

Rancang Bangun Aplikasi Teknologi RFID dan SMS Gateway (Studi Kasus: SMA Negeri 2 Karawang)

Ganjar Ramadhan, Nenny Anggraeni dan Khodijah Hullyyah
Jurusan Teknik Informatika Fakultas Sains dan Teknologi
UIN Syarif Hidayatullah Jakarta
ganjar.ramadhan05@gmail.com, nenny_a@yahoo.com, khodijah2@yahoo.com

Abstrak—Pendidikan merupakan salah satu faktor pendorong utama dalam memajukan suatu bangsa. Ada banyak faktor yang mempengaruhi kualitas pendidikan, salah satunya adalah kedisiplinan. Kedisiplinan pelajar di Indonesia dapat dikatakan masih kurang. Data yang penulis kumpulkan menunjukkan bahwa 59,6% siswa mengaku pernah membolos dari sekolah. Hal ini mungkin terjadi akibat kurang baiknya komunikasi antara pihak sekolah dan orang tua. Dengan perkembangan teknologi sekarang ini, permasalahan di atas bisa diatasi dengan menggunakan teknologi SMS. Pihak sekolah bisa mengirimkan informasi absensi kedatangan siswa di sekolah. Solusi tersebut yang kemudian dijadikan tujuan dari penelitian yang penulis lakukan. Data yang digunakan pada penelitian ini diperoleh dari buku-buku literatur, internet dan melakukan wawancara dengan pihak sekolah SMA Negeri 2 Karawang. Data tersebut akan digunakan sebagai dasar proses rancang bangun aplikasi absensi siswa menggunakan teknologi RFID dan SMS gateway yang merupakan hasil akhir dari penelitian yang penulis lakukan. Aplikasi tersebut akan penulis kembangkan menggunakan bahasa pemrograman Java menggunakan teknologi Java RMI (*Client-Server*) dan database MySQL.

Kata Kunci—absensi siswa, RFID, SMS gateway, Java, client-server, Java RMI, MySQL

I. PENDAHULUAN

Pendidikan merupakan salah satu faktor pendorong utama dalam memajukan suatu bangsa. Ada banyak faktor yang mempengaruhi kualitas pendidikan, salah satunya adalah kedisiplinan. Kedisiplinan pelajar di Indonesia dapat dikatakan masih kurang. Data yang penulis kumpulkan menunjukkan bahwa 59,6% siswa mengaku pernah membolos dari sekolah. Setidaknya ada tiga faktor yang menyebabkan siswa membolos dari sekolah yaitu faktor sekolah, personal serta keluarga [1][2]. Faktor sekolah yang berisiko meningkatkan munculnya perilaku membolos pada remaja antara lain kebijakan mengenai pembolosan yang tidak konsisten dan suasana sekolah yang membosankan bagi siswa. Faktor personal misalnya terkait dengan menurunnya motivasi atau hilangnya minat akademik siswa, kondisi ketinggalan pelajaran, atau karena kenakalan remaja seperti konsumsi alkohol dan minuman keras. Sedangkan faktor keluarga

meliputi pola asuh orang tua atau kurangnya partisipasi orang tua dalam pendidikan anak.

Dari ketiga faktor yang telah diuraikan tadi, mungkin faktor keluarga merupakan faktor utama yang harus dibenahi. Kurangnya partisipasi orang tua dalam pendidikan seharusnya bisa diatasi dengan memperbanyak fasilitas-fasilitas untuk orang tua agar bisa mengawasi kegiatan anaknya di sekolah. Perlu digaris bawahi bahwa fasilitas ini harus efektif dan efisien baik dari segi harga maupun waktu. Fasilitas tersebut dapat berupa sebuah media pengawasan absensi yang informasinya bisa langsung diakses oleh orang tua siswa. Seiring perkembangan teknologi fasilitas tersebut dapat dipenuhi yaitu dengan membuat sebuah aplikasi absensi menggunakan teknologi RFID dan media komunikasi SMS Gateway.

