



UNIVERSITAS INDONESIA

**PERANCANGAN SISTEM *OTOMASI* PEMINJAMAN DAN
PENGEMBALIAN BUKU PADA PERPUSTAKAAN MENGGUNAKAN
TEKNOLOGI RFID**

SKRIPSI

Paramadina Hasby

040403066Y

**DEPARTEMEN TEKNIK ELEKTRO
FAKULTAS TEKNIK UNIVERSITAS INDONESIA
DESEMBER, 2009**



UNIVERSITAS INDONESIA

**PERANCANGAN SISTEM *OTOMASI* PEMINJAMAN DAN
PENGEMBALIAN BUKU PADA PERPUSTAKAAN MENGGUNAKAN
TEKNOLOGI RFID**

SKRIPSI

Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik

Paramadina Hasby

040403066Y

**DEPARTEMEN TEKNIK ELEKTRO
FAKULTAS TEKNIK UNIVERSITAS INDONESIA
DESEMBER, 2009**

HALAMAN PERNYATAAN ORISINALITAS

Skripsi ini adalah hasil karya saya sendiri,
dan semua sumber baik yang dikutip maupun dirujuk
telah saya nyatakan dengan benar.

Nama : Paramadina Hasby

NPM : 040403066Y

Tanda Tangan :

Tanggal : Desember 2009

HALAMAN PENGESAHAN

Skripsi ini diajukan oleh :

Nama : Paramadina Hasby

NPM : 040403066Y

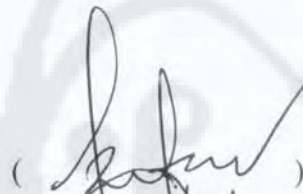
Program Studi : Elektro

Judul Skripsi : Perancangan Sistem *Otomasi* Peminjaman dan Pengembalian Buku
Pada Perpustakaan Menggunakan Teknologi RFID

Telah berhasil dipertahankan dihadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Strata 1 pada Program Studi Teknik Elektro, Fakultas Teknik, Universitas Indonesia.

DEWAN PENGUJI

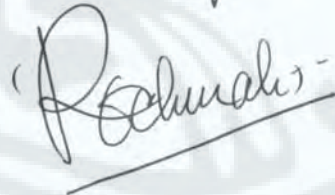
Pembimbing : Ir. Arifin Djauhari, MT

()

Penguji : Dr. Ir. Arman Djohan Diponegoro, M.Eng

()

Penguji : Ir. Rochmah N. Sukardi, M.Eng.Sc

()

Ditetapkan di : Depok

Tanggal : 23 Desember 2009

UCAPAN TERIMA KASIH

Puji syukur penulis sampaikan kepada Allah SWT atas segala Karunia dan Rahmat-Nya sehingga skripsi ini dapat terselesaikan. Penulis mengucapkan terima kasih kepada:

Ir. Arifin Djauhari, MT

Sebagai dosen pembimbing yang telah meluangkan waktunya untuk memberikan arahan, bimbingan dan diskusi sehingga skripsi ini dapat terselesaikan dengan baik. Penulis juga mengucapkan terima kasih kepada:

1. Orangtua juga kakak dan adik-adikku atas segala dukungan moril dan materiil yang telah diberikan.
2. Rekan-rekan elektro angkatan 2004 dan 2005.
3. Seluruh civitas akademika Departemen Elektro Universitas Indonesia yang tidak dapat disebutkan satu persatu.

Depok, 17 Desember 2009

Penulis



**HALAMAN PERSETUJUAN PUBLIKASI KARYA ILMIAH UNTUK
KEPENTINGAN AKADEMIS**

Sebagai civitas akademika Universitas Indonesia, saya yang bertanda tangan di bawah ini :

Nama : Paramadina Hasby
NPM : 040403066Y
Program Studi : Elektro
Departemen : Elektro
Fakultas : Teknik
Jenis Karya : Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Indonesia **Hak Bebas Royalti Noneksklusif** (*Non-Exclusive Royalty-Free Right*) atas karya ilmiah saya yang berjudul :

**PERANCANGAN SISTEM OTOMASI PEMINJAMAN DAN
PENGEMBALIAN BUKU PADA PERPUSTAKAAN MENGGUNAKAN
TEKNOLOGI RFID**

Dengan Hak Bebas Royalti Non Eksklusif ini Universitas Indonesia berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk Basis Data, merawat dan mempublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok

Pada tanggal : 23 Desember 2009

Yang menyatakan

(Paramadina Hasby)

ABSTRAK

Nama : Paramadina Hasby

Program Studi : Teknik Elektro

Judul : Perancangan Sistem *Otomasi* Peminjaman dan Pengembalian Buku pada Perpustakaan Menggunakan Teknologi RFID

Pembimbing : Ir. Arifin Djauhari, MT

Pada skripsi ini dirancang sebuah sistem *otomasi* peminjaman dan pengembalian buku pada perpustakaan menggunakan RFID *reader* Starter Kit ID-12 dan *tag* berformat EM4001 dengan fitur notifikasi masa pinjam *via* email. Sistem peminjaman dan pengembalian buku ini menggunakan RFID *reader* Starter Kit ID-12 untuk mendeteksi ID *member* peminjam dan *tag* buku pinjaman. Data transaksi peminjaman dan pengembalian yang terdeteksi oleh RFID tersebut akan diolah secara langsung oleh *software* aplikasi yang dibangun menggunakan Delphi 7 yang terinstalasi pada komputer untuk kemudian disimpan ke dalam *database*.

Sistem *otomasi* peminjaman dan pengembalian buku ini juga menggunakan teknologi email (electronic mail) untuk pemberitahuan masa pinjam yang akan segera habis kepada seluruh *member* yang melakukan peminjaman.

Hasil pengujian terhadap sistem *otomasi* peminjaman dan pengembalian buku ini, baik dari aspek fungsionalitas maupun aspek durabilitas sistem, menunjukkan bahwa sistem ini dapat bekerja dengan baik.

Kata kunci : Perpustakaan, RFID, Delphi 7, email

ABSTRACT

Name : Paramadina Hasby

Study Program: Electrical Engineering

Title : The Design of Automatic Check-out and Check-in Books System in Library
Using RFID Technology

Supervisor : Ir. Arifin Djauhari, MT

This thesis design an automatic check-out and check –in books system in library based on RFID reader Starter Kit ID-12 and tag EM4001 technologies which have feature parking fee debit via SMS. This automatic check-out and check-in books system use RFID reader Starter Kit ID-12 to identify . The record of check-out and check-in books from RFID reader will be proceed by application software builded with Delphi 7 which is installed in computer and then are saved to *database*.

This automation system is also using email application for notifying borrowing period that is going to be expired to all of library members that borrowed library books.

The result of testing this check-out and check-in system, either functionality aspect or durability aspect, show that this system do well.

Keyword : Library, RFID, Delphi 7, email

DAFTAR ISI

| | |
|---|-----|
| HALAMAN JUDUL | i |
| HALAMAN PERNYATAAN ORISINALITAS | ii |
| LEMBAR PERSETUJUAN | iii |
| UCAPAN TERIMA KASIH..... | iv |
| ABSTRAK..... | v |
| ABSTRACT..... | vi |
| DAFTAR ISI..... | vii |
| DAFTAR TABEL..... | x |
| DAFTAR GAMBAR..... | xi |
| BAB 1 PENDAHULUAN..... | 1 |
| 1.1 Latar Belakang..... | 1 |
| 1.2 Tujuan Penulisan..... | 2 |
| 1.4 Batasan Masalah | 3 |
| 1.5 Sistematika Penulisan | 3 |
| BAB 2 KONSEP DASAR <i>RADIO FREQUENCY IDENTIFICATION (RFID)</i> | 4 |
| 2.1 SISTEM RFID..... | 4 |
| 2.2 RFID <i>Tag</i> | 5 |
| 2.2.1 <i>Passive RFID Tag</i> | 6 |
| 2.2.2 <i>Active RFID Tag</i> | 7 |
| 2.2.3 <i>Semipasif RFID Tag</i> | 7 |
| 2.3 RFID <i>Reader</i> | 9 |
| 2.4 Pengklasifikasian Frekuensi Operasi pada Sistem RFID..... | 12 |
| 2.5 Sistem Pada Beberapa Kategori RFID..... | 14 |
| 2.5.1 Sistem Pada RFID Pasif..... | 14 |
| 2.5.2 Sistem Pada RFID Aktif..... | 15 |
| 2.5.3 Sistem Pada RFID Semipasif..... | 16 |
| 2.6 MAX232 <i>SERIAL LEVEL CONVERTER</i> | 16 |
| 2.7 Perangkat Lunak..... | 18 |
| 2.7.1 Basis Data..... | 18 |

| | |
|---|-----------|
| 2.7.2 Delphi 7..... | 18 |
| 2.7.2.1 IDE (Integrated Development Environment) Delphi..... | 19 |
| 2.7.2.2 Tipe Data Pada Delphi..... | 22 |
| 2.7.2.3 Macam-macam Komponen di Delphi..... | 22 |
| 2.7.2.4 Management Project..... | 25 |
| 2.7.2.5 Komunikasi Serial Pada Delphi..... | 25 |
| BAB3 PERANCANGAN PROTOTYPE..... | 27 |
| 3.1 Konsep Prototype..... | 27 |
| 3.2 Gambaran Teknis Sistem <i>Otomasi</i> Peminjaman dan Pengembalian Buku Pada Perpustakaan Menggunakan Teknologi RFID | 28 |
| 3.2.1 Arsitektur Sistem <i>Otomasi</i> Peminjaman dan Pengembalian Buku Pada Perpustakaan Menggunakan Teknologi RFID..... | 28 |
| 3.2.2 Layout Sistem <i>Otomasi</i> Peminjaman dan Pengembalian Buku Pada Perpustakaan Menggunakan Teknologi RFID..... | 29 |
| 3.3 Alur Kerja Sistem <i>Otomasi</i> Peminjaman dan Pengembalian Buku Pada Perpustakaan..... | 30 |
| 3.3.1 Sub Peminjaman..... | 30 |
| 3.3.2 Sub Pengembalian..... | 31 |
| 3.3.3 Sub Pengiriman Email Notifikasi Masa Pinjam..... | 33 |
| 3.4 Interfacing Perangkat Keras dengan Perangkat Lunak..... | 34 |
| 3.5 Penggunaan Perangkat Lunak (program utama) pada Sistem <i>Otomasi</i> Peminjaman dan Pengembalian Buku Perpustakaan..... | 34 |
| 3.6 Penggunaan Database untuk Sistem Peminjaman dan Pengembalian Buku pada Perpustakaan..... | 35 |
| BAB 4 PENGUJIAN SISTEM DAN ANALISIS..... | 40 |
| 4.1 Pengujian Pembacaan <i>Tag</i> oleh <i>Reader</i> | 41 |

| | |
|---|----|
| 4.1.1 Pengujian Pembacaan <i>Tag</i> Sebelum dan Sesudah Terhubung Aplikasi <i>Software</i> | 42 |
| 4.1.2 Pengujian Pembacaan <i>Tag</i> Oleh <i>Reader</i> dengan Berbagai Jenis Penghalang..... | 43 |
| 4.2 Pengujian Perangkat Lunak Berdasarkan Spesifikasi Fungsi..... | 44 |
| 4.2.1 Pengujian Setting <i>Hardware</i> | 44 |
| 4.2.2 Pengujian Pengaturan Scan <i>Tag</i> Buku Pada Sub Peminjaman..... | 45 |
| 4.2.3 Pengujian Pengiriman Email Notifikasi Masa Pinjam yang Akan Habis..... | 46 |
| 4.2.3.1 Pengujian Pengiriman Email Notifikasi ke Sebuah Account Email untuk Beberapa Domain Email yang Berbeda..... | 46 |
| 4.2.3.2 Pengujian Pengiriman Email Notifikasi ke Beberapa Account Email Berbeda..... | 47 |
| 4.2.4 Pengujian Penyimpanan Data ke Basis Data..... | 49 |
| 4.3 Pengujian Ketahanan Sistem Dengan RFID <i>Reader</i> | 50 |
| 4.4 Analisis Keseluruhan Perangkat Lunak Berdasarkan Spesifikasi Fungsi dari Sistem <i>Otomasi</i> Peminjaman dan Pengembalian Buku..... | 51 |
| BAB 5 KESIMPULAN..... | 53 |
| DAFTAR ACUAN | 54 |
| DAFTAR PUSTAKA..... | 56 |

DAFTAR TABEL

| | |
|--|----|
| Tabel 2.1. Perbandingan beberapa jenis <i>tag</i> | 8 |
| Tabel 2.2. Pengklasifikasian Frekuensi Operasi Sistem RFID..... | 13 |
| Tabel 3.1 Script pada Delphi untuk mengakses Database MS.Access..... | 36 |
| Tabel 3.2 Script pada Delphi untuk melakukan pengiriman email notifikasi masa pinjam..... | 38 |
| Tabel 4.1 Hasil pengujian interval waktu pembacaan <i>tag</i> sebelum terhubung aplikasi <i>software</i> | 42 |
| Tabel 4.2 Hasil pengujian interval waktu pembacaan <i>tag</i> sesudah terhubung aplikasi <i>software</i> | 42 |
| Tabel 4.3 Hasil Pengujian Pembacaan <i>Tag</i> dengan Berbagai Penghalang..... | 43 |
| Tabel 4.4 Hasil Pengujian Scan Buku dengan Batasan Waktu 15 detik..... | 46 |
| Tabel 4.5 Pengiriman Email ke account email dengan domain berbeda..... | 47 |
| Tabel 4.6 Hasil Pengujian Pengiriman Email ke Beberapa Account Email Berbeda..... | 48 |
| Tabel 4.7 Hasil Pengujian Ketahanan Sistem Dengan RFID Menyala selama 6 jam..... | 50 |

DAFTAR GAMBAR

| | |
|--|----|
| Gambar 2.1 Komponen Utama Sistem RFID..... | 5 |
| Gambar 2.2 <i>Layout</i> Dasar RFID <i>Tag</i> | 6 |
| Gambar 2.3 Bentuk-bentuk <i>Tag</i> | 9 |
| Gambar 2.4 <i>Reader</i> MIFARE buatan Panasonic..... | 10 |
| Gambar 2.5 Gambar 2.5. IC <i>reader</i> ID-12 buatan Innovative..... | 10 |
| Gambar 2.6 Bagan 2 Blok Fungsi dari <i>Reader</i> | 11 |
| Gambar 2.7 Blok diagram HF <i>interface</i> untuk <i>Inductive coupled RFID system</i> | 11 |
| Gambar 2.8. Diagram blok dari <i>control unit</i> | 12 |
| Gambar 2.9. Proses Catu Daya pada Sistem RFID Pasif..... | 14 |
| Gambar 2.10 Konfigurasi Pin pada DB9..... | 17 |
| Gambar 2.11. Rangkaian Skematik untuk IC MAX232..... | 17 |
| Gambar 2.12 IDE Delphi 7..... | 20 |
| Gambar 3.1 Arsitektur Sistem <i>otomasi</i> pada perpustakaan..... | 28 |
| Gambar 3.2 Gambar 3.2(a) <i>Layout</i> Sistem check-out.(b) <i>Layout</i> Sistem check-in..... | 29 |
| Gambar 3.3 Diagram alir Sub pengembalian..... | 32 |
| Gambar 3.4 Diagram alir pengiriman email notifikasi..... | 33 |
| Gambar 3.5 Tampilan Awal Aplikasi untuk Sub Peminjaman Buku..... | 35 |
| Gambar 3.6 Tampilan Awal untuk Aplikasi Sub Pengembalian Buku..... | 35 |
| Gambar 4.1 (a) Setting Comport, dan (b) Koneksi Basis Data..... | 44 |

| | |
|---|----|
| Gambar 4.2 Tampilan Email Notifikasi masa pinjam buku yang diterima oleh <i>Member</i> Perpustakaan..... | 47 |
| Gambar 4.3 Tampilan Basis Data Transaksi Peminjaman Pada Delphi..... | 49 |
| Gambar 4.4 Tampilan Menu Ubah Data..... | 50 |



BAB 1

PENDAHULUAN

1.1. Latar Belakang Masalah

Pelayanan akan fasilitas umum merupakan hal yang sangat penting untuk diperhatikan karena fasilitas ini menyangkut kepentingan orang banyak. Peningkatan akan pelayanan fasilitas umum ini menjadi tuntutan yang harus dipenuhi oleh para penyedia jasa. Salah satu fasilitas umum yang banyak dimanfaatkan oleh berbagai kalangan maupun instansi adalah perpustakaan. Perpustakaan merupakan pusat dari kumpulan buku-buku dan arsip dari berbagai disiplin ilmu yang tentunya terdiri dari jutaan item. Hal ini membutuhkan manajemen yang baik untuk menjamin semua kegiatan yang berlangsung di perpustakaan mulai dari transaksi peminjaman dan pengembalian buku hingga inventory buku-buku tersebut dapat berjalan dengan baik dan lancar.

