical Specification*. France.
4. Kadir, A. (1996). *Java By Example : Edisi Indonesia*. Yogyakarta: Andi Offset.
5. Peacock, C. (1998). *Interfacing the Standard Parallel Port*. Retrieved from Beyond Logic: <http://www.senet.com.au/~cpeacock>
6. Ditjen Perkeretaapian. (2010). *Statistik Penumpang dan Barang*. Retrieved May 5, 2011, from Direktorat Jenderal Perkeretaapian: [http://perkeretaapian.dephub.go.id/index.php?option=com\\_content&view=article&id=63&Itemid=63&ffe5d588932e0dd5fc957eca7f6225ad=67884ea1315f8a282bf3d0e9c19cfad1](http://perkeretaapian.dephub.go.id/index.php?option=com_content&view=article&id=63&Itemid=63&ffe5d588932e0dd5fc957eca7f6225ad=67884ea1315f8a282bf3d0e9c19cfad1)
7. Widodo, R. B. (2007). *Interfacing Paralel & Serial Menggunakan Delphi*. Yogyakarta: Graha Ilmu.
8. Wikipedia. (n.d.). *IEEE 1284*. Retrieved May 10, 2011, from Wikipedia: [http://en.wikipedia.org/wiki/IEEE\\_1284](http://en.wikipedia.org/wiki/IEEE_1284)