II. TINJAUAN PUSTAKA

A. Radio Frequency Identification

Radio Frequency Identification (RFID) adalah sebuah teknologi yang secara konseptual mirip dengan *barcode* yang banyak dijumpai pada barang dagangan, atau *magnetic strips* yang banyak digunakan di lingkungan kartu kredit. Perbedaan *barcode* dengan RFID adalah RFID memiliki sebuah *chip* elektronik yang bisa dijejali dengan aneka macam data atau informasi sesuai kebutuhan [3].

RFID terdiri atas dua bagian yaitu *reader* dan *transponder*. RFID *Reader* yaitu alat pembaca *transponder* atau *tag* RFID yang kompatibel dengan frekuensi yang dikirimkan oleh *reader*, sedangkan *transponder* atau *tag* adalah media dengan bentuk tertentu yang dapat menerima frekuensi dari RFID *reader* untuk dibaca dan diproses menurut kebutuhan dari aplikasi tertentu. Sebuah *tag* RFID terdiri dari sebuah *microchip* dan sebuah antena. *Chip* tersebut menyimpan nomor seri yang unik atau informasi lainnya tergantung kepada tipe memorinya [4].

B. Transmisi Data

Transmisi data merupakan proses untuk melakukan pengiriman data dari salah satu sumber data ke penerima data menggunakan komputer atau media elektronik. Terdapat dua mode pada proses transmisi data yakni mode transmisi paralel dan mode transmisi serial. Transmisi paralel dapat dikirimkan serentak menggunakan beberapa jalur sekaligus.

Jadi untuk mode transmisi model ini jalur yang digunakan tentu lebih dari satu media transmisi. Data akan dikirimkan terus menerus menggunakan jalur-jalur yang disediakan tersebut sampai semua data selesai dikirimkan. Pada transmisi serial jalur yang disediakan hanya ada satu, dimana data nantinya akan dikirimkan secara bergantian hingga semua data tersebut nanti dapat diterima oleh pengirim. Transmisi serial memiliki dua mode transmisi, yaitu *synchronous transmission* dan *asynchronous transmission* [5].

C. Short Message Services

Short Message Services (SMS) adalah salah satu tipe *Instant Messaging* (IM) yang memungkinkan untuk bertukar pesan singkat kapanpun walaupun *user* sedang melakukan panggilan data atau suara. SMS merupakan layanan pesan singkat berbasis tulisan atau teks dengan media komunikasi ponsel. Pesan teks yang biasa digunakan adalah huruf, angka, atau karakter alfanumerik.

Gammu adalah sebuah proyek yang meliputi aplikasi, *script* dan *driver* untuk mengelola berbagai fungsi pada telepon seluler dan perangkat sejenis. Gammu merupakan aplikasi SMS *gateway* populer yang banyak digunakan oleh para pengembang aplikasi. Gammu bisa diimplementasikan diberbagai bahasa aplikasi pemrograman [6].

D. Sistem Terdistribusi

Sistem terdistribusi merupakan kumpulan komputer *independent* yang tampak oleh *user* sebagai satu sistem komputer. Sistem terdistribusi juga bisa diartikan sebagai kumpulan *computer autonomy* yang dihubungkan oleh jaringan dengan *software* yang dirancang untuk menghasilkan fasilitas komputasi terintegrasi. *Client-Server* adalah arsitektur jaringan yang memisahkan *client* dengan *server*. Masing-masing *client* dapat meminta data atau informasi dari *server*. Sistem *client server* didefinisikan sebagai sistem terdistribusi [7].

E. Unified Modelling Language

UML (*Unified Modeling Language*) adalah sebuah bahasa yang berdasarkan grafik atau gambar untuk memberikan visualisasi, memberikan spesifikasi, serta membangun dan mendokumentasikan pengembangan *software* berbasis objek (*Object-Oriented*). UML sendiri juga memberikan standar penulisan sebuah sistem *blueprint* yang meliputi konsep bisnis proses, penulisan kelas-kelas dalam bahasa program yang spesifik, skema database dan komponen-komponen yang diperlukan dalam sistem *software*. UML adalah standar dunia yang dibuat oleh *Object Management Group* (OMG), sebuah badan yang bertugas mengeluarkan standar-standar teknologi *object-oriented* dan *software component* [8].