Peningkatan akan pelayanan terhadap perpustakaan ini dapat didukung oleh perkembangan teknologi. Teknologi sebelumnya yang digunakan sebagai solusi terhadap sistem manajemen perpustakaan ini adalah *barcode*. Namun *barcode* ini mempunyai keterbatasan dalam pembacaan yang harus *line of sight* dan mudah untuk diduplikasi. Teknologi saat ini yang sedang dikembangkan untuk menyempurnakan kelemahan *barcode* tersebut adalah RFID (*Radio Frequency Identification*). Teknologi RFID telah digunakan untuk berbagai kepentingan seperti kontrol akses, *supply chain*, sistem perpustakaan, sistem parkir, sistem pembayaran jalan tol, sistem absensi dll[1]. Dengan menggunakan RFID, kegiatan yang biasanya menggunakan operasi manual kini dapat dilakukan dengan cara otomatis yang *contactless*, melewati bahan non konduktor seperti karton kertas dengan kecepatan akses beberapa ratus *tag* per detik pada jarak beberapa (+100) meter bahkan RFID produk IBM telah mencapai daya jangkau 200 meter. Fakta yang juga sangat mendukung banyaknya penggunaan

RFID ini adalah bahwa RFID dapat dibaca sekaligus secara massal hingga ratusan label (*tag*) per detik[2].

Aplikasi RFID untuk sistem manajemen perpustakaan ini diterapkan untuk menjalankan beberapa fungsi yaitu *self check-in* dan *check-out* buku, pengecekan inventaris buku, penyusunan buku pada rak dan pintu pemeriksaan anti pencurian[3].

Aktifitas pada perpustakaan yang berhubungan erat dengan kepuasan anggota adalah pelayanan peminjaman dan pengembalian buku. Sistem peminjaman dan pengembalian buku menggunakan RFID diterapkan dalam bentuk mekanisme *self check-in* dan *self check-out station*. Layaknya sebuah mesin ATM, peminjam selain berinteraksi dengan *reader* juga harus melakukan beberapa aktifitas pada komputer untuk melakukan transaksi peminjaman dan pengembalian buku tersebut. Meski aktifitas peminjaman dan pengembalian buku dapat dilakukan secara pribadi oleh tiap *customer* tanpa harus melalui tahapan pencatatan pada petugas, namun hal ini masih kurang efektif karena *customer* harus melakukan beberapa interaksi dengan komputer sebelum dapat melakukan scan pada *reader*. Solusi yang dapat dikembangkan adalah dengan merancang sistem otomasi agar customer tidak perlu berinteraksi dengan komputer tetapi dapat langsung melakukan scan pada *reader*. Sebagai fungsi pengingat maka pada sistem ini perlu dilengkapi komponen kontrol tambahan untuk mengatasi masalah peredaran buku diluar bila *member* lupa mengembalikan buku tepat waktu. Komponen ini berupa email pemberitahuan otomatis yang dikirim ke alamat email peminjam bila masa pinjam sudah mendekati waktu pengembalian.

1.2 Tujuan

Tujuan dari penyusunan skripsi ini adalah merancang sebuah prototype sistem otomasi agar proses peminjaman dan pengembalian buku pada perpustakaan dapat memudahkan anggota dengan menggunakan teknologi RFID disertai fitur email notifikasi masa pinjam.

1.3 Batasan Masalah

Pembahasan hanya dibatasi pada perancangan prototype proses peminjaman dan pengembalian buku dimana hanya digunakan satu *reader*. Perangkat lunak yang digunakan yaitu Delphi 7. Penggunaan kabel data untuk interfacing dari *reader* ke komputer tidak murni USB tapi ada konversi serial.

1.4 Sistematika Penulisan

Skripsi ini dibagi menjadi 5 bab. Sistematika penulisan yang diterapkan dalam skripsi ini menggunakan urutan sebagai berikut :

Bab 1 Pendahuluan, berisikan tentang latar belakang pemilihan tema, perumusan masalah, tujuan, batasan masalah dan sistematika penulisan.

Bab 2 RFID, berisikan tentang teori RFID sistem mulai dari definisi, cara kerja, komponen-komponen dasar dan tambahan yang digunakan untuk sistem otomasi peminjaman dan pengembalian buku pada perpustakaan.

Bab 3 Konsep Prototype, memaparkan konsep dari penggabungan proses peminjaman dan pengembalian buku dengan pintu deteksi anti pencurian pada perpustakaan, rancangan serta cara kerja sistem secara umum.

Bab 4 Pengujian dan analisis perangkat lunak untuk proses peminjaman dan pengembalian buku yang digabungkan dengan fungsi pintu deteksi anti pencurian.

Bab 5 Kesimpulan, berisikan tentang kesimpulan dan saran dari Skripsi yang berjudul "Perancangan Sistem Peminjaman dan Pengembalian Buku Pada Perpustakaan Menggunakan Teknologi RFID".

BAB 2

KONSEP DASAR RADIO FREQUENCY IDENTIFICATION(RFID)

2.1 Sistem RFID

RFID merupakan sebuah teknologi yang menggunakan media radio frekuensi dalam proses identifikasinya. Identifikasi dilakukan secara otomatis terhadap obyek-obyek atau manusia tanpa memerlukan operasi manual. Maka dapat disimpulkan, RFID adalah teknologi penangkapan data yang dapat digunakan secara elektronik untuk mengidentifikasi, melacak dan menyimpan informasi yang tersimpan dalam *tag* RFID [4].

RFID adalah sebuah metode identifikasi dengan menggunakan sarana yang disebut label RFID atau transponder untuk menyimpan dan mengambil data jarak jauh. Label atau kartu RFID adalah sebuah benda yang bisa dipasang atau dimasukkan di dalam sebuah produk, hewan atau bahkan manusia dengan tujuan untuk identifikasi menggunakan gelombang radio [5].

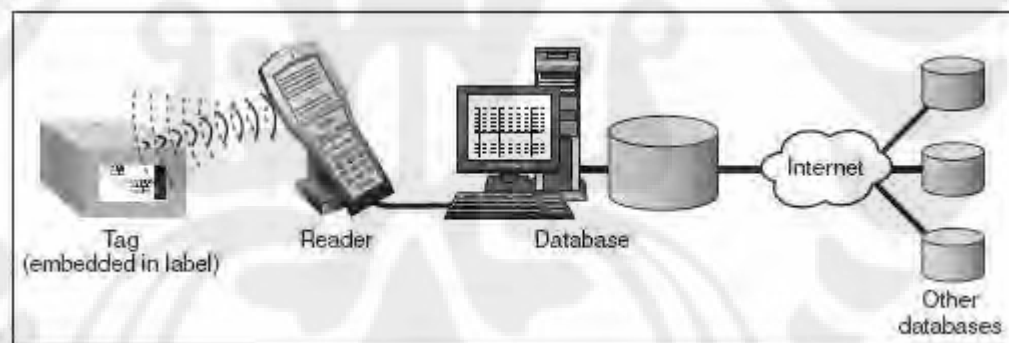
Teknologi RFID merupakan pengembangan dari teknologi identifikasi sebelumnya yaitu *barcode*. Keunggulan dari RFID yang amat signifikan dibandingkan dengan *barcode* adalah bahwa [6]:

- 1) Data yang dapat disimpan oleh *tag* RFID lebih banyak serta dapat dilakukan program ulang atas data yang tersimpan didalamnya.
- 2) Otomasi : Barkode harus di-*scan* secara optik, memerlukan kontak *line-of-sight* dengan *reader* sehingga peletakan fisik dari obyek yang di-*scan* haruslah tepat. Dan bahkan diperlukan campur tangan manusia untuk melakukan scanning terhadap *barcode*. Sedangkan *tag-tag* RFID dapat dibaca tanpa kontak *line-of-sight* dan tanpa penempatan yang presisi.

RFID terdiri atas 3 komponen utama, yaitu :

- a. *RFID Tag*, berisi antenna sebagai *coupling element* yang memungkinkan untuk menerima dan merespon terhadap suatu query yang dipancarkan oleh suatu *RFID reader*. Juga terdapat chip yang mampu menyimpan sejumlah informasi unik.
- b. *Terminal Reader RFID*, berisi modul frekuensi radio (*transmitter* dan *receiver*), pengontrol dan *coupling element* ke *tag*
- c. *Host computer*, terdiri atas Basis data yang menyimpan semua data yang terkandung dalam *tag*. Sistem computer yang mengatur alur informasi dari item-item yang terdeteksi dalam lingkup sistem RFID dan mengatur komunikasi antara *tag* dan *reader*.

3 komponen utama dari sistem RFID tersebut terlihat pada gambar 2.1 berikut

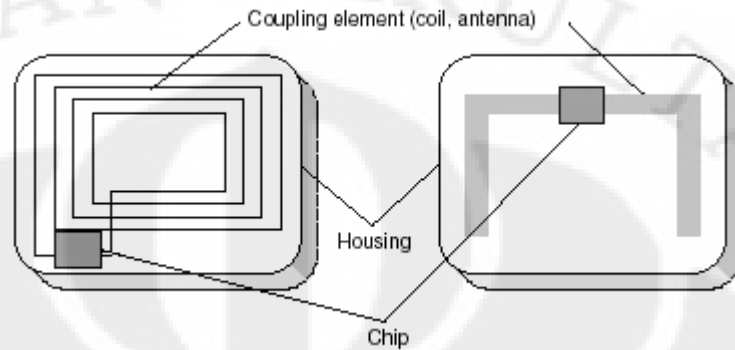


Gambar 2.1. Komponen Utama Sistem RFID[4]

2.2 RFID Tag

RFID Tag adalah sebuah benda kecil, komponen yang terdiri dari chip dan antenna. *Tag* ini dapat berupa stiker adesif yang ditempelkan pada suatu barang atau produk. Selain itu, *tag* juga dapat berupa koin dan kartu. Bentuk juga struktur dari *tag* ini dibuat sefleksibel mungkin sesuai dengan objek yang

akan diidentifikasi. Gambar 2.2 dibawah ini menunjukkan *layout* dasar dari sebuah RFID *tag* :



Gambar 2.2 *Layout* Dasar RFID *Tag*[7]

RFID *tag* ini akan bekerja ketika menerima sinyal dari RFID *reader* dan akan memantulkan kembali sinyal tersebut bersama data yang tersimpan didalam *tag* tersebut.

Ada tiga jenis RFID *tag* berdasarkan sifatnya, yaitu :

1. *Passive RFID Tag*
2. *Active RFID Tag*
3. *Semipasif RFID Tag*

2.2.1 *Passive RFID Tag*

Passive RFID Tag adalah *tag* yang bersifat pasif karena tidak memiliki power supply sendiri. *Tag* ini bekerja dengan berbekal induksi listrik yang ada pada antenna kecil yang terkandung didalamnya. Bila ada frekuensi radio scanning yang masuk dari *reader* barulah induksi listrik akan terjadi sehingga *tag* dapat merespon dengan mengirimkan data (informasi unik) yang disimpannya. Biasanya informasi sederhana seperti hanya nomor ID saja.

Dengan tidak adanya power supply pada *tag* pasif maka ukurannya bisa dibuat sekecil mungkin. Saat ini RFID *tag* yang sudah beredar di pasaran

ada yang bisa diletakkan di bawah kulit. Namun jarak jangkauan dimana *tag* ini dapat terdeteksi oleh *reader* sangat terbatas (beberapa cm).

2.2.2 *Active RFID Tag*

Active RFID Tag memiliki power supply sendiri, misalnya baterai yang dapat digunakan sebagian atau seluruhnya untuk mengaktifkan *circuit microchip* dan antena, juga untuk mengirimkan sinyal ke *reader*. Beberapa *tag* juga dapat dihubungkan dengan catu daya dari luar. *Tag* ini memiliki jarak jangkauan yang lebih jauh. Memori yang dimilikinya juga lebih besar sehingga bisa menampung berbagai macam informasi di dalamnya. Jarak jangkauan dari *RFID tag* yang aktif ini bisa sampai sekitar 10 meter dan dengan umur baterai yang bisa mencapai beberapa tahun lamanya. Kelemahannya adalah ukurannya yang lebih besar karena ada baterai tambahan. Diantara versi *tag* yang lain, *tag* aktif yang harganya paling mahal.

2.2.3 *Semi pasif RFID Tag*

Tag semi pasif menyerupai *tag* aktif dimana *tag* jenis ini juga mempunyai catu daya sendiri, namun baterai yang dimiliki hanya digunakan untuk mengaktifkan *microchip* dan antena, tidak digunakan untuk mengirimkan sinyal ke *reader*. Untuk proses *sinyal broadcast*, metodenya sama dengan cara *tag* pasif memantulkan sinyal ke *reader*.

Ketika mengaktifkan *tag*, catu daya yang dipakai adalah catu daya sendiri, sehingga tidak dipengaruhi oleh besarnya daya yang di induksikan oleh *reader*,. Tetapi, saat pengiriman balik, daya yang digunakan sepenuhnya berasal dari *reader*, sehingga akan terjadi kegagalan kalau jaraknya terlalu jauh. Tabel 2.1 merupakan perbandingan spesifikasi beberapa jenis *tag*.

Tabel 2.1. Perbandingan beberapa jenis *tag*[8]

| | <i>Tag</i> Pasif | <i>Tag</i> Semipasif | <i>Tag</i> Aktif |
|-----------------|--------------------------------|-------------------------|-------------------------|
| Catu daya | Eksternal(dari <i>reader</i>) | Baterai internal | Baterai internal |
| Rentang baca | Dapat mencapai 20 kaki | Dapat mencapai 100 kaki | Dapat mencapai 750 kaki |
| Tipe memori | Umumnya read-only | Read-write | Read-write |
| Harga | \$0.2 hingga beberapa dolar | \$2 hingga \$10 | \$20 atau lebih |
| Usia <i>tag</i> | Dapat mencapai 20 tahun | 2 sampai 7 tahun | 5 sampai 10 tahun |

Berdasarkan tipe memori yang dimiliki oleh *tag*, maka terdapat 3 variasi *tag* :

1. *Tag read-only*, *tag* ini memiliki kapasitas memori yang kecil (biasanya kurang dari 64bit) dan data yang terprogram didalamnya bersifat permanen sehingga data tidak dapat diubah. *Tag* ini hanya bisa digunakan sekali, tidak dapat reuse. Informasi yang terkandung didalam *tag* seperti ini terutama adalah informasi identifikasi item. *Tag* pasif biasanya memiliki tipe memori seperti ini.
2. *Tag read/write*, data dapat di update jika diperlukan. Namun sebagai konsekuensinya kapasitas memorinya akan lebih besar dan harganya lebih mahal dibandingkan *tag read-only*. *Tag* seperti ini biasanya digunakan bila data yang tersimpan didalamnya perlu untuk di-update seiring dengan daur hidup produk, misalnya di pabrik.
3. *Tag Write-once read-many*, informasi disimpan sekali dan tidak bias dilakukan perubahan berikutnya terhadap data tapi dapat dibaca berulang-

ulang. *Tag* tipe ini memiliki fitur keamanan *read-only* dengan menambahkan fungsionalitas tambahan dari *tag read/write*.

Dibawah ini pada Gambar 2.3 merupakan bentuk-bentuk dari *tag* :



Gambar 2.3. Bentuk-bentuk *Tag*, berikut dengan uang koin sebagai perbandingan[9]

2.3 RFID Reader

RFID *reader* merupakan salah satu komponen dasar dari sistem RFID. RFID *reader* berperan sebagai interogator yang membaca dan mendeteksi keberadaan *tag*. Sebuah *reader* menggunakan antena sendiri untuk berkomunikasi dengan *tag*. *Reader* akan memancarkan gelombang radio dengan frekuensi tertentu yang menyebabkan *tag* yang dirancang pada frekuensi tersebut serta berada pada rentang bacanya akan *memberikan* respon. Sebuah *reader* bisa berkomunikasi dengan *tag* tanpa *line of sight* langsung, tergantung kepada frekuensi radio dan tipe *tag* (aktif, pasif atau semipasif) yang digunakan. Menurut bentuknya, *reader* dapat berupa *reader*

bergerak seperti telepon genggam, atau stasioner seperti *point-of-sales* di supermarket.

Beberapa *reader* yang beredar di pasaran, sudah dikemas dalam *reader module*, tetapi ada beberapa perusahaan yang khusus menjual IC *reader*-nya saja. Ini diperuntukkan bagi mereka yang memilih pengembangan aplikasi yang lebih *costume*. Gambar 2.4 dan Gambar 2.5 merupakan contoh dari IC *reader* :

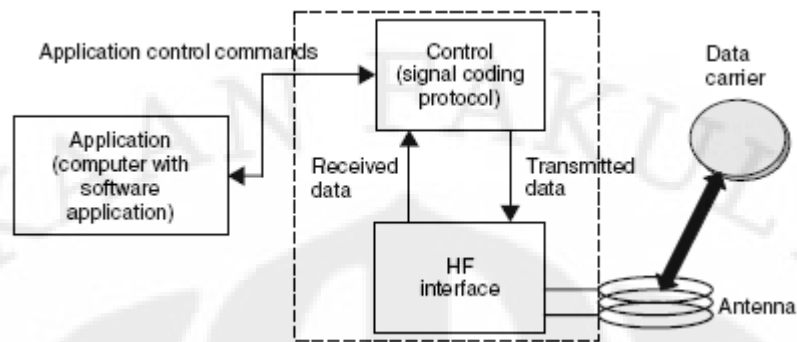


Gambar 2.4. *Reader* MIFARE buatan Panasonic[10]



Gambar 2.5. IC *reader* ID-12 buatan Innovative[10]

Reader dibedakan berdasarkan kapasitas penyimpanannya, kemampuan pemrosesannya, serta frekuensi yang dapat dibacanya. Kompatibilitas *reader* terhadap *tag* juga perlu diperhatikan karena tidak semua *tag* dengan *reader* akan *match*, ada pasangannya masing-masing. Berikut ini Gambar 2.6 adalah bagan 2 blok fungsi dari *reader*.