III. METODE PENELITIAN

Metodologi penelitian yang penulis lakukan dibagi ke dalam dua bagian, yaitu pengumpulan data dan pengembangan sistem. Dalam penelitian ini peneliti menggunakan penelitian kualitatif. Karena data dan informasi

yang peneliti kumpulkan lebih banyak bersifat keterangan-keterangan atau penjelasan yang bukan berbentuk angka.

A. Metode Pengumpulan Data

Penulis membagi metode pengumpulan data pada penyusunan skripsi ini menjadi dua bagian yaitu dengan melakukan studi lapangan dan studi pustaka. Studi lapangan dilakukan dengan melakukan pengamatan langsung kepada objek yang penulis teliti, dalam hal ini SMA Negeri 2 Karawang. Studi kepustakaan dilakukan oleh penulis untuk menghimpun informasi yang relevan dengan RFID dan bagaimana RFID bisa dijadikan sebuah teknologi yang bisa dijadikan solusi untuk dijadikan alat absensi. Informasi-informasi tersebut penulis peroleh dari buku-buku ilmiah, laporan penelitian, jurnal ilmiah, skripsi, tesis, disertasi dan sumber-sumber tertulis lain baik tercetak maupun elektronik.

B. Metode Pengembangan

Metodologi pengembangan sistem yang digunakan dalam pengembangan aplikasi yang penulis buat adalah menggunakan metodologi pemodelan *waterfall*. Menurut Pressman model sekuensial linier mengusulkan sebuah pendekatan kepada perkembangan perangkat lunak yang sistematis dan sekuensial yang dimulai pada tingkat dan kemajuan sistem pada seluruh analisis, desain, kode pengujian, dan pemeliharaan. Setidaknya ada lima tahapan yang harus dilakukan dalam pemodelan menggunakan model sekuensial linear atau *waterfall* [9].

1. Analisis kebutuhan perangkat lunak
2. Desain
3. Generasi Kode
4. Pengujian
5. Pemeliharaan

IV. HASIL DAN PEMBAHASAN

A. Analisis Permasalahan

Absensi merupakan salah satu tolak ukur kedisiplinan siswa di sekolah. Tidak adanya media komunikasi antara pihak sekolah dan pihak orang tua membuka peluang siswa untuk membolos dari sekolah. Hal ini akan mengakibatkan tingkat kedisiplinan siswa menurun yang kemudian prestasi siswa juga akan ikut menurun.

Sistem absensi yang sedang berjalan pada SMA Negeri 2 Karawang masih menggunakan cara manual yaitu dengan diabsen oleh sekretaris dari masing-masing kelas. Hal ini tentunya masih membuka peluang manipulasi data, kerusakan data dan kehilangan data. Selain itu sistem absensi tersebut hanya bisa bermanfaat bagi pihak sekolah tidak bagi pihak orang tua. Pihak orang tua hanya bisa mengetahui data tersebut saat putra/i mereka melebihi batas toleransi ketidakhadiran dari pihak sekolah melalui guru piket dengan dipanggil ke sekolah. Hal ini yang kemudian membuat pihak sekolah tidak dapat melakukan tindakan preventif pada kasus tersebut. Pihak sekolah memerlukan media komunikasi yang cepat, murah dan efisien untuk mengkomunikasikan