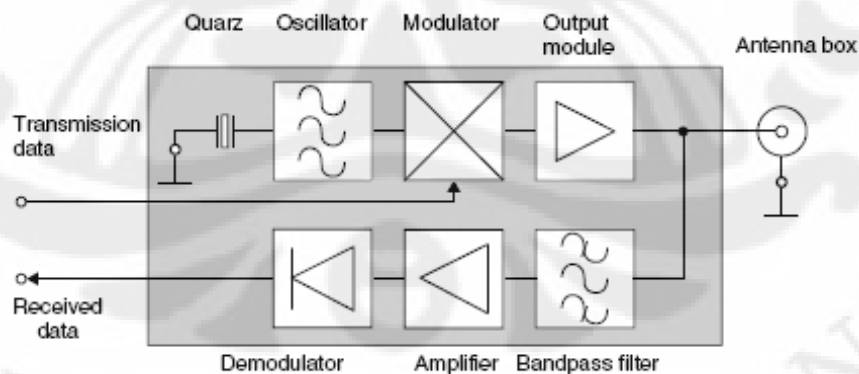


Gambar 2.6. Bagan 2 Blok Fungsi dari *Reader*[11]

a. *HF interface*

High Frequency interface digunakan oleh *reader* untuk menghasilkan sinyal transmisi agar dapat mengaktifkan *tag* dan menyuplai daya (pada *tag* pasif), memodulasi sinyal transmisi untuk mengirimkan data ke *tag*, serta menerima dan mendemodulasikan sinyal transmisi dari sebuah *tag*.

HF interface terdiri dari dua jalur sinyal yang terpisah, berdasarkan pada dua arah aliran data dari dan ke *tag*, sebagaimana ditunjukkan pada Gambar 2.7, Data yang dikirimkan ke *tag* melalui bagian transmisi dan data yang diterima dari *tag* akan di proses pada bagian *receiver*.

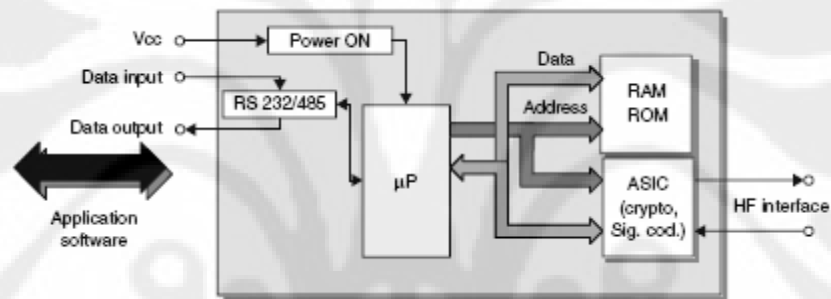


Gambar 2.7. Blok diagram *HF interface* untuk *Inductive coupled RFID system*[12]

b. *Control sistem*

Pada gambar 2.8 dibawah ini ditunjukkan bagian *control* sistem dari sebuah *reader*. *Control* sistem ini memiliki fungsi-fungsi sebagai berikut:

- berkomunikasi dengan aplikasi perangkat lunak dan melakukan perintah dari aplikasi perangkat lunak
- mengontrol metode komunikasi dengan *tag*
- melakukan proses *coding* dan *decoding* dari sinyal informasi.



Gambar 2.8. Diagram blok dari *control unit*[13]

Control unit memiliki sistem sesuai dengan mikroprosesor yang digunakan. Pertukaran data antara aplikasi perangkat lunak dan *reader* dilakukan menggunakan komunikasi serial melalui *RS232 interface* atau *RS485 interface*.

2.4 Pengklasifikasian Frekuensi Operasi pada Sistem RFID

Berdasarkan namanya, sistem RFID menggunakan frekuensi radio dalam mengidentifikasi item. Frekuensi yang digunakan dalam operasi sebuah sistem RFID mempengaruhi jarak jangkauan *reader* dalam membaca *tag*, resistansi terhadap interferensi dengan frekuensi sistem radio lain, kecepatan komunikasi data, dan ukuran antena. Frekuensi operasi yang dapat digunakan pada sistem RFID diklasifikasikan menjadi 4, yaitu :

- Low frequency (125 kHz-134 kHz)
- High frequency (13,56 MHz)
- Ultra High Frequency (868 MHz-956 MHz)
- Microwave (2,45 GHz)

Untuk frekuensi yang rendah umumnya digunakan *tag* pasif, dan untuk frekuensi tinggi digunakan *tag* aktif. Pada Tabel 2.2 dibawah ini menunjukkan klasifikasi penggunaan frekuensi operasi sistem RFID.

Tabel 2.2. Pengklasifikasian Frekuensi Operasi Sistem RFID[14]

| Frequency Band | Description | Range | Common Applications |
|----------------|----------------------------|-----------------------|---|
| 125 - 134 KHz | Low Frequency | To 18 Inches | Vehicle Identification |
| 13.56 MHz | High Frequency (HF) | Near contact - 3 feet | Electronic ticketing and fare collection; contactless payment; access control; commercial laundry and garment tracking; sample tracking. |
| 858 - 930 MHz | Ultra-high Frequency (UHF) | 1 to 30 feet | Compliance tagging and other case and pallet ID; returnable container tracking, work-in-process tracking; asset management; baggage tagging; WIFI-based RTLS. |
| 2.45 GHz | Microwave | 10+ feet | Long-range Identification with active tags |

2.5 Sistem Pada Beberapa Kategori RFID

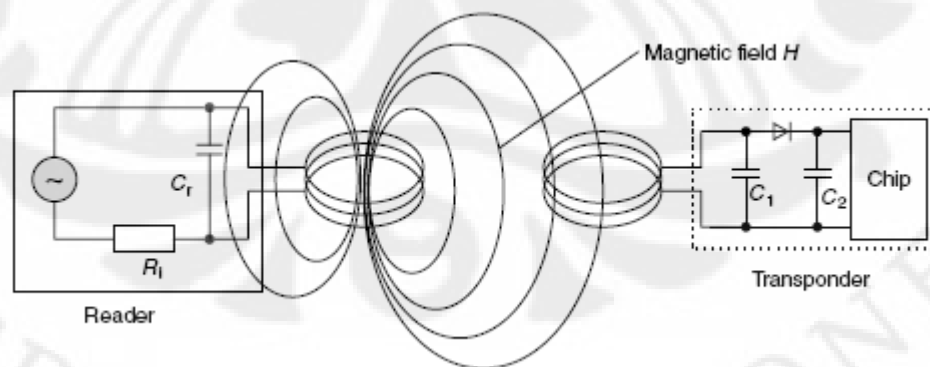
Sistem pada RFID dibagi menjadi 3 kategori berdasarkan sifat *tag* yang digunakan, yaitu sistem pada RFID pasif, sistem pada RFID aktif dan sistem pada RFID semipasif.

2.5.1 Sistem Pada RFID Pasif

Secara umum, proses yang terjadi saat sistem RFID pasif beroperasi adalah :

1. *Tag* memasuki medan elektromagnetik frekuensi radio
2. Sinyal RF mensuplai energi ke *tag*
3. *Tag* mentransmisikan ID dan data yang tersimpan didalamnya
4. *Reader* menangkap data yang dikirim oleh *tag*
5. *Reader* mengirim data dari *tag* tersebut ke komputer
6. Komputer mengolah dan menganalisis data
7. Komputer memberikan instruksi kepada *reader*
8. *Reader* mentransmisikan data ke *tag*

Pada Gambar 2.9 berikut ini adalah ilustrasi dari proses yang terjadi pada sistem RFID pasif :



Gambar 2.9. Proses Catu Daya pada Sistem RFID Pasif[15]

Inductive coupled tag yang beroperasi secara pasif, terdiri dari *microchip* tunggal dan area kumparan yang luas sebagai antena. *Reader* yang menjadi pensuplai daya untuk mengaktifkan *microchip*, dimana antena *reader* akan membangkitkan medan elektromagnetik (dengan frekuensi tinggi) yang kuat dan akan menembus area kumparan dan di sekitar kumparan.

Sebagian medan akan menembus kumparan *tag (transponder)* yang letaknya berjauhan dengan *reader* dan menghasilkan tegangan pada antena *tag* akibat proses induksi. Tegangan ini akan disearahkan oleh dioda pada rangkaian *tag* dan kemudian digunakan sebagai sumber tegangan untuk mengaktifkan chip.

Kapasitor dihubungkan secara paralel dengan kumparan antena untuk menghasilkan frekuensi resonansi yang sesuai dengan frekuensi *reader*. Kumparan antena dan kapasitor ini telah disesuaikan dengan frekuensi *reader*, dan tegangan induksi akan mencapai titik maksimum saat resonansi meningkat pada rangkaian paralel tersebut.

Pada *inductive coupled systems* proses transmisi sinyal antara kedua perangkat tersebut tergantung pada kumparan yang dimiliki pada antena keduanya. Sinyal *feedback* dari *tag* ke antena *reader* dapat direpresentasikan sebagai impedansi pada kumparan antena *reader*. Perubahan resistor beban dari *on* dan *off* akan mengubah impedansi ini dan juga mengubah tegangan pada antena *reader*.

Perubahan resistor ini diatur oleh data, sehingga data dapat dikirimkan dari *tag* ke *reader*, pengiriman ini disebut *load modulation*. Sinyal yang diterima oleh *reader* akan dimodulasi dengan cara menyearahkan tegangan yang masuk ke *reader*. Sehingga hasil modulasinya berupa amplitudo dari sinyal modulasi

2.5.2 Sistem pada RFID aktif

Dalam RFID aktif, *tag* mencatu sendiri dayanya dari baterai *onboard*, yang digunakan untuk memproses dan mengirim data. *Tag* RFID aktif mentransmit data secara periodik ketika diminta oleh *reader* atau kadang oleh *tag* itu sendiri. RFID

pasif mengirimkan data hanya ketika ada permintaan dari *reader*, itu karena RFID pasif menggunakan power dari sinyal yang dikirimkan oleh *reader*.

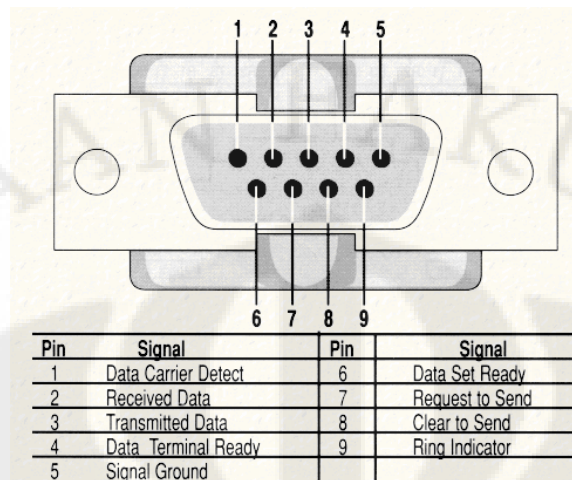
Pada RFID aktif, dikarenakan memakai baterai *onboard*, jumlah data yang dapat ditransmisikan dan jarak transmisi meningkat, tetapi keterbatasan umur baterai itu sendiri masih menjadi kendala besar yang harus dihadapi. Tidak seperti pada *tag* pasif, dimana arus informasi adalah dari *tag* ke *reader*, kebanyakan *tag* aktif dapat menyimpan informasi yang didapat dari *reader*.

2.5.3 Sistem pada RFID semi pasif

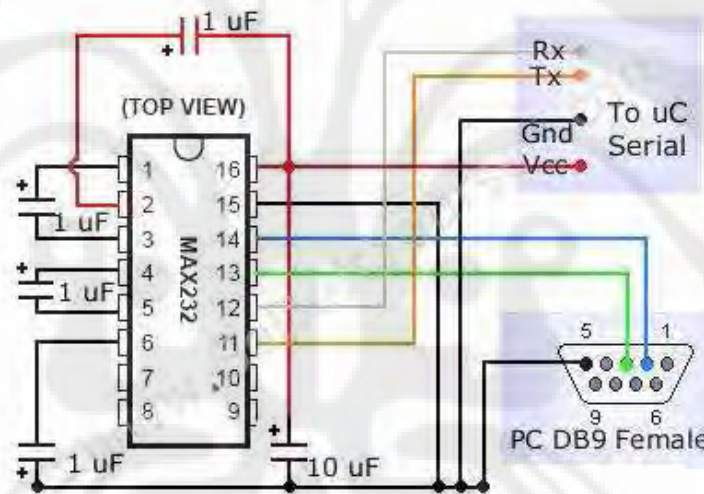
Sistem yang ketiga adalah penggabungan sifat dan atribut dari sistem RFID pasif dan aktif. Sistem RFID semipasif ini mempunyai arsitektur yang cukup unik karena menggunakan baterai sendiri untuk sumber power dalam memproses data internal sedangkan untuk meradiasikan data kembali ke *reader* menggunakan power dari sinyal yang diterima dari *reader*.

2.6 MAX232 SERIAL LEVEL CONVERTER

Ketika kita berkomunikasi dengan berbagai jenis *microprocessor*, kita perlu mengkonversi level RS232 ke level yang lebih bawah, biasanya 3.3 atau 5.0 Volts. Komunikasi serial RS-232 (V.24) bekerja pada level voltase -15V to +15V untuk *high* dan *low*. Pada sisi lain *TTL logic* beroperasi antara 0V dan +5V, oleh karenanya kita harus menggunakan MAX232 *Serial level converter* dengan konfigurasi pin pada DB9 seperti yang terlihat pada Gambar 2.10. Walaupun level sinyal RS-232 terlalu tinggi untuk *TTL electronics*, dan voltasenegative dari RS-232 tidak dapat di tangani oleh computer logic. Untuk mendapatkan data serial dari interface RS-232 level voltasenya harus dikurangi. Juga, level high dan low dari voltasenya harus di balik. Converter level ini menggunakan IC Max232 dan lima kapasitor seperti yang terlihat pada Gambar 2.11.



Gambar 2.10 Konfigurasi Pin pada DB9[16]



Gambar 2.11. Rangkaian Skematik untuk IC MAX232[16]

Pada komunikasi serial dengan *port* serial RS232, terdapat 3 pin yang memegang peranan penting, yaitu TxD, RxD, dan *Ground*. Data dikirimkan melalui pin TxD dan diterima pada pin RxD.

2.7 Perangkat Lunak

2.7.1 Basis Data

Basis data merupakan kumpulan data yang saling berhubungan. Hubungan antar data dapat ditunjukkan dengan adanya field/kolom kunci dari tiap file/tabel yang ada. Dalam satu file atau table terdapat record-record yang sejenis, sama besar, sama bentuk, yang merupakan satu kumpulan entitas yang seragam. Satu record (umumnya digambarkan sebagai baris data) terdiri dari field yang saling berhubungan menunjukkan bahwa field tersebut dalam satu pengertian yang lengkap dan disimpan dalam satu record[17].

Basis data bisa dikatakan sebagai suatu kumpulan dari data yang tersimpan dan diatur atau diorganisasikan sehingga data tersebut bisa diambil atau dicari dengan mudah dan efisien. Sebagai contoh sederhana dari basis data adalah buku telepon yang mungkin sering Anda lihat. Dengan adanya komputer maka informasi yang ada di dalam basis data akan sangat mudah untuk di-update dan sangat cepat untuk dicari. *Software* atau aplikasi yang bertugas untuk mengatur, menyimpan, memodifikasi data disebut dengan *software* basis data *engine* dan lebih resminya disebut dengan DBMS (*Data Base* Management Sistem). Ada beberapa aplikasi DBMS ini mulai yang berjalan di komputer personal (PC) sampai ke komputer skala mainframe. Diantaranya adalah MS.Access, SQL, oracle, dll [18].

2.7.2 Delphi 7[19]

Pada perancangan RFID sistem manajemen perpustakaan, aplikasi perangkat lunak akan dihubungkan dengan *modul reader* menggunakan suatu program yang dikembangkan dengan menggunakan Delphi 7.

Delphi 7 adalah salah satu bahasa pemrograman berbasis visual yang digunakan untuk membuat program aplikasi pada komputer (seperti Visual basic). Bahasa pemrograman yang digunakan oleh Delphi sebenarnya merupakan turunan

dari bahasa pemrograman pascal, yang dahulu pada Delphi dikenal sebagai objek pascal.

Pemrograman visual pada Delphi :

- 1) Merancang antarmuka (form dan komponen pendukungnya) secara visual (dapat dilihat)
- 2) Menuliskan kode untuk melakukan tindakan tertentu
- 3) Mengkompilasi kode Pascal dan form ke dalam bentuk file yang dapat dieksekusi

2.7.2.1 IDE (Integrated Development Environment) Delphi

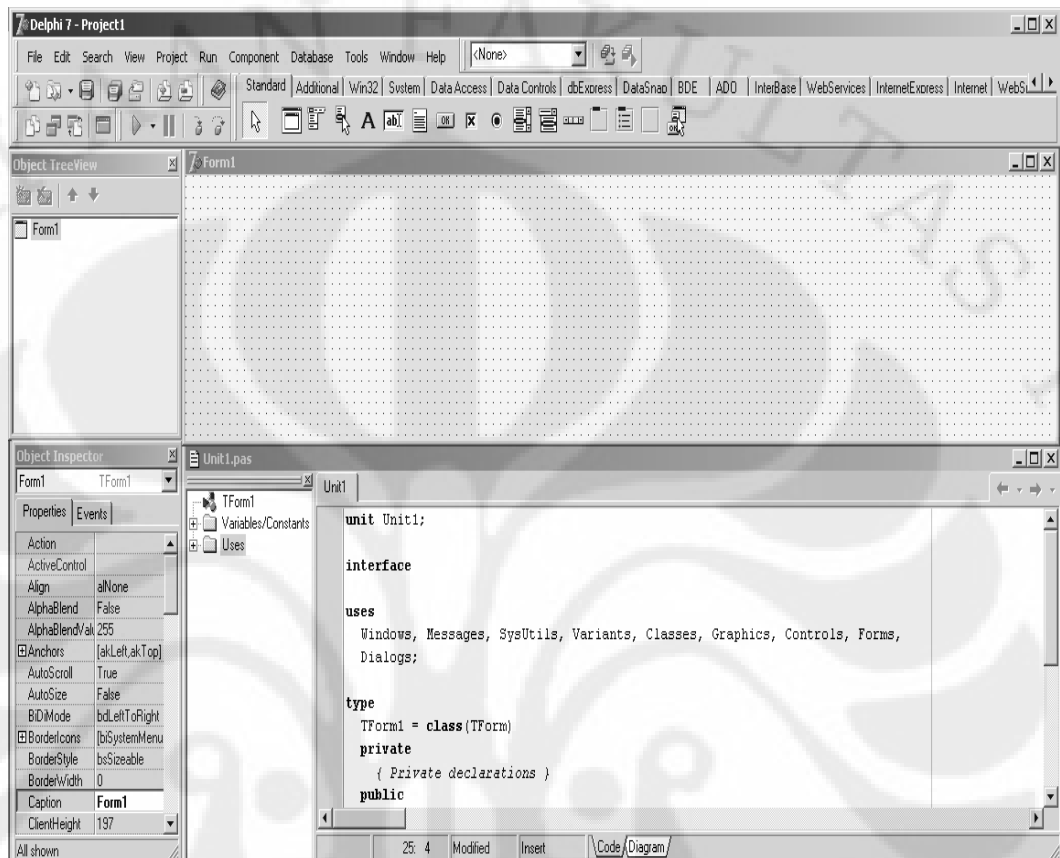
IDE adalah lingkungan dimana semua tool yang diperlukan untuk merancang, menjalankan, dan menguji sebuah aplikasi disajikan dan terhubung dengan baik sehingga memudahkan pengembangan program. IDE Delphi dibagi menjadi 8 bagian utama yaitu *main menu*, *toolbar/speedbar*, *component palette*, *form designer*, *code editor*, *code explorer*, *object inspector*, *object tree view*. Gambar 2.12 merupakan tampilan dari IDE Delphi 7.

1. Main Menu

Semua perintah yang diperlukan selama merancang dan membangun program aplikasi tersedia dalam menu ini. Jadi segala sesuatu yang berhubungan dengan IDE Delphi dapat dilakukan dari main menu.

2. Toolbar / Speedbar

Toolbar adalah sekumpulan tombol yang tidak lain adalah pengganti beberapa item menu yang sering digunakan. Biasanya yang tersedia pada toolbar adalah perintah-perintah (item menu) yang sering digunakan dalam proses pembuatan program aplikasi.



Gambar 2.12 IDE Delphi 7

3. Component Palette

Component palette adalah tool yang berupa kumpulan tab (*page control*), dimana setiap tab (*page control*) memuat berbagai tombol komponen (VCL / Visual Component Library) yang dapat diletakkan pada form dan sebagai interface program aplikasi. Tab tersebut diantaranya adalah Standard, Additional, Win32, dan lain-lain.

4. Form Designer

Form designer merupakan tempat untuk merancang program aplikasi. Perancangan form dilakukan dengan meletakkan komponen-komponen yang diambil dari component palette.

5. Object Inspector

Object inspector digunakan untuk mengubah properti atau karakteristik dari suatu komponen. Terdiri dari 2 tab yaitu :

- Properties

Digunakan untuk menentukan seting suatu objek. Satu objek memiliki beberapa properti yang dapat diatur langsung dari object inspector maupun melalui kode program. Seting ini mempengaruhi cara kerja objek tersebut saat aplikasi dijalankan.

- Event

Merupakan bagian yang dapat diisi dengan kode program tertentu yang berfungsi untuk menangani event-event (berupa sebuah procedure) yang dapat direspon oleh sebuah komponen. Event adalah peristiwa atau kejadian yang diterima oleh suatu objek, misal : klik, drag, dan lain-lain. Event yang diterima objek akan memicu Delphi menjalankan kode program yang ada didalamnya. Misalnya ingin sesuatu dikerjakan pada saat form ditutup, maka untuk menyatakan tindakan tersebut (berupa sebuah procedure) menggunakan OnClose.

6. Object Tree View

Object tree view berisi daftar komponen yang sudah diletakkan di form designer.

7. Code Editor

Code editor merupakan tempat untuk menuliskan kode program menggunakan bahasa object Pascal. Disini tidak perlu dituliskan seluruh kode sumber karena Delphi telah menyediakan kerangka penulisan sebuah program.

8. Code Explorer

Digunakan untuk memudahkan berpindah antar file unit di dalam jendela code editor. Code explorer berisi daftar yang menampilkan semua tipe, class, properti ,

Universitas Indonesia

method, variabel global, rutin global yang telah didefinisikan di dalam unit. Saat memilih sebuah item dalam code explorer, kursor akan berpindah menuju implementasi dari item yang dipilih di dalam code editor.

2.7.2.2 Tipe Data Pada Delphi

Pada umumnya, sebuah program komputer akan membutuhkan informasi yang dibutuhkan dari pengguna ketika digunakan. Informasi ini disebut dengan data. Delphi 7 mengenal beberapa tipe data, yaitu :

- 1) *Integer*, adalah tipe data untuk bilangan bulat;
- 2) *String*, adalah tipe data untuk teks (huruf, angka, atau tanda baca);
- 3) *Single*, adalah tipe data untuk pecahan;
- 4) *Currency*, adalah tipe data untuk mata uang;
- 5) *Boolean*, adalah tipe data yang bernilai *True* atau *False*;
- 6) *Date*, adalah tipe data untuk tanggal;
- 7) *Time*, adalah tipe data untuk jam;

2.7.2.3 Macam-macam Komponen di Delphi

Komponen yang disediakan oleh Delphi terdiri dari berbagai macam dapat digunakan dalam program. Komponen-komponen tersebut diletakkan pada komponen palet yang terletak di bawah menu pull down. Komponen – komponen yang berhubungan dengan sistem identifikasi *tag* yaitu *Comport*, *ADOConnection*, *ADOTable*, *DBGrid*, dan *DataSource*. Selain itu sistem juga menerapkan fungsi pengiriman email ke pelanggan perpustakaan setelah identifikasi *tag* dan pengecekan basis data buku dijalankan. Ada beberapa komponen Delphi yang dapat diaplikasikan untuk menerima dan mengirim email. Salah satunya dengan menggunakan library dan komponen sederhana dari Indy9. Berikut ini adalah pembahasannya :

1) *Comport*

Comport merupakan komponen yang berfungsi menerima data serial melalui port serial komputer. Komponen bernama *Comport* ini terdapat pada *Comport Library* versi 3.10 disingkat *CportLib* dan belum ter-install saat meng-install Delphi, jadi

harus di-install terpisah. *Comport* Library versi 3.10 ini mendukung Delphi versi 7. *Comport* memberikan kemudahan berkomunikasi serial dengan perlengkapan luar menggunakan koneksi RS232, seperti *RFID reader*, modem, *bar code reader*, dan lainnya. *Property* milik *Comport* lebih mudah diatur melalui *window Comm settings*, yaitu dengan men-double klik komponen *Comport* tersebut. *Property Port, Baud rate, Data bits, Stop bits, Parity, dan Flow control* dapat diatur.

2) ADOConnection

ADOConnection digunakan untuk mengkoneksikan basis data yang digunakan sistem otomasi peminjaman dan pengembalian buku ini dengan program utama. ActiveX Data Objects (ADO) merupakan suatu *driver* yang menyediakan informasi yang menghadirkan sumber data didalam sebuah dataset. ADOConnection digunakan untuk berkoneksi dengan *ADO data store*. Kelebihan ADOConnection bahwa koneksinya tidak membutuhkan *Borland Database Engine (BDE)*, artinya walaupun BDE bersama Delphi tidak ter-install di komputer, ADOConnection tetap dapat melakukan koneksi karena biasanya sudah ter-install bersama OS Windows.

3) ADOTable

ADOTable merupakan komponen yang mengambil suatu tabel dalam file basis data untuk ditempatkan sebagai suatu tabel pada Delphi yang siap pakai. Syarat pengambilannya adalah basis data harus dikoneksikan dahulu. Untuk berkoneksi dengan basis data menggunakan ADOConnection, pengambilan tabelnya menggunakan ADOTable. Cara pemakaian ADOTable, yaitu pada *property*-nya yang bernama Connection dipilih sebagai nama dari ADOConnection yang sudah ada. Pada *property* bernama TableName dipilih sebagai nama tabel dalam basis data yang ingin diletakkan pada Delphi.

4) DBGrid dan DataSource

DBGrid merupakan komponen yang menampilkan data dalam bentuk tabel. DBGrid harus dihubungkan dahulu dengan ADOTable. Komponen yang dapat menghubungkan DBGrid dengan ADOTable adalah komponen yang bernama DataSource. *Property* DataSource yang bernama DataSet dipilih sebagai nama dari ADOTable yang sudah ada. *Property* DBGrid yang bernama DataSource dipilih

sebagai nama dari komponen DataSource yang sudah ada. Pada komponen DBGrid di-double klik, kemudian pada window yang baru saja muncul diklik kanan dan *Add all fields* dipilih. Maka semua data pada ADOTable akan ditampilkan pada DBGrid.

5) IdSMTP

Komponen IdSMTP berfungsi untuk menjalankan fungsi klien Simple Mail Transfer Protocol dengan demikian user dapat mengirim email ke alamat yang dituju dari aplikasi mail klien yang dibuat dengan delphi atau yang biasa digunakan seperti microsoft outlook. Sebelumnya user harus mempunyai akun *webmail* (yahoo, gmail, atau email kantor), untuk kemudian user harus mengaktifasi fungsi SMTP di akun webmailnya. Selain itu pada komponen IdSMTP sendiri, user harus mengisi port SMTP, alamat SMTP webmailnya, username beserta password webmailnya, sebagai contoh gmail menggunakan port 465 untuk SMTP dan alamat "smtp.gmail.com".

6) IdPOP3

IdPOP3 merupakan komponen yang berfungsi untuk menjalankan fungsi klien Post Office Protocol version 3 dengan demikian user dapat menerima email dari suatu mail-server untuk kemudian dibaca di aplikasi komputernya user (sebagai contoh Microsoft Outlook atau aplikasi mail Delphi). Sebagaimana fungsi SMTP, POP3 suatu akun juga harus diaktivasi disisi mail server serta mengatur port POP3 dan alamat POP3-nya di aplikasi mail klien user. Sebagai contoh gmail menggunakan port 995 dan alamat "pop.gmail.com".

7) IdMessage

Komponen ini berfungsi untuk menterjemahkan pesan yang kita tulis di program delphi untuk kemudian dienkapsulasikan dalam pesan internet. Fungsi ini dijalankan bersama dengan IdSMTP dan IdPOP3 agar pesan bisa terbaca.

8) IdSSLIOHandlerSocket

Komponen ini berfungsi untuk menjalankan handler I/O (input/output) berbasis soket agar bisa melakukan notifikasi dan verifikasi password, informasi untuk koneksi SSL (Secure Socket Layer), dengan demikian keamanan user dapat terjaga. IdSSLIOHandlerSocket harus didefinisikan pada setiap komponen IdSMTP atau IdPOP3 di atribut "On-Status"-nya. Supaya komponen ini bisa berfungsi, pada

project delphi harus dimasukkan library "libeay32.dll" dan "ssleay32.dll" agar komponen IdSSLIOHandlerSocket bisa mengunduh *library socketnya*.

2.7.2.4 Management Project

Delphi tidak hanya menyimpan *file* kode dengan perluasan PAS, tetapi karena pada Delphi juga terdapat *form* beserta parameternya, maka ada beberapa berkas yang akan disimpan. Untuk memudahkan, sebuah program disebut dengan sebuah *project*. *Project* tersebut akan berisi *form*, *source code* untuk *form*, dan *source code* untuk *project*. Untuk sebuah *form* akan diberi sebuah unit, yang akan berisi kode-kode program untuk merekayasa form tersebut, terutama untuk *events* yang dimiliki oleh *form* tersebut.

Berikut ini berkas-berkas yang dihasilkan dan disimpan oleh Delphi:

- *Project files* (*.DPR), yaitu *source code* sebuah project pada Delphi. Setiap kali aplikasi pada Delphi dirancang, maka akan terdapat satu *project*.
- *Form files* (*.DFM), yaitu *text files* atau *binary files* yang berisi informasi tentang sebuah *form*.
- *Unit files* (*.PAS), yaitu *source code* untuk unit. Setiap kali sebuah *form* dirancang maka akan diberikan sebuah berkas unit—nama berkas keduanya sama, hanya perluasannya berbeda. Tetapi berkas unit dapat berdiri sendiri seperti halnya pada Turbo Pascal tanpa korespondensi sebuah *form*.

Ada beberapa berkas lain yang ada pada sebuah *project*, semacam (*.RES), berkas konfigurasi (*.CFG), berkas untuk *options* (*.DOF), hasil kompilasi unit (*.DCU), dan lain-lain. Bila sebuah *project* akan disalin antar media, maka semua berkas yang ada pada *project directory* harus disertakan.

2.7.2.5 Komunikasi Serial Pada Delphi

Komunikasi serial pada konsepnya adalah metode transmisi data per bit dalam satu waktu melalui sebuah jalur transmisi baik dengan kabel maupun *wireless*. Sistem komunikasi ini lebih lambat daripada komunikasi paralel yang memungkinkan

pengiriman semua bit data dalam satu waktu. Akan tetapi, panjang kabel yang digunakan mampu mencapai 100 meter.

Ada beberapa karakteristik yang penting pada komunikasi serial, yaitu :

a) *Baud Rate*

Merupakan sistem perhitungan untuk komunikasi, *baud rate* mengidentifikasi berapa bit data yang dikirimkan setiap detik. Pada *clock cycle*, *baud rate* menunjukkan frekuensi yang digunakan oleh *clock* tersebut.

b) *Data bits*

Menunjukkan perhitungan jumlah dari data bit yang sedang ditransmisikan, pengiriman standarnya yaitu 5, 7 atau 8 bit, itu tergantung pada data yang ditransmisikan.

c) *Stop bit*

Digunakan untuk mengakhiri komunikasi untuk satu paket, selain itu juga digunakan untuk menangani error pada kecepatan *clock*.

d) *Parity*

Merupakan bit tambahan yang akan mendeteksi adanya kesalahan pada komunikasi serial.

BAB 3

PERANCANGAN PROTOTYPE

3.1 Konsep Prototype

Sistem otomasi pada perpustakaan ini dirancang agar anggota perpustakaan tidak perlu berinteraksi dengan komputer dan melakukan beberapa aktifitas pada computer itu sendiri sebelum dapat melakukan scan ID *member* dan scan *tag* buku menggunakan *reader* RFID. Anggota dapat langsung melakukan scan ID *member* dan scan *tag* buku pada 1 *reader* yang telah disediakan, sistem akan secara otomatis merekam transaksi peminjaman tersebut. Sistem ini terdiri atas *reader* untuk men-scan ID *member* dan sekaligus untuk mendeteksi buku/item yang dipinjam dan juga *reader* yang terletak di pintu keluar untuk deteksi anti pencurian. Semua *reader* tersebut terhubung dengan komputer pusat. Seluruh informasi basis data juga sudah terintegrasi didalam komputer tersebut. Komputer pusat ini harus terhubung dengan internet agar dapat mengirim email notifikasi masa pinjam buku.

Pada perancangan sistem otomasi untuk proses peminjaman dan pengembalian buku pada perpustakaan ini digunakan 2 kategori *tag*. *Tag* pertama berfungsi sebagai identitas anggota perpustakaan karena yang diizinkan untuk melakukan peminjaman buku hanyalah anggota perpustakaan. *Tag* yang berperan sebagai ID card anggota perpustakaan ini berisikan data-data mengenai anggota tersebut meliputi nama, pekerjaan, alamat rumah, nomor telepon, dan email. *Tag* kedua digunakan sebagai identitas buku dimana *tag* ini ditempel pada tiap-tiap buku. *Tag* buku ini memuat informasi tentang buku meliputi judul buku, kategori buku, pengarang dan penerbit. Sesuai dengan namanya Radio Frequency Identification, maka proses identifikasi dilakukan dengan menggunakan frekuensi radio yang mendeteksi ada tidaknya suatu informasi yang dikenali oleh sistem. Perangkat teknologi RFID ini terdiri dari 2 komponen dasar yaitu *tag* dan *reader*. *Reader* bertugas membaca informasi dari *tag* yang berada pada jangkauan baca dari *reader* tersebut. Sebagai antarmuka antar perangkat keras dengan basis data maka dibangun perangkat lunak menggunakan

bahasa pemrograman Delphi 7. Komputer client terhubung dengan komputer server melalui LAN. Lalu informasi dari *reader* yang diterima oleh komputer client akan disimpan dalam sebuah basis data yang ada pada computer server.