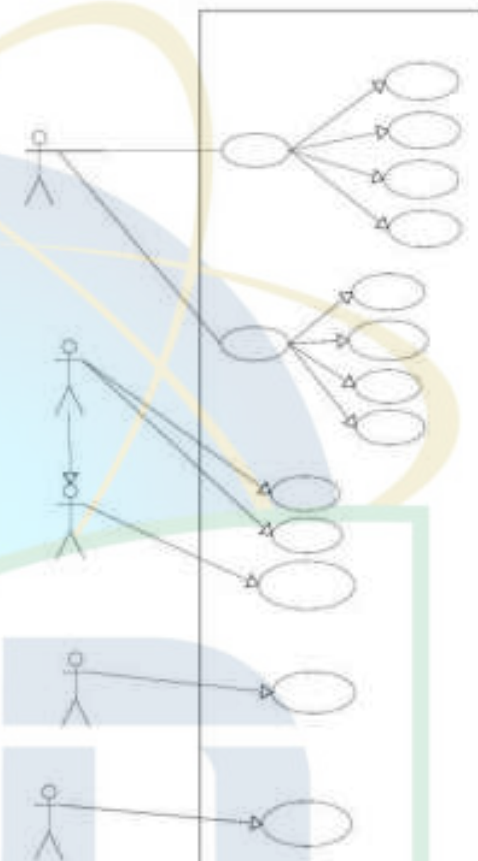
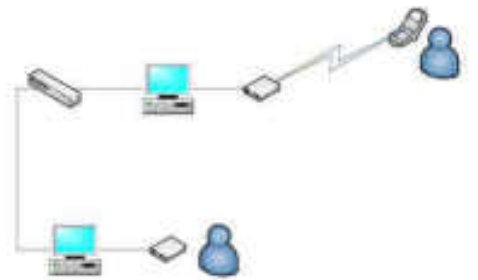
kedisiplinan siswa dalam hal absensi kepada pihak orang tua siswa. Media komunikasi tersebut bisa menggunakan pesan singkat (SMS) yang memberitahukan pihak orang tua bahwa putra/i mereka telah datang dan melakukan absensi di sekolah. Hal ini diharapkan dapat membantu mengurangi angka kasus membolos dan dapat memberikan tindakan preventif yang mencegah siswa membolos dari sekolah.

B. Perancangan Perangkat Lunak

Cukup tingginya angka permasalahan siswa tidak hadir tanpa izin atau biasa disebut bolos di sekolah, maka pihak sekolah memerlukan sebuah alat bantu untuk mengurangi angka siswa membolos. Alat bantu tersebut merupakan aplikasi absensi yang harus memenuhi kebutuhan sebagai berikut:

1. Dapat mencatat kehadiran siswa secara detil, baik itu nama siswa, kelas, jam siswa tiba di sekolah dan status kehadiran siswa.
2. Dapat mencatat keterangan siswa tidak hadir.
3. Dapat memberikan keterangan kehadiran siswa tersebut kepada orang tua siswa masing-masing secara *real time*.
4. Proses absensi harus berjalan cepat sehingga tidak terjadi antrian saat siswa melakukan absensi.
5. Dapat memberikan laporan absensi per kelas untuk masing-masing wali kelas.

Penulis membutuhkan beberapa alat yang dapat memenuhi kebutuhan-kebutuhan yang telah diuraikan sebelumnya. Beberapa alat yang akan digunakan dalam aplikasi tersebut antara lain RFID *reader* ID-12, *tag*/kartu RFID, kabel *converter* RS232 ke USB dan modem GSM untuk mengirim SMS. Berikut ini adalah gambaran dari arsitektur sistem dan *use case diagram* dari Aplikasi Absensi Berbasis *Smart Card* RFID.



Gambar 1. Arsitektur sistem dan *use case diagram*

Database yang digunakan pada aplikasi absensi siswa adalah MySQL [10]. Skema dari *database* dapat dilihat pada Gambar 2.

Setelah proses perancangan selesai maka tahapan selanjutnya adalah generasi kode atau biasa disebut dengan *coding*. Pada tahap generasi kode desain diubah menjadi baris-baris kode sehingga aplikasi bisa melakukan perintah sesuai dengan yang diharapkan pada proses desain.



Gambar 2. Skema database

1) Kodifikasi Membuka Layanan Server.