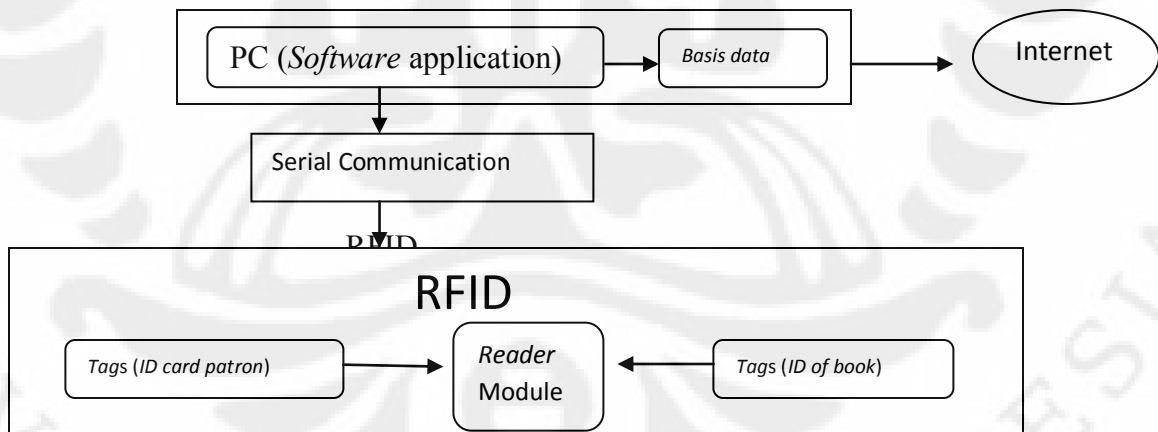
3.2 Gambaran Teknis Sistem Otomasi Peminjaman dan Pengembalian Buku Pada Perpustakaan Menggunakan Teknologi RFID

Secara teknis, sistem otomasi peminjaman dan pengembalian buku pada perpustakaan ini dapat digambarkan melalui arsitektur, dan *layout* dari perpustakaan berbasis teknologi RFID.

3.2.1 Arsitektur Sistem Otomasi Peminjaman dan Pengembalian Buku Pada Perpustakaan Menggunakan Teknologi RFID

Secara umum, arsitektur sistem otomasi peminjaman dan pengembalian buku pada perpustakaan ini terdiri dari beberapa bagian. Bagian pertama yaitu RFID yang terdiri dari *reader* dan *tag*. Bagian kedua yaitu *middleware* atau dalam sistem ini berupa komputer yang didalamnya berisi program dan basis data.

Arsitektur sistem otomasi perpustakaan ini terlihat pada gambar 3.1 berikut:

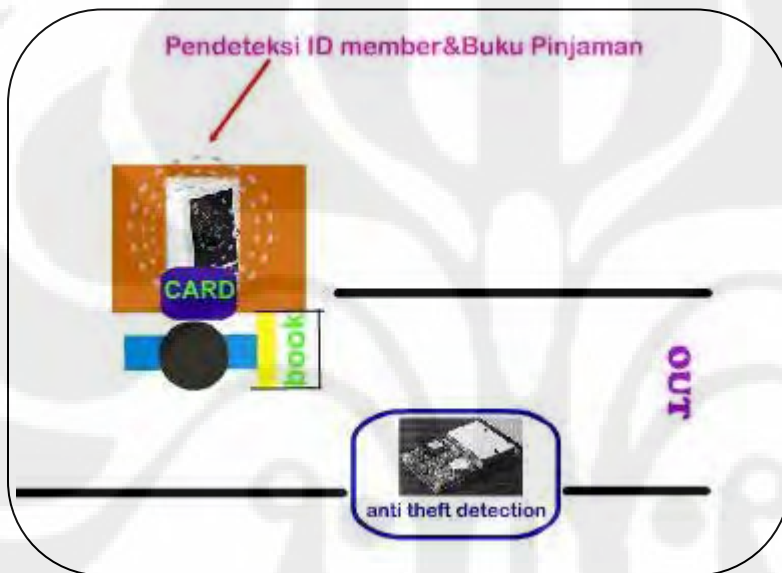


Komunikasi data antara RFID dengan program utama pada PC berlangsung hanya satu arah saja, dari RFID ke PC melalui *port* serial RS232.

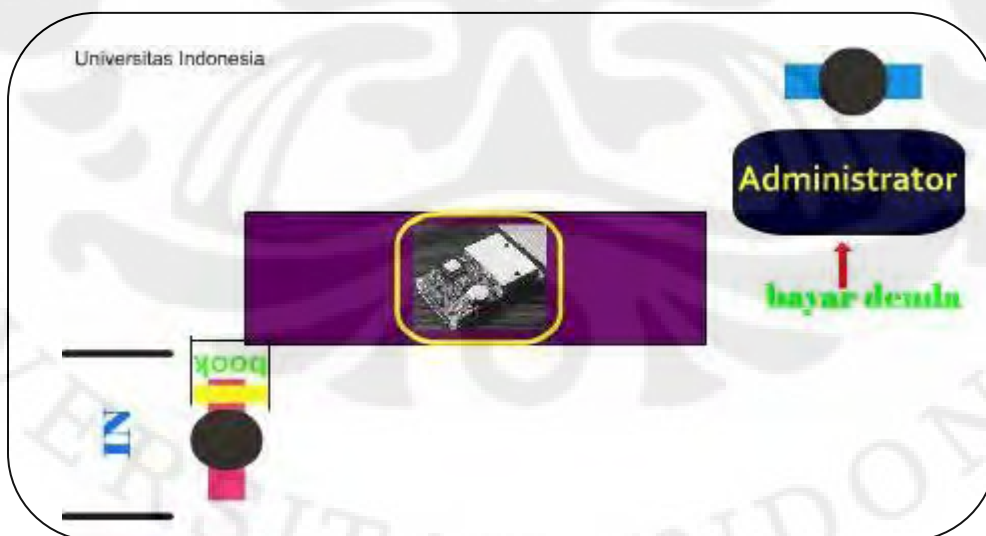
3.2.2 Layout Sistem Otomasi Peminjaman dan Pengembalian Buku Pada Perpustakaan Menggunakan Teknologi RFID

Gambar 3.2 dibawah ini adalah *Layout* Sistem otomasi peminjaman dan pengembalian buku pada perpustakaan :

(a)



(b)



Gambar 3.2(a) *Layout* Sistem check-out.(b) *Layout* Sistem check-in.

3.3 Alur Kerja Sistem Otomasi Peminjaman dan Pengembalian Buku Pada Perpustakaan

Sistem otomasi peminjaman dan pengembalian buku pada perpustakaan ini memiliki beberapa tahapan kerja. Alur kerja dibagi menjadi 3 bagian, yaitu :

- Sub peminjaman
- Sub pengembalian, dan
- Sub pengiriman email notifikasi masa pinjam.

3.3.1 Sub Peminjaman

Untuk proses peminjaman, alur kerja dari sistem otomasi menggunakan RFID ini adalah :

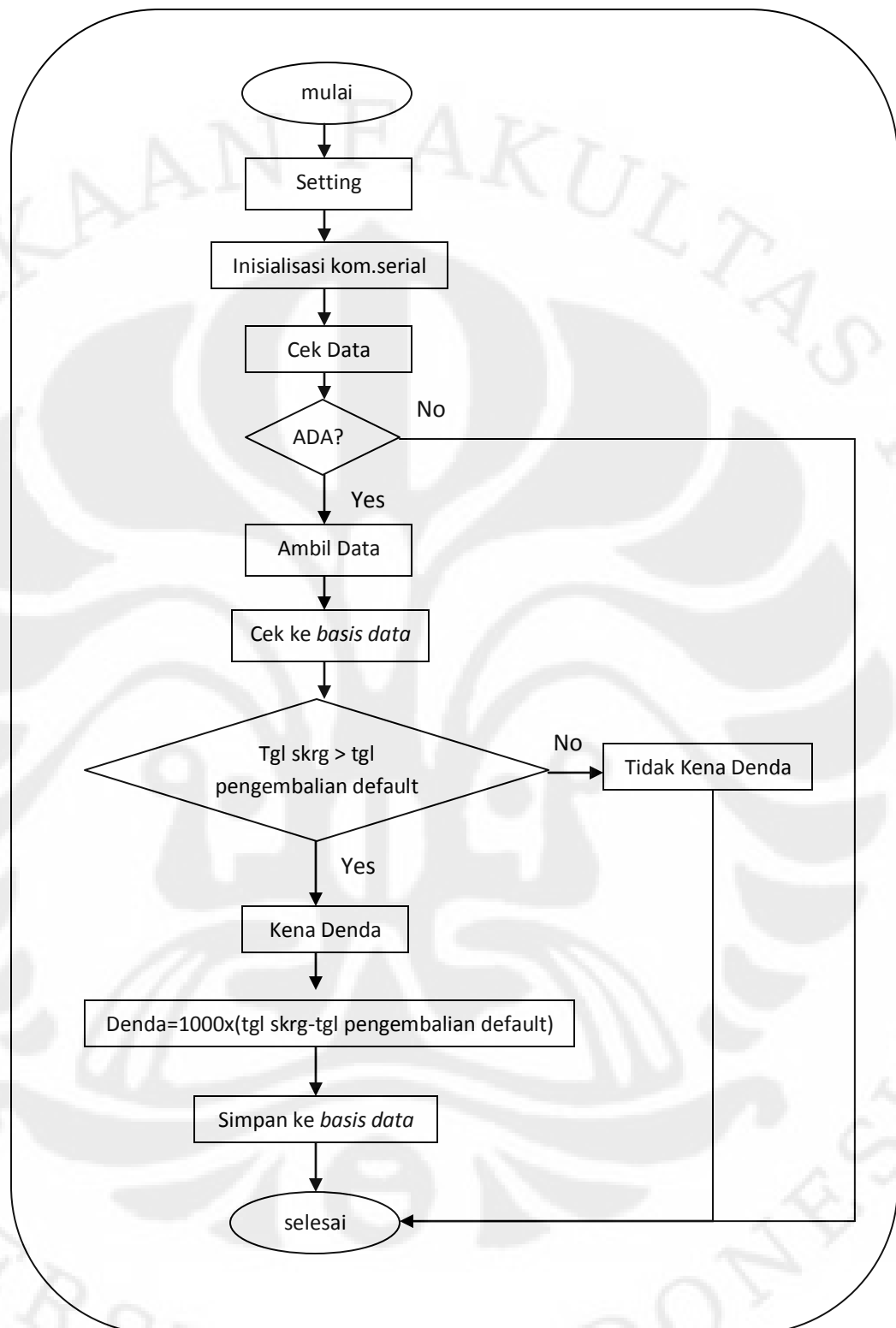
1. Tidak semua pengunjung perpustakaan dapat melakukan peminjaman buku. Hak peminjaman atas buku&inventaris perpustakaan diberikan hanya bagi orang-orang yang teridentifikasi *member* oleh sistem. Oleh karena itu, pertama kali, *reader* akan meminta scan *tag member* terlebih dahulu sebelum melakukan scan terhadap item-item yang akan dipinjam.
2. Bila ID *member* tidak teridentifikasi maka *reader* yang bertugas untuk scan buku tidak akan aktif dan status *tag* buku 'LOCK'. Bila buku tetap dibawa keluar, pintu deteksi anti pencurian akan membunyikan alarm.
3. Bila *member* telah teridentifikasi, item/buku di-scan pada *reader* terletak pada meja.
4. Peminjam berjalan ke pintu keluar dan mengambil item/buku yang telah melalui tahap pembacaan oleh *reader*.
5. Peminjam hanya diperbolehkan maksimal meminjam 3 item buku/CD.

6. Ditetapkan waktu 15 detik untuk pembacaan *tag* buku. Setelah waktu habis, scan *tag* buku selesai.
7. Simpan semua data *member*, data buku/item yang dipinjam dan tanggal peminjaman pada tabel basis data peminjaman. Tanggal pengembalian default akan ditentukan secara otomatis oleh sistem yaitu 1 minggu dari tanggal peminjaman.
8. Proses peminjaman selesai dan dilanjutkan untuk *member* berikutnya dengan melakukan scan *tag* ID *member* terlebih dahulu.

3.3.2 Sub Pengembalian

Pada jalur pintu masuk disediakan ban berjalan untuk proses pengembalian buku. Dalam hal ini, sistem tidak meminta untuk scan *tag* ID *member*. Hanya dengan meletakkan buku pada ban berjalan, *reader* akan mendeteksi *tag* buku.

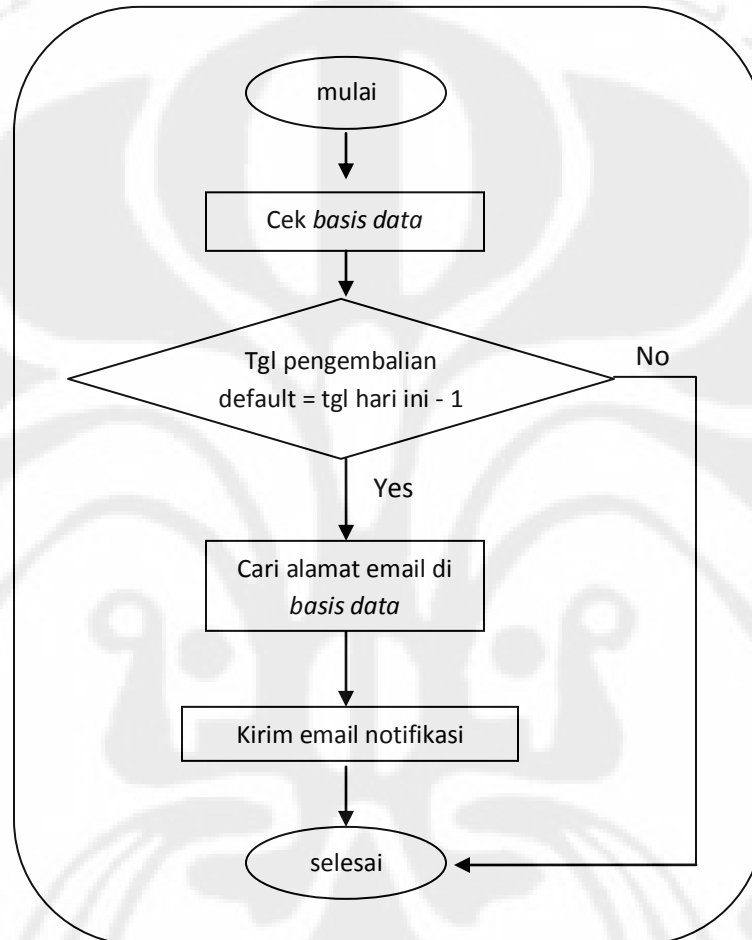
Sistem akan mencari data buku tersebut pada basis data peminjaman. Bila tanggal saat buku dikembalikan telah melewati tanggal pengembalian default maka akan terkena denda. Denda dibayarkan pada admin yang berada pada ujung meja ban berjalan. Diagram alir dari sub peminjaman ditunjukkan oleh Gambar 3.3.



Gambar 3.3 Diagram alir Sub pengembalian

3.3.3 Sub Pengiriman Email Notifikasi Masa Pinjam

Untuk sub pengiriman email notifikasi ini, diagram alirnya adalah seperti yang ditunjukkan pada Gambar 3.4.



Gambar 3.4 Diagram alir pengiriman email notifikasi

Dalam hal ini, komputer server harus terhubung ke jaringan dengan koneksi internet tanpa proxy (direct) karena SMTP gmail yang digunakan sebagai account `_adminperpusftui` bekerja di jaringan yang mengizinkan koneksi internet public tanpa proxy (direct).

Pada perangkat lunak terdapat beberapa pengaturan terhadap komponen-komponen email, antara lain pengaturan port pada IdSMTP untuk domain email tertentu (dalam hal ini gmail) untuk dapat melakukan proses pengiriman email.

Universitas Indonesia

Fungsi SMTP pada account gmail `_adminperpusftui` juga harus diaktifkan. Script pada program untuk menjalankan flowchart tersebut dapat dilihat pada Tabel 3.2.

3.4 Interfacing perangkat keras dengan perangkat lunak

Sistem otomasi ini memiliki perangkat lunak dan perangkat keras dimana perangkat tersebut saling berkomunikasi. Komunikasi antara perangkat lunak dan perangkat keras dapat menggunakan dua buah komunikasi, yaitu komunikasi serial dan komunikasi paralel. Namun pada sistem ini digunakan komunikasi serial sebagai alat berkomunikasi antara perangkat keras dan perangkat lunaknya.

Pada Delphi sudah ada komponen yang berfungsi untuk melakukan komunikasi dengan perangkat keras yaitu komponen *comport*. Komponen *comport* ini dapat disetting sesuai dengan karakteristik dari perangkat keras yang digunakan.

3.5 Penggunaan Perangkat Lunak (Program Utama) pada Sistem Otomasi Peminjaman dan Pengembalian Buku Perpustakaan

Untuk sistem otomasi peminjaman dan pengembalian buku pada perpustakaan ini digunakan perangkat lunak Delphi. Perangkat lunak ini dirancang untuk melayani 2 transaksi yang ingin dilakukan oleh anggota perpustakaan yaitu transaksi peminjaman buku dan transaksi pengembalian buku. Delphi sendiri merupakan bahasa pemrograman tingkat tinggi yang digunakan pada pemrograman berorientasi objek. Perangkat lunak dalam sistem otomasi peminjaman dan pengembalian buku pada perpustakaan berfungsi untuk menjalankan algoritma sistem dan juga menghubungkan basis data dengan sistem yang ada sehingga sistem berjalan dengan baik. Untuk menghubungkan basis data dengan Delphi maka digunakan komponen yang terdapat di dalam Delphi yaitu ADO (Active Data Object).

Perangkat lunak ini juga berfungsi untuk menghubungkan (interfacing) antara komputer dengan perangkat keras yang akan dihubungkan nantinya seperti mikrokontroler dan RFID. Dalam berkomunikasi dengan perangkat keras, Delphi menggunakan komponen yang bernama *Comport*.

Tampilan awal aplikasi yang dibuat adalah seperti tampak pada Gambar 3.5 dan 3.6 dibawah ini :

Gambar 3.5 Tampilan Awal Aplikasi untuk Sub Peminjaman Buku

Gambar 3.6 Tampilan Awal untuk Aplikasi Sub Pengembalian Buku

3.6 Penggunaan Basis data untuk Sistem Peminjaman dan Pengembalian Buku pada Perpustakaan

Pada perancangan basis data digunakan basis data MS Access. MS Access sendiri merupakan basis data yang dibuat menggunakan program Microsoft Office. Di dalam basis data MS Access nanti akan diisi dengan data-data yang dibutuhkan seperti nomor *tag* ID card anggota perpustakaan dan *tag* ID buku, data lengkap

anggota dan buku, juga data buku yang keluar dan masuk. Kemudian basis data MS Access akan diakses oleh program utama (dalam hal ini Delphi) menggunakan komponen yang terdapat di Delphi sendiri yaitu ADO (Active Data Object). Dengan adanya ADO pada Delphi, distribusi program hanya membutuhkan file exe dan basis data-nya (*.mdb) saja.

Pada perancangan basis data untuk sistem otomasi peminjaman dan pengembalian buku pada perpustakaan ini dibuat 4 jenis tabel. Tabel pertama merupakan tabel yang berisi data-data mengenai *member* perpustakaan. Tabel ini terdiri dari beberapa *field* yaitu nama, ID *member*, NPM/NIP, jurusan, alamat rumah, no.telp, dan alamat email. Tabel kedua digunakan untuk menyimpan informasi mengenai identitas buku. Tabel ini terdiri dari *field* ID buku, judul buku, pengarang, dan penerbit.

Tabel ketiga dan Tabel keempat adalah tabel yang digunakan untuk menyimpan data transaksi peminjaman dan pengembalian buku. Tabel ini akan terus ter-update setiap kali ada transaksi yang terjadi baik peminjaman ataupun pengembalian. Kedua tabel ini terdiri dari tabel yang sama yaitu nama peminjam, ID peminjam, email, judul buku, ID buku, tanggal peminjaman, dan tanggal pengembalian default. Hanya saja pada tabel transaksi pengembalian terdapat beberapa tambahan *field* yaitu tanggal pengembalian real dan total denda.

Agar basis data yang dibuat pada MS Access dapat terkoneksi dengan Delphi, maka harus dituliskan script seperti pada Tabel 3.1 di bawah ini :

Tabel 3.1 Script pada Delphi untuk mengakses basis data MS Access

```

procedure TForm1.connectClick(Sender: TObject);
begin
if OpenFileDialog1.Execute then
begin
    TableRFID1.ConnectionString :=
    'Provider=Microsoft.Jet.OLEDB.4.0;Data Source=' +
    opendirialog1.FileName + ';Persist Security Info=False';
    TableRFID1.TableName := 'Dbase Member';

    TableRFID2.ConnectionString :=
    'Provider=Microsoft.Jet.OLEDB.4.0;Data Source=' +
    opendirialog1.FileName + ';Persist Security Info=False';

```

```

TableRFID2.TableName := 'Dbase Buku';

TableRFID3.ConnectionString :=
'Provider=Microsoft.Jet.OLEDB.4.0;Data Source=' +
opendialog1.FileName + ';Persist Security Info=False';
TableRFID3.TableName := 'Tabel Transaksi Peminjaman';

TableRFID4.ConnectionString :=
'Provider=Microsoft.Jet.OLEDB.4.0;Data Source=' +
opendialog1.FileName + ';Persist Security Info=False';
TableRFID4.TableName := 'Tabel Transaksi Pengembalian';

end;

TableRFID1.Active := true;
DataSource1.DataSet := TableRFID1;

TableRFID2.Active := true;
DataSource2.DataSet := TableRFID2;

TableRFID3.Active := true;
DataSource3.DataSet := TableRFID3;
form2.DBGrid1.DataSource := DataSource3;

TableRFID4.Active := true;
DataSource4.DataSet := TableRFID4;
form3.DBGrid2.DataSource := DataSource4;
C1.ShowSetupDialog;
C1.Connected:=true;
C1.FlowControl.FlowControl:=fcHardware;

end;

```

Selain itu, basis data juga digunakan sebagai referensi untuk melakukan pengiriman email notifikasi masa pinjam ini. Email ini merupakan komponen control tambahan yang bertujuan untuk mengontrol buku yang berada di luar perpustakaan. Email ini berupa pemberitahuan kepada semua peminjam bahwa buku yang dipinjam harus segera dikembalikan. Berikut alur kerjanya :

1. Setiap hari ketika administrator menyalakan komputer, sistem akan mengecek basis data peminjaman.

2. Bila ditemukan buku/item yang masa pinjamnya tinggal 1 hari lagi, sistem akan secara otomatis mengirimkan email notifikasi kepada semua peminjam untuk segera mengembalikan buku/item yang mereka pinjam.

Untuk dapat melakukan pengiriman email maka pada Delphi perlu ditulis script seperti pada Tabel 3.2 dibawah ini :

Tabel 3.2 Script pada Delphi untuk melakukan pengiriman email notifikasi masa pinjam

```

procedure TForm1.checkstatusbukuClick(Sender: TObject);
var
  i, warned : integer;
  nomor, nama, buku : string;
  adadb4      : Boolean;
  optbdb4     : TlocateOptions;
begin
  for i:=1 to TableRFID3.RecordCount+1 do
  begin
    adadb4 := TableRFID3.Locate('No', i, optbdb4);
    if adadb4 then
    begin
      warned := TableRFID3.FieldValues['tgl pengembalian default'] -
      DateTimePicker1.Date;
      if warned = 1 then
      begin
        nomor:=TableRFID3.FieldValues['No'];
        nama:=TableRFID3.FieldValues['Nama Anggota'];
        buku:=TableRFID3.FieldValues['Judul Buku'];
        IdMessage1.From.Address := 'adminperpusftui@gmail.com';
        IdMessage1.Recipients.EmailAddresses :=

```

```
TableRFID3.FieldValues['email'];

    IdMessage1.Subject := 'Kepada '+nama+' Anggota Perpustakaan UI
yang terhormat';

    IdMessage1.Body.Text := 'Batas peminjaman buku '+buku+'
tinggal sehari lagi';

    IdSMTP1.Connect();

    if IdSMTP1.Connected then
    begin
        try
            IdSMTP1.Send(IdMessage1);

            finally

                MessageDlg(nomor+'. Pesan Terkirim ke '+nama+' bahwa
buku '+buku+' harus dikembalikan sehari
lagi!',mtConfirmation,[mbOK],0);

                IdSMTP1.Disconnect;

            end
        end
    else
        MessageDlg('Error in
Connection!',mtConfirmation,[mbOK],0);

        IdSMTP1.Disconnect;

        Sleep(1000);

    end
end;
end;

    MessageDlg('pengecekan buku selesai!',mtConfirmation,[mbOK],0);
end;
```

BAB 4 PENGUJIAN SISTEM DAN ANALISIS

Pengujian sistem bertujuan untuk mengetahui cara kerja perangkat dan menganalisa tingkat reliabilitas, kelemahan dan keterbatasan spesifikasi fungsi dari aplikasi yang telah dibuat. Pengujian terhadap sistem ini dilakukan dengan menggunakan sebuah Laptop pribadi yang memiliki spesifikasi sebagai berikut :

Prosesor : Intel Pentium Dual-Core, 2 GHz

RAM/HD : 2 Gb/250 Gb

Sistem Operasi : Microsoft Windows XP Professional (5.1, build 2600)

Pada skripsi ini dilakukan beberapa pengujian yang dibagi menjadi 3 bagian, bagian-bagian dari pengujian tersebut adalah sebagai berikut.

1. Pengujian pembacaan *tag* oleh *reader*, pengujian ini meliputi :
 - a) pengujian pembacaan *tag* sebelum dan sesudah terhubung aplikasi *software*
 - b) pengujian pembacaan *tag* oleh *reader* dengan berbagai jenis penghalang
2. Pengujian perangkat lunak berdasarkan spesifikasi fungsi, pengujian ini meliputi :
 - a) pengujian setting hardware
 - b) pengujian pengaturan scan *tag* buku pada sub peminjaman
 - c) pengujian pengiriman email notifikasi masa pinjam yang akan habis
 - d) pengujian penyimpanan data ke basis data
3. Pengujian ketahanan sistem dengan RFID *reader*

Pada implementasi, setingan komunikasi antara mikrokontroler dengan perangkat lunak harus disamakan terlebih dahulu. Dalam pengujian ini, setingan *port* komunikasi serial antara computer dan *reader* menggunakan Com4. Karena pada laptop tidak ada connector DB-9 maka digunakan 1 buah converter serial to *USB*. Converter serial to *USB* ini menggunakan *USB* to RS-232 Cable Model:HE800A. Penggunaan port *USB* untuk komunikasi serial pada computer yang berbeda tidak sama, untuk itu, sebelum dilakukan setingan pada perangkat lunak terlebih dahulu harus diketahui *port* yang digunakan. Hal ini biasa dilakukan dengan menggunakan komunikasi *hyperterminal* pada PC.

Komunikasi antara *reader* dan komputer hanya berlangsung ketika setting *Comport* pada perangkat lunak yang dibangun menggunakan Delphi 7 pada computer sesuai dengan port yang digunakan oleh *reader*.

4.1 Pengujian Pembacaan Tag oleh Reader

Pada perancangan sistem otomasi peminjaman dan pengembalian buku ini, hanya digunakan 1 *reader* yang melakukan pendeteksian terhadap *tag member* juga *tag* buku.

Reader yang digunakan adalah Starter Kit ID-12 dimana memiliki spesifikasi sebagai berikut :

- Berbasis RFID *reader* ID-12 dengan frekuensi kerja 125 kHz untuk kartu berformat EM4001 / sejenis.
- Mendukung format data ASCII (UART TTL / RS-232), Wiegand26, maupun Magnetic ABA Track2.
- Catu daya : 9 sampai 12 V DC.
- Tersedia jalur komunikasi serial UART RS-232 dengan konektor RJ11 dan kabel serial untuk menghubungkan modul ini ke COM port komputer.

- Dilengkapi dengan buzzer sebagai indicator baca, serta LED sebagai indicator tulis.

Pengujian pembacaan *tag* ini dilakukan pada dua kondisi yaitu sebelum terhubung dengan aplikasi *software* dan sesudah terhubung aplikasi *software*. Pengujian ini bertujuan untuk menguji interval waktu yang dibutuhkan antara satu pembacaan dengan pembacaan selanjutnya tanpa eror dalam pembacaan pada dua kondisi tersebut.

4.1.1 Pengujian Pembacaan *Tag* Sebelum dan Sesudah Terhubung Aplikasi *Software*

Pengujian ini dilakukan pada kondisi *reader* belum terhubung dengan komputer dimana aplikasi *software* dijalankan. Dari percobaan yang dilakukan, didapatkan hasil seperti yang terlihat pada Tabel 4.1.

Tabel 4.1 Hasil pengujian interval waktu pembacaan *tag* sebelum terhubung aplikasi *software*

| Interval waktu pembacaan | Tes 1 | Tes 2 | Tes 3 | Tes 4 | Tes 5 | Tes 6 | Tes 7 | Tes 8 | Tes 9 | Tes 10 |
|--------------------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|
| 1 sekon | Tidak terbaca | Tidak terbaca | Tidak terbaca | Tidak terbaca | Tidak terbaca | Tidak terbaca | Tidak terbaca | Tidak terbaca | Tidak terbaca | Tidak terbaca |
| 2 sekon | terbaca | terbaca | terbaca | terbaca | terbaca | terbaca | terbaca | terbaca | terbaca | terbaca |
| 3 sekon | terbaca | Terbaca | terbaca | terbaca | terbaca | terbaca | terbaca | terbaca | terbaca | terbaca |

Tabel 4.2 Hasil pengujian interval waktu pembacaan *tag* sesudah terhubung aplikasi *software*

| Interval waktu pembacaan | Tes 1 | Tes 2 | Tes 3 | Tes 4 | Tes 5 | Tes 6 | Tes 7 | Tes 8 | Tes 9 | Tes 10 |
|--------------------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|
| 1 sekon | Tidak terbaca | Tidak terbaca | Tidak terbaca | Tidak terbaca | Tidak terbaca | Tidak terbaca | Tidak terbaca | Tidak terbaca | Tidak terbaca | Tidak terbaca |
| 2 sekon | terbaca | terbaca | terbaca | terbaca | terbaca | terbaca | terbaca | terbaca | terbaca | Tidak terbaca |
| 3 sekon | terbaca | Terbaca | terbaca | terbaca | terbaca | terbaca | terbaca | terbaca | terbaca | terbaca |

Dari hasil pengujian yang dilakukan pada 2 kondisi tersebut, dapat disimpulkan bahwa pada interval waktu 2 detik, pembacaan *tag* sudah dapat berjalan dengan baik meski terdapat 1 data yang tidak terbaca pada tes 10 Tabel 4.2. Hal ini terjadi karena adanya kesalahan pada penyambungan komunikasi serial yang tiba-tiba terlepas antara *reader* dan komputer yang menggunakan kabel USB converter serial 232. Hal ini menunjukkan bahwa aplikasi sudah dapat berjalan dengan baik untuk menerima data pembacaan *tag* oleh *reader*.

4.1.2 Pengujian Pembacaan *Tag* Oleh *Reader* dengan Berbagai Jenis Penghalang

Pengujian ini dilakukan untuk mengetahui kemungkinan apa saja yang akan mengganggu kerja sistem dengan tidak terdeteksinya *tag* oleh *reader*. Hasil pengujian ditunjukkan oleh Tabel 4.3.

Tabel 4.3 Hasil Pengujian Pembacaan *Tag* dengan Berbagai Penghalang

| No | Jenis Penghalang | Percobaan | Terbaca |
|----|------------------|-----------|---------|
| 1 | Kaca | 1 | Terbaca |
| | | 2 | Terbaca |
| | | 3 | Terbaca |
| | | 4 | Terbaca |
| | | 5 | Terbaca |
| 2 | Plastik | 1 | Terbaca |
| | | 2 | Terbaca |
| | | 3 | Terbaca |
| | | 4 | Terbaca |
| | | 5 | Terbaca |
| 3 | Kertas | 1 | Terbaca |
| | | 2 | Terbaca |
| | | 3 | Terbaca |
| | | 4 | Terbaca |
| | | 5 | Terbaca |
| 4 | Aluminium | 1 | Tidak |
| | | 2 | Tidak |
| | | 3 | Tidak |
| | | 4 | Tidak |
| | | 5 | Tidak |

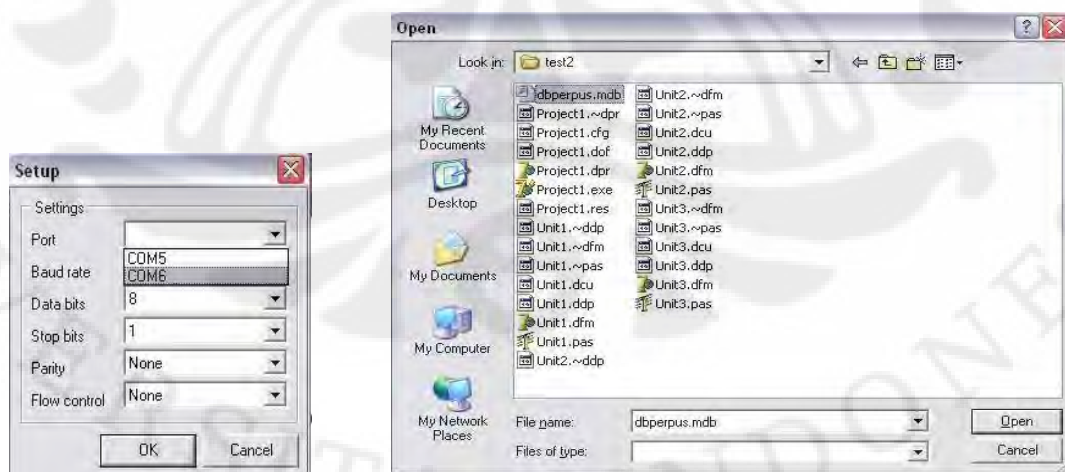
Berdasarkan Tabel 4.1 diatas, dapat diambil kesimpulan bahwa *reader* dapat mendeteksi *tag* meski diberi penghalang bila masih berada pada jarak jangkauan bacanya. Namun dalam percobaan dengan aluminium, *reader* tidak dapat mendeteksi *tag*. Hal ini dikarenakan aluminium merupakan bahan logam dimana logam bersifat kuat menolak dan meredam EMI (electro magnetic interference) dan RF (radio frequency) dibanding bahan pengujian lainnya[20]. Permasalahan ini bisa diatasi dengan peningkatan power supply dan kemampuan baca dari *reader* itu sendiri.