Aplikasi absensi yang penulis buat adalah aplikasi berbasis *client-server* oleh karena itu perancangan aplikasi ini akan menghasilkan dua buah aplikasi yakni aplikasi *client* dan aplikasi *server*. Aplikasi ini dibangun menggunakan Java RMI. Agar kedua aplikasi ini bisa saling berkomunikasi (*client* bisa melakukan *request* dan *server* bisa memberikan *respond*) maka aplikasi *server* harus membuka layanan di salah satu *port* komputer. Untuk membuka layanan Java RMI penulis membuat sebuah objek dari kelas *Registry* [11-14].

```
Registry registry = new Registry();
```

Objek tersebut kemudian akan menempati *port* tertentu (contoh di bawah ini akan membuka layanan RMI pada *port* 1234) di komputer menggunakan perintah.

```
registry=LocateRegistry.createRegistry(12 34);
```

Objek *registry* tersebut kemudian akan “mengikat” metode-metode dalam suatu kelas agar objek *registry* bisa melakukan layanan-layanan tertentu seperti menyimpan data ke dalam *database*. Untuk melakukan hal tersebut objek *registry* hanya perlu memanggil metode *rebind* dan memberikan nama pada layanan tersebut serta objek dari kelas yang memiliki metode untuk melakukan layanan tersebut.

```
ServiceServerAbsen service = new ServiceServerAbsen();
registry.rebind("serviceAbsen",service);
```

2) Kodifikasi Menutup Layanan Server.

Layanan di *server* harus dapat ditutup saat terjadi masalah teknis atau sedang terjadi pemeliharaan *server*. Untuk menutup layanan di *server* sehingga *client* tidak bisa lagi melakukan *request* objek *registry* yang sebelumnya dibuat hanya perlu memanggil metode *unbind* diikuti dengan nama *service* yang sebelumnya dideklarasikan.

```
registry.unbind("serviceAbsen");
```

3) Kodifikasi Mendapatkan Layanan Server.

Aplikasi *client* akan melakukan *request* ke *port* dari komputer *server* yang digunakan sebagai layanan aplikasi tersebut. *Request* tersebut dilakukan oleh kode sebagai berikut (misalkan aplikasi *client* ingin melakukan *request* layanan atur siswa di *server*).

```
Registry registry = LocateRegistry.getRegistry("localhost",
1234);
SiswaService siswa = (SiswaService) registry.lookup
("serviceSiswa");
```

4) Kodifikasi Mendapatkan Data Kartu RFID.

Data dari kartu RFID bisa dibaca melalui *reader* yang terhubung dengan komputer yang menjalankan aplikasi absensi siswa *client* dengan memanfaatkan *interface* dari kelas *SerialPortListener*. Pada metode *SerialEvent* yang didapat dari implementasi *interface* jika event yang ditangkap oleh *listener* berisi data maka data tersebut akan ditangkap dan “dialirkan” dengan menggunakan kelas *InputStreamReader* yang kemudian akan ditampung pada sebuah *String*.

5) Kodifikasi Menyisipkan Data Absensi Siswa ke Database.

Setelah data kartu dibaca oleh RFID *reader*, aplikasi akan mengecek nomor kartu yang merupakan representasi dari data siswa di dalam *database*. Aplikasi akan mengecek apakah siswa dengan nomor kartu tersebut ada di dalam *database* dan belum melakukan absensi pada hari itu. Jika benar maka aplikasi akan menyimpan data absensi siswa tersebut ke dalam *database*. Penulis menggunakan JDBC (Java Database Connectivity) untuk menghubungkan aplikasi Java yang penulis buat dengan *database server* MySQL. Untuk melakukan koneksi dengan MySQL maka diperlukan beberapa properti seperti *hostname*, *database username* dan *password database*.

```
DriverManager.registerDriver(new
com.mysql.jdbc.Driver());
Connection connection = null;
connection =
DriverManager.getConnection(url, user, pass);
```

Koneksi tersebut digunakan untuk objek *statement* atau *preparedstatement* yang berisi perintah-perintah SQL agar bisa menyisipkan data ke dalam *database* MySQL.

```
PreparedStatement statement;
String sql = "INSERT INTO log_absen (id_log,
tanggal_absen, waktu, nis, keterangan, sumber_absen,
status) VALUES (null, ?, ?, ?, ?, ?, ?)";
statement = connection.
prepareStatement(sql); statement.setDate(1, new Date(new
java.util.Date().getTime()));
statement.setString(2, absen.getTime());
statement.setString(3, nis); statement.setString(4,
absen.getKeterangan()); statement.setString(5,
absen.getSumberAbsen()); statement.setInt(6, 0);
statement.executeUpdate();
```