4.2 Pengujian Perangkat Lunak Berdasarkan Spesifikasi Fungsi

Perangkat lunak yang dibangun menggunakan bahasa pemrograman Delphi 7 ini bertugas untuk menjalankan beberapa fungsi. Fungsi-fungsi tersebut adalah setting *hardware*, pengaturan scan *tag* buku, penentuan tanggal pengembalian default, pengiriman email notifikasi masa pinjam yang akan habis, penentuan denda dan update basis data.

4.2.1 Pengujian Setting *Hardware*

Pengujian terhadap setting hardware ini meliputi pemilihan *comport* dan koneksi basis data. Hasil pengujian dapat dilihat pada Gambar 4.1(a) dan (b) berikut ini :



Gambar 4.1(a) Setting *Comport*, dan (b) Koneksi Basis data

4.2.2 Pengujian Pengaturan Scan Tag Buku Pada Sub Peminjaman

Pada sub peminjaman, sistem akan meminta scan ID *member* sebelum diizinkan untuk melakukan scan *tag* buku. Lama waktu yang diberikan untuk melakukan scan *tag* buku oleh member akan ditetapkan secara otomatis oleh sistem.

Pengujian ini dilakukan untuk menguji apakah waktu 15 detik yang ditetapkan oleh sistem sesuai dengan waktu yang dibutuhkan *reader* dalam melakukan pendeteksian terhadap buku pinjaman dalam 1 siklus untuk 1 ID *member*. Bila waktu dalam 1 siklus (15 detik) sudah habis maka *reader* scan buku akan non-aktif dan sudah tidak dapat melakukan peminjaman sehingga peminjam berikutnya dapat melakukan transaksi. Selain itu masing-masing member hanya diperbolehkan meminjam maksimal 3 item dalam 1 siklus. Hal ini perlu dipertimbangkan agar pelayanan proses peminjaman bisa berjalan tertib dan efektif.

Tabel 4.4 Hasil Pengujian Scan Buku dengan Batasan Waktu 15 detik

| Percobaan | jumlah <i>tag</i> buku yang di-scan | <i>tag</i> buku yang terbaca | keterangan | Waktu [detik] |
|-----------|-------------------------------------|------------------------------|------------|---------------|
| 1 | 1 | 1 | Sesuai | 15 |
| 2 | 1 | 1 | Sesuai | 15 |
| 3 | 2 | 2 | Sesuai | 15 |
| 4 | 2 | 2 | Sesuai | 15 |
| 5 | 2 | 2 | Sesuai | 15 |
| 6 | 3 | 3 | Sesuai | 15 |
| 7 | 3 | 3 | Sesuai | 15 |
| 8 | 3 | 3 | Sesuai | 15 |
| 9 | 4 | 3 | Sesuai | 15 |
| 10 | 5 | 3 | sesuai | 15 |

Dari tabel hasil percobaan terhadap lama waktu yang diatur untuk mengaktifkan *reader* scan buku dan pembatasan pinjaman sebanyak maksimal 3 item, dapat disimpulkan bahwa pengaturan telah berjalan dengan baik sesuai rancangan. Masing-masing peminjam diberi waktu 15 detik melakukan scan buku dan hanya diizinkan meminjam maksimal 3 buku sehingga jika peminjam tersebut melakukan

scan lebih dari 3 buku, sistem tidak akan mengizinkannya dan menghentikan proses pembacaan *tag* buku untuk member tersebut agar bisa dilanjutkan ke transaksi berikutnya (ditunjukkan oleh percobaan 9 dan 10).

4.2.3 Pengujian Pengiriman Email Notifikasi Masa Pinjam yang Akan Habis

Pengujian pengiriman email notifikasi masa pinjam yang akan habis ini dilakukan dengan mengkoneksikan komputer server pada perpustakaan dengan jaringan internet. Dalam hal ini, terhubung jaringan dengan koneksi internet tanpa proxy (direct) karena SMTP gmail yang digunakan sebagai account `_adminperpusftui` bekerja di jaringan yang mengizinkan koneksi internet public tanpa proxy (direct).

Sistem akan mengecek basis data pada field `_tanggal pengembalian default` setiap hari. Bila ditemukan masa pinjam buku yang tinggal 1 hari maka sistem akan melakukan pengiriman email kepada nama email dimana buku yang terdeteksi.

4.2.3.1 Pengujian Pengiriman Email Notifikasi ke Sebuah Account Email untuk Beberapa Domain Email yang Berbeda

Pengujian ini bertujuan untuk menguji kemampuan sub pengiriman email notifikasi dalam mengirim email ke beberapa domain email yang berbeda.

Hasil pengujian untuk pengiriman email notifikasi dari `adminperpusftui@gmail.com` ke 1 account email dengan 5 domain berbeda ditunjukkan oleh Tabel 4.5 berikut ini :

Tabel 4.5 Pengiriman Email ke account email dengan domain berbeda

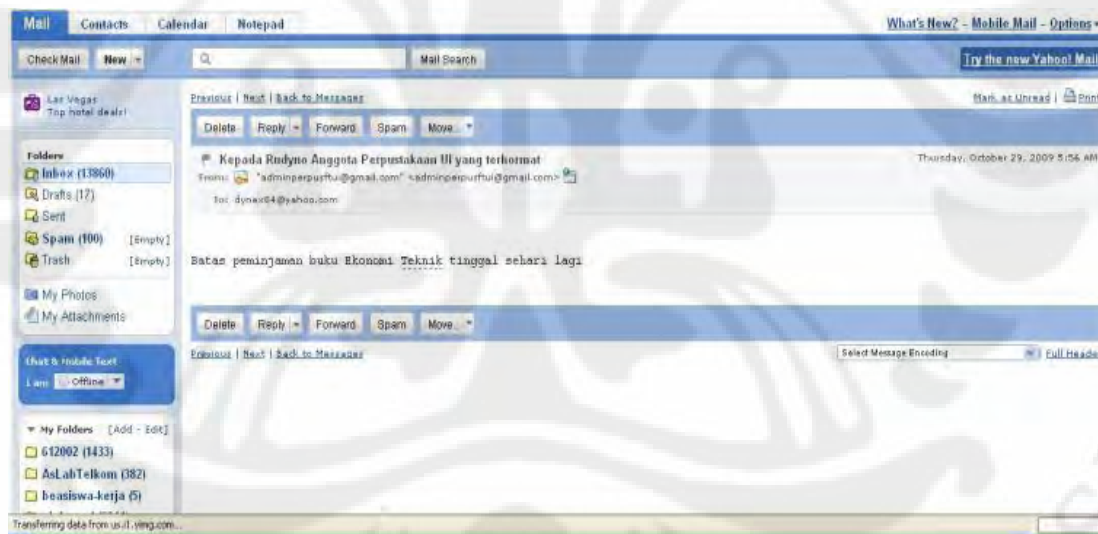
| Percobaan | Domain Email | Waktu Total [detik] |
|-----------|-----------------|---------------------|
| 1 | Gmail.com | 5 |
| 2 | Telkom.net | 6 |
| 3 | Yahoo.com | 6 |
| 4 | windowslive.com | 6 |
| 5 | Mail.com | 5 |

Dari hasil pengujian tersebut, dapat disimpulkan bahwa sub pengiriman email notifikasi dapat melakukan pengiriman email dengan baik ke semua domain email tanpa perbedaan waktu pengiriman.

4.2.3.2 Pengujian Pengiriman Email Notifikasi ke Beberapa Account Email Berbeda

Pengujian ini dilakukan untuk menguji kehandalan sistem dan waktu yang dibutuhkan oleh sistem dalam melakukan pengiriman email notifikasi ke beberapa account email berbeda. Hasil pengujian untuk pengiriman email notifikasi dari adminperpusftui@gmail.com kepada beberapa account email berbeda ditunjukkan oleh Tabel 4.6.

Berikut ini pada Gambar 4.2 merupakan contoh tampilan dari hasil pengiriman email notifikasi kepada salah satu account email anggota perpustakaan.



Gambar 4.2 Tampilan Email Notifikasi yang diterima oleh *Member* Perpustakaan

Tabel 4.6 Hasil Pengujian Pengiriman Email ke Beberapa Account Email Berbeda

| No | Jumlah Pengiriman | Percobaan | Waktu Total [detik] | Email diterima oleh anggota |
|----|-------------------|-----------|---------------------|-----------------------------|
| 1 | 2 account email | 1 | 12 | diterima |
| | | 2 | 13 | Diterima |
| | | 3 | 13 | Diterima |
| | | 4 | 12 | Diterima |
| | | 5 | 12 | diterima |
| 2 | 4 account email | 1 | 25 | diterima |
| | | 2 | 24 | Diterima |
| | | 3 | 25 | Diterima |
| | | 4 | 24 | Diterima |
| | | 5 | 24 | diterima |
| 3 | 6 account email | 1 | 35 | diterima |
| | | 2 | 36 | Diterima |
| | | 3 | 36 | Diterima |
| | | 4 | 36 | Diterima |
| | | 5 | 37 | diterima |
| 4 | 8 account email | 1 | 48 | diterima |
| | | 2 | 49 | Diterima |
| | | 3 | 49 | Diterima |
| | | 4 | 50 | Diterima |
| | | 5 | 47 | Diterima |
| 5 | 10 account email | 1 | 58 | Diterima |
| | | 2 | 59 | Diterima |
| | | 3 | 60 | Diterima |
| | | 4 | 58 | Diterima |
| | | 5 | 59 | Diterima |
| 6 | 20 account email | 1 | 119 | Diterima |
| | | 2 | 118 | Diterima |
| | | 3 | 120 | Diterima |
| | | 4 | 117 | Diterima |
| | | 5 | 118 | Diterima |

Dari hasil pengujian dapat dilihat secara umum bahwa pengiriman email notifikasi ke beberapa account email telah berjalan dengan baik. Rata-rata waktu

yang dibutuhkan untuk mengirim tiap email tetap 6 sekon meski dilakukan banyak pengiriman email. Tingkat keberhasilan dari penerimaan email yaitu sebesar :

$$30/30 \times 100\% = 100\%$$

4.2.4 Pengujian Penyimpanan Data ke Basis Data

Pengujian ini bertujuan untuk mengetahui apakah penyimpanan data ke basis data pada aplikasi telah berjalan dengan sesuai. Data-data hasil pembacaan *tag* ID *member* dan *tag* buku oleh *reader* akan tersimpan ke basis data baik untuk transaksi peminjaman dan transaksi pengembalian. Selain itu, pada aplikasi ini juga dapat dilakukan kegiatan update basis data seperti menambah data baru atau merubah data yang sudah ada pada basis data *member* maupun buku oleh administrator seperti pada Gambar 4.4. Pada aplikasi perangkat lunak yang dibangun menggunakan Delphi, tabel akan ditampilkan pada DBGrid. Gambar 4.3 merupakan tampilan untuk tabel basis data transaksi peminjaman pada Delphi.

| No | ID | Nama Anggota | email | ID buku |
|----|------------|--------------------|--------------------------|------------|
| 1 | 010215157B | Paramadina | dkina.hasby@gmail.com | 01020E05D0 |
| 2 | 01020F26CC | Fadliha | dina_toon@telk.com.net | 0102136191 |
| 3 | 01020F287A | Rudgho | dyhex84@yahoo.com | 0102136378 |
| 4 | 01020E313F | Sally Dwi Diklavia | dina_toon@windowsixe.com | 0102136844 |
| 5 | 0102151FFD | Andini | andini_hasby@yahoo.co.id | 010215146D |
| 6 | 0453100036 | ani | ne2k.taizan@gmail.com | 01023345F5 |
| 7 | 2341313431 | emilia | emilia.amirada@gmail.com | 0102876660 |
| 8 | 958756856 | hasbuchiar | hasbuchiar.mba@gmail.com | 0102089E45 |

Gambar 4.3 Tampilan Basis Data Transaksi Peminjaman Pada Delphi

Dari Gambar 4.3 diatas dapat dilihat bahwa proses penyimpanan data transaksi peminjaman yang dideteksi oleh *reader* telah berjalan dengan baik dan sesuai dengan rancangan.

Gambar 4.4 Tampilan Menu Ubah Data

4.3 Pengujian Ketahanan Sistem Dengan RFID Reader

Pengujian ketahanan sistem dengan keadaan *reader* menyala selama beberapa jam yang bertujuan untuk melihat apakah sistem dapat berjalan dengan baik pada kondisi sebenarnya saat diimplementasikan. *Reader* diletakkan pada meja dengan dilapisi kaca sebagai pelindung.

Pengujian dilakukan selama 6 jam dengan pengambilan data setiap 1 jam. Hasil pengujian ditunjukkan oleh Tabel 4.7 dibawah ini :

Tabel 4.7 Hasil Pengujian Ketahanan Sistem Dengan RFID Menyala selama 6 jam

| Lama waktu (jam) | Jumlah Antrian | Tag buku yg di-scan | Tag buku terdeteksi | keterangan |
|------------------|----------------|---------------------|---------------------|------------|
| jam ke-1 | 40 | 85 | 85 | OKE |
| jam ke-2 | 30 | 70 | 70 | OKE |
| jam ke-3 | 24 | 48 | 48 | OKE |
| jam ke-4 | 30 | 50 | 50 | OKE |
| jam ke-5 | 25 | 45 | 45 | OKE |
| jam ke-6 | 30 | 62 | 62 | OKE |

Dari pengujian pada Tabel 4.7 dapat disimpulkan bahwa pembacaan RFID *tag* selama *reader* menyala 6 jam tidak ada masalah dalam pembacaan *tag*. ID member terdeteksi dan buku pinjaman juga terdeteksi.

4.4 Analisis Keseluruhan Perangkat Lunak Berdasarkan Spesifikasi Fungsi dari Sistem Otomasi Peminjaman dan Pengembalian Buku

Setelah dilakukan pengujian terhadap spesifikasi fungsi secara keseluruhan pada sub peminjaman, sub pengembalian, dan sub pengiriman email notifikasi maka didapatkan hasil bahwa semua fungsi pada perangkat lunak berjalan dengan baik sesuai rancangan. Mulai dari setting hardware seperti ditunjukkan oleh Gambar 4.1(a) dan (b), penentuan tanggal pengembalian default yaitu 7 hari setelah transaksi peminjaman, pengaturan jumlah maksimum peminjaman buku yaitu 3 item seperti ditunjukkan Tabel 4.4, penentuan denda atas keterlambatan pengembalian buku yaitu Rp.1000,-/hari, penyimpanan data transaksi peminjaman dan pengembalian ke basis data seperti ditunjukkan oleh Gambar 4.3, serta pengiriman email notifikasi bahwa masa pinjam akan segera habis seperti pada Tabel 4.5, Tabel 4.6 dan Gambar 4.2.

BAB 5
KESIMPULAN

1. Dengan merancang fungsi pembacaan *tag* ID member dan buku yang dilakukan langsung oleh 1 *reader* maka sistem ini memudahkan peminjam dalam melakukan peminjaman dan pengembalian buku tanpa berinteraksi dengan komputer atau manusia.
2. Dari hasil pengujian dapat dilihat secara umum bahwa fungsi pengecekan buku yang disertai pembatasan waktu dan jumlah item pinjaman, penyimpanan dan pembaharuan basis data perpustakaan, serta perhitungan denda atas keterlambatan pengembalian telah berjalan dengan baik sesuai rancangan.
3. Aplikasi pemberitahuan masa pinjam yang akan segera habis melalui pengiriman email notifikasi dapat berjalan dengan baik dan rata-rata waktu yang dibutuhkan untuk mengirim tiap email adalah 6 detik.

DAFTAR ACUAN

- [1] <http://www.gaorfid.com>. Diakses tanggal 5 Maret 2009
- [2] Maryono, –Dasar-dasar Radio Frequency Identification (RFID), Teknologi yang Berpengaruh di Perpustakaan”. 2005
- [3] 0826.RFID_Application_For_Libraries.PDF. diakses tanggal 12 Mei 2009
- [4] Dedi Supriatna, –Studi Mengenai Aspek Privasi Pada Sistem RFID”. Sekolah Teknik Elektro Dan Informatika Institut Teknologi Bandung, Januari 2007
- [5] Wikipedia Bahasa Indonesia Diakses tanggal 16 Februari 2009
<http://wikipedia.org>
- [6] <http://puslit.petra.ac.id/journals/industrial>, diakses tanggal 16 Februari 2009
- [7] Klaus Finkenzeller, *RFID HANDBOOK Fundamentals and Application in Contactless Smart Cards and Identification* (West Sussex: John Wiley & Sons Ltd, 2003), hal 9.
- [8] White Paper –ABCs of RFID: Understanding and using radio frequency identification”
- [9] Zeplin Jiwa Husada Tarigan, –Integrasi Teknologi RFID Dengan Teknologi ERP Untuk Otomasi Data (Studi Kasus Pada Gudang Barang Jadi Perusahaan Furniture)”. *Jurnal Teknik Industri* Vol. 6, No. 2, Desember 2004: 134 – 141
- [10] United States Government Accountability Office (2005), *Informaton Security : Radio Frequency Identification Technology in the Federal Government*, <http://www.gao.gov/new.items/d05551.pdf>, 4 Januari 2009
- [11] Klaus Finkenzeller, *RFID HANDBOOK Fundamentals and Application in Contactless Smart Cards and Identification* (West Sussex: John Wiley & Sons Ltd, 2003), hal 311.