6) Kodifikasi Mengirim SMS detil Absensi Siswa

Setelah data absensi tersimpan ke dalam *database*, secara periodik aplikasi akan mengecek data absensi yang belum dikirim kepada orang tua siswa masing-masing. Jika data tersebut tersedia maka data tersebut akan ditampung ke dalam sebuah *list*, kemudian aplikasi akan membuat *file* berekstensi **.bat* yang berisikan perintah untuk mengirimkan pesan singkat (SMS) kepada SMS gateway (Gammu). File tersebut kemudian akan dieksekusi oleh aplikasi sehingga Gammu akan mengirim SMS tersebut. Setelah SMS tersebut dikirim maka status absensi di dalam *database* akan dirubah menjadi terkirim.

Setelah proses kodifikasi selesai dan kode-kode tersebut diintegrasikan dengan desain tampilan aplikasi, maka tahap selanjutnya adalah pengujian aplikasi. Pengujian dilakukan dengan menggunakan metode pengujian *black box* untuk memastikan setiap fungsi dari aplikasi berjalan sesuai dengan yang diharapkan.

V. KESIMPULAN

Komunikasi antara pihak sekolah dengan orang tua mengenai kedisiplinan anak bisa diakomodir oleh aplikasi absensi yang penulis rancang. Aplikasi ini bisa memberikan informasi detail mengenai kedatangan siswa ke sekolah dan mengirimkannya melalui pesan singkat yang dikirim ke ponsel orang tua siswa masing-masing. Hal ini kemudian membuat pihak sekolah lebih mudah untuk menjalin komunikasi dengan para orang tua siswa dan para orang tua

siswa bisa berpartisipasi lebih aktif lagi dalam pengontrolan kedisiplinan putra ataupun putrinya datang ke sekolah.

DAFTAR PUSTAKA

- [1] Kompas.com. (10 Januari 2012). *Peran Sekolah Atasi Perilaku Membolos pada Remaja* [Online]. Tersedia: <http://www1.kompas.com/read/xml/2009/02/18/23563648/peran.sekolah.atasi.perilaku.membolos.pada.remaja>
- [2] Kompas.com. (10 Januari 2012). *Waduh Ratusan Siswa Bolos UN* [Online]. Tersedia: <http://regional.kompas.com/read/2011/04/21/13181987/Waduh.Ratusan.Siswa.Bolos.UN> (Akses 10 Januari 2012)
- [3] P. J. Sweeney, "RFID For Dummies". Indianapolis: Wiley Publishing, 2005.
- [4] M. Meyers, "Mike Meyer's Comptia RFID+ Certification Passport". Osborne: McGraw-Hill, 2007.
- [5] D. C. Green, "Komunikasi Data". Yogyakarta: Penerbit Andi, 2002.
- [6] G. Le Bodie, "Mobile Messaging Technology and Services". West Sussex, England: John Wiley & Sons Ltd, 2005.
- [7] B. S. D. Oetomo, "Konsep dan Aplikasi Pemrograman Client Server dan Sistem Terdistribusi". Yogyakarta: Penerbit Andi, 2006.
- [8] A. Nugroho, "Rekayasa Perangkat Lunak Menggunakan UML dan Java". Yogyakarta: Penerbit Andi, 2010.
- [9] R. S. Pressman, "Rekayasa Perangkat Lunak, Pendekatan Praktisi (Buku Satu)". Yogyakarta: Penerbit Andi, 2012.
- [10] A. Solichin, "MySQL 5 Dari Pemula Hingga Mahir", Jakarta: Universitas Budi Luhur, 2010.
- [11] W. O. Purbo dan A. Sagitri, "Paper Sekilas Tentang RMI", 1998.
- [12] P. Deitel, "Java for Programmer Second Edition Deitel Developer Series". Boston: Pearson Education, 2012.
- [13] D. Lowe, "Java All in One Desk Reference for Dummies". Indianapolis: Wiley Publishing, 2005.
- [14] D. S. Malik, "Java Programming From Analysis to Program Design". Boston: Course Technology, 2012.