- [12] Klaus Finkenzeller, *RFID HANDBOOK Fundamentals and Application in Contactless Smart Cards and Identification* (West Sussex: John Wiley & Sons Ltd, 2003), hal 312.
- [13] Klaus Finkenzeller, *RFID HANDBOOK Fundamentals and Application in Contactless Smart Cards and Identification* (West Sussex: John Wiley & Sons Ltd, 2003), hal 316.
- [14] United States Government Accountability Office (2005), *Information Security : Radio Frequency Identification Technology in the Federal Government*, <http://www.gao.gov/new.items/d05551.pdf>, 4 Januari 2009
- [15] Finkenzeller, Klaus.(2003).*RFID Handbook :Fundamentals and Application in Contactless Smart Cards and Identification, Second Edition*.Singapore : John Wiley & Sons, Inc.
- [16] Simon Holloway. *RFID: An Introduction*". (EMEA: Microsoft EMEA, 2006)
- [17] [http://aurino.com/pengertian basis data](http://aurino.com/pengertian_basis_data). diakses tanggal 16 Mei 2009
- [18] Krizzna's weblog, *pengertian basis data*". Diakses tanggal 16 Mei 2009
- [19] Idhawati Hestiningih, PDF Pemrograman Delphi
- [20] http://en.wikipedia.org/wiki/Electromagnetic_interference diakses tanggal 20 November 2009

DAFTAR PUSTAKA

Gibson, Candace. Kevin Bonsor. *"How RFID works"*. Diakses 11 Desember 2008 dari howstuffworks.

<http://electronics.howstuffworks.com/rfid.htm>

Serial Communication Overview. Diakses 11 Desember 2008 dari National Instrument.

<http://zone.ni.com/devzone/cda/tut/p/id/2895>

Jennifer, S.(2005).*RFID How It Works*.Michigan : Michigan State University.

Saputra, Fahdly H. Skripsi –*Sistem Absensi Menggunakan Teknologi RFID*”. 2007

SUB PEMINJAMAN

```
unit Unit1;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, DB, ADODB, ExtCtrls, IdBaseComponent, IdComponent,
  IdTCPConnection, IdTCPClient, IdMessageClient, IdSMTP, IdMessage,
  IdIOHandler, IdIOHandlerSocket, IdSSLOpenSSL, ComCtrls, CPort, jpeg;

type
  TForm1 = class(TForm)
    Edit1: TEdit;
    Edit2: TEdit;
    Label1: TLabel;
    Label2: TLabel;
    Edit5: TEdit;
    Edit7: TEdit;
    Label4: TLabel;
    Label7: TLabel;
    Label8: TLabel;
    Label9: TLabel;
    Label10: TLabel;
    Edit8: TEdit;
    Edit6: TEdit;
    TableRFID1: TADOTable;
    DataSource1: TDataSource;
    TableRFID2: TADOTable;
    DataSource2: TDataSource;
    Timer1: TTimer;
    OpenDialog1: TOpenDialog;
    DataSource3: TDataSource;
    DataSource4: TDataSource;
    TableRFID3: TADOTable;
    TableRFID4: TADOTable;
    Button6: TButton;
    Edit4: TEdit;
    Label5: TLabel;
    Panel1: TPanel;
    IdSMTP1: TIdSMTP;
    IdSSLIHandlerSocket1: TIdSSLIHandlerSocket;
    IdMessage1: TIdMessage;
    Memo1: TMemo;
    Memo2: TMemo;
    Memo3: TMemo;
    Panel2: TPanel;
    DateTimePicker1: TDateTimePicker;
    C1: TComPort;
    DateTimePicker2: TDateTimePicker;
    DateTimePicker3: TDateTimePicker;
    Timer3: TTimer;
    Edit10: TEdit;
    Timer2: TTimer;
    Edit11: TEdit;
    Edit12: TEdit;
    Edit9: TEdit;
    Label3: TLabel;
    Edit13: TEdit;
    Panel3: TPanel;
    Panel4: TPanel;
    Panel5: TPanel;
    Panel6: TPanel;
    Timer4: TTimer;
    Label6: TLabel;

    procedure Button6Click(Sender: TObject);
  end;
end;
```

```

procedure IdSSLIHandlerSocket1Status(ASender: TObject;
  const AStatus: TIdStatus; const AStatusText: String);

procedure Timer1Timer(Sender: TObject);

procedure Timer3Timer(Sender: TObject);

procedure Timer2Timer(Sender: TObject);
procedure Panel3Click(Sender: TObject);

procedure Panel4Click(Sender: TObject);

procedure Panel6Click(Sender: TObject);
procedure Panel5Click(Sender: TObject);
procedure Timer4Timer(Sender: TObject);
procedure Panel3MouseUp(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
procedure Panel3MouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
procedure Panel4MouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
procedure Panel4MouseUp(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
procedure Panel5MouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
procedure Panel5MouseUp(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
procedure Panel6MouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
procedure Panel6MouseUp(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);

private
  { Private declarations }
  procedure DateTimeToDuration(const Delta: TDateTime; var Days: DWORD;
    var Hours, Minutes, Seconds, MilliSeconds: Word);

public
  { Public declarations }
  waktu : TDateTime;
  x : integer;
end;

var
  Form1: TForm1;

implementation

uses Unit2, Unit3, Unit4, Unit5;

{$R *.dfm}

//=====//
// Rutin ini digunakan untuk menghitung selisih waktu
//=====//
procedure TForm1.DateTimeToDuration(const Delta: TDateTime; var Days: DWORD;
var Hours, Minutes, Seconds, MilliSeconds: Word);
begin
  Days := Trunc(Delta);
  Hours := Trunc((Delta - Days) * 24);
  Minutes := Trunc(((Delta - Days) - (Hours/24)) * 1440);
  Seconds := Trunc(((Delta - Days) - (Hours/24) -
  (Minutes/1440)) * 86400);
  MilliSeconds := Trunc(((Delta - Days) - (Hours/24) -
  (Minutes/1440) - (Seconds/86400)) * 86400000);
end;

procedure TForm1.Button6Click(Sender: TObject);

```



```

begin
form1.Close
end;
//=====//
// Rutin ini merupakan fungsi untuk mengecek masa pinjam dan mengirim email notifikasi //
//=====//
procedure TForm1.IdSSLIOHandlerSocket1Status(ASender: TObject;
const AStatus: TIdStatus; const AStatusText: String);
begin
Memo3.Lines.Add('idSSL: ' + 'sending');
Memo2.Lines.Add(AStatusText);
end;

//=====//
// Fungsi reader membaca tag buku //
//=====//
procedure TForm1.Timer1Timer(Sender: TObject);
var
nomor,i,a : integer;
buku,bukul : string;
adadb : Boolean;
optsdb : TlocateOptions;
begin
Cl.ReadStr(buku,nomor);
Edit11.Text:= buku;

if length(buku) > 10 then
begin
for a := 1 to 10 do
begin
bukul:= bukul + buku[a];

for i:=1 to TableRFID2.RecordCount+1 do
begin
adadb := TableRFID2.Locate('No',i,optsdb);
if adadb then
begin
if TableRFID2.FieldValues['ID Buku'] = bukul then
begin
Edit1.Text := TableRFID2.FieldValues['ID buku'];
Edit2.Text := TableRFID2.FieldValues['Judul Buku'];
Edit7.Text := DateToStr(now);
DateTimePicker2.Date := StrToDate(edit7.Text);
DateTimePicker3.Date := DateTimePicker2.Date+7;
Edit4.Text := DateToStr (DateTimePicker3.Date);
Edit9.Text := TimeToStr(now);

TableRFID3.Append;
TableRFID3['No']:= TableRFID3.RecordCount +1;
TableRFID3['ID']:= Edit5.Text;
TableRFID3['Nama Anggota']:= Edit6.Text;
TableRFID3['email']:= Edit8.Text;
TableRFID3['ID buku']:= TableRFID2.FieldValues['ID buku'];
TableRFID3['Judul Buku']:= TableRFID2.FieldValues['Judul Buku'];
TableRFID3['tgl peminjaman']:= Edit7.Text;
TableRFID3['jam peminjaman']:= Edit9.Text;
TableRFID3['tgl pengembalian default']:= Edit4.Text;
TableRFID3.Post;

x := x + 1;
if x = 3 then
begin
timer1.Enabled := False;
MessageDlg('telah mencapai batas pinjaman maksimum!',mtConfirmation,[mbOK],0);
end;
end;
end;
end;
end;
end;
end;
end;

```

```

end;
end;
end;
end;
end;

procedure TForm1.Timer3Timer(Sender: TObject);
var
  D      : DWORD;
  H, M, S, MS : Word;
begin
  DateTimeToDuration(Now - waktu, D, H, M, S, MS);
  Edit13.Text := IntToStr(S);
  if S = 15 then
  begin
    Timer1.Enabled := False;

    edit5.Text := '';
    edit6.Text := '';
    edit8.Text := '';
    edit1.Text := '';
    edit2.Text := '';
    edit4.Text := '';
    edit7.Text := '';
    edit9.Text := '';
    MessageDlg('waktu anda habis!', mtConfirmation, [mbOK], 0);
  end;
  Timer2.Enabled := True;
end;

//=====//
//      Fungsi reader membaca tag member      //
//=====//
procedure TForm1.Timer2Timer(Sender: TObject);
var
  nomor,i,a : integer;
  member,member1 : string;
  adadb2      : Boolean;
  optsdb2     : TlocateOptions;

begin
  Cl.ReadStr(member,nomor);
  Edit11.Text:= member;

  if length(member) > 10 then
  begin
    for a := 1 to 10 do
    begin
      member1:= member1 + member[a];

      for i:=1 to TableRFID1.RecordCount+1 do
      begin
        adadb2 := TableRFID1.Locate('No',i,optsdb2);
        if adadb2 then
        begin
          if TableRFID1.FieldValues['ID'] = member1 then
          begin
            Edit5.Text := TableRFID1.FieldValues['ID'];
            Edit6.Text := TableRFID1.FieldValues['Nama Anggota'];
            Edit8.Text := TableRFID1.FieldValues['email'];
          end;
        end;
      end;
    end;
  end;
  waktu := now;
  Timer1.enabled:= True;
  Timer3.enabled:= True;
  x := 0;
end;
end;

```

```

end;
end;
end;
//Timer2.Enabled := True;
end;

procedure TForm1.Panel13Click(Sender: TObject);
begin
  if OpenDialog1.Execute then
  begin
    begin
      TableRFID1.ConnectionString := 'Provider=Microsoft.Jet.OLEDB.4.0;Data Source=' +
opendialog1.FileName + ';Persist Security Info=False';
      TableRFID1.TableName := 'Dbase Member';

      TableRFID2.ConnectionString := 'Provider=Microsoft.Jet.OLEDB.4.0;Data Source=' +
opendialog1.FileName + ';Persist Security Info=False';
      TableRFID2.TableName := 'Dbase Buku';

      Form3.TableRFID1.ConnectionString := 'Provider=Microsoft.Jet.OLEDB.4.0;Data Source=' +
opendialog1.FileName + ';Persist Security Info=False';
      Form3.TableRFID1.TableName := 'Dbase Member';

      Form3.TableRFID2.ConnectionString := 'Provider=Microsoft.Jet.OLEDB.4.0;Data Source=' +
opendialog1.FileName + ';Persist Security Info=False';
      Form3.TableRFID2.TableName := 'Dbase Buku';

      TableRFID3.ConnectionString := 'Provider=Microsoft.Jet.OLEDB.4.0;Data Source=' +
opendialog1.FileName + ';Persist Security Info=False';
      TableRFID3.TableName := 'Tabel Transaksi Peminjaman';

      TableRFID4.ConnectionString := 'Provider=Microsoft.Jet.OLEDB.4.0;Data Source=' +
opendialog1.FileName + ';Persist Security Info=False';
      TableRFID4.TableName := 'Tabel Transaksi Pengembalian';

end;

TableRFID1.Active := true;
DataSource1.DataSet := TableRFID1;

TableRFID2.Active := true;
DataSource2.DataSet := TableRFID2;

Form3.TableRFID1.Active := true;
Form3.DataSource1.DataSet := Form3.TableRFID1;
form4.DBGrid1.DataSource := DataSource1;

Form3.TableRFID2.Active := true;
Form3.DataSource2.DataSet := Form3.TableRFID2;
form5.DBGrid1.DataSource := DataSource2;

TableRFID3.Active := true;
DataSource3.DataSet := TableRFID3;
form2.DBGrid1.DataSource := DataSource3;

TableRFID4.Active := true;
DataSource4.DataSet := TableRFID4;

C1.ShowSetupDialog;
C1.Connected:=true;
C1.FlowControl.FlowControl:=fcHardware;

Timer2.Enabled:=True;

Edit10.Text := DateToStr(now);
datetimepicker1.Date := StrToDate(Edit10.Text);

```

```

end;

procedure TForm1.Panel4Click(Sender: TObject);
begin
  form3.Show;
end;

//=====//
//      Fungsi untuk mengirim email notifikasi      //
//=====//
procedure TForm1.Panel6Click(Sender: TObject);
var
  i, warned : integer;
  nomor, nama, buku : string;
  adadb4      : Boolean;
  optbdb4     : TlocateOptions;
begin
  for i:=1 to TableRFID3.RecordCount+1 do
  begin
    adadb4 := TableRFID3.Locate('No', i, optbdb4);
    if adadb4 then
    begin
      warned := Trunc(TableRFID3.FieldValues['tgl pengembalian default']) -
Trunc(DateTimePicker1.Date);
      if warned = 1 then
      begin
        nomor:=TableRFID3.FieldValues['No'];
        nama:=TableRFID3.FieldValues['Nama Anggota'];
        buku:=TableRFID3.FieldValues['Judul Buku'];
        IdMessage1.From.Address := 'adminperpusftui@gmail.com';
        IdMessage1.Recipients.EmailAddresses := TableRFID3.FieldValues['email'];
        IdMessage1.Subject := 'Kepada '+nama+' Anggota Perpustakaan UI yang terhormat';
        IdMessage1.Body.Text := 'Batas peminjaman buku '+buku+' tinggal sehari lagi';
        IdSMTP1.Connect();
        if IdSMTP1.Connected then
        begin
          try
            IdSMTP1.Send(IdMessage1);
          finally
            // MessageDlg(nomor+'. Pesan Terkirim ke '+nama+' bahwa buku '+buku+' harus
dikembalikan sehari lagi!', mtConfirmation, [mbOK], 0);
            IdSMTP1.Disconnect;
          end
          end
          else
            MessageDlg('Error in Connection!', mtConfirmation, [mbOK], 0);

            IdSMTP1.Disconnect;
            Sleep(1000);
          end
          //else
          //MessageDlg(nomor+'. Status peminjaman buku '+buku+' oleh '+nama+' masih
OK!', mtConfirmation, [mbOK], 0);
        end;

        end;
        MessageDlg('pengecekan buku selesai!', mtConfirmation, [mbOK], 0);
        end;
        procedure TForm1.Panel5Click(Sender: TObject);
        begin
          form2.Show;
        end;

        procedure TForm1.Timer4Timer(Sender: TObject);
        begin
          label6.Left := label6.Left+50;
          if label6.Left=600 then label6.Left:= -550;
        end;

```

```

procedure TForm1.Panel3MouseUp(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
begin
panel3.Color:=$00FF8000;
end;

procedure TForm1.Panel3MouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
begin
panel3.Color:=clyellow;
end;

procedure TForm1.Panel4MouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
begin
panel4.Color:=clyellow;
end;

procedure TForm1.Panel4MouseUp(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
begin
panel4.Color:=$00FF8000;
end;

procedure TForm1.Panel5MouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
begin
panel5.Color:=clyellow;
end;

procedure TForm1.Panel5MouseUp(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
begin
panel5.Color:=$00FF8000;
end;

procedure TForm1.Panel6MouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
begin
panel6.Color := clyellow;
end;

procedure TForm1.Panel6MouseUp(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
begin
panel6.Color:=$00FF8000;
end;

end.

unit Unit2;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, Grids, DBGrids, StdCtrls;

type
  TForm2 = class(TForm)
    DBGrid1: TDBGrid;
    Label1: TLabel;
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var

```

```

Form2: TForm2;

implementation
{$R *.dfm}

end.

unit Unit3;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, Grids, DBGrids, StdCtrls, ExtCtrls, DB, ADODB;
type
  TForm3 = class(TForm)
    Panel1: TPanel;
    Panel2: TPanel;
    Label1: TLabel;
    Label2: TLabel;
    NewMember: TButton;
    EditMember: TButton;
    NewBook: TButton;
    EditBook: TButton;
    Edit1: TEdit;
    Edit2: TEdit;
    Edit3: TEdit;
    Edit4: TEdit;
    Edit5: TEdit;
    Edit6: TEdit;
    Edit7: TEdit;
    Edit8: TEdit;
    Edit9: TEdit;
    Edit10: TEdit;
    Edit11: TEdit;
    Edit12: TEdit;
    Label13: TLabel;
    Label14: TLabel;
    Label15: TLabel;
    Label16: TLabel;
    Label17: TLabel;
    Label18: TLabel;
    Label19: TLabel;
    Label10: TLabel;
    Label11: TLabel;
    Label12: TLabel;
    Label13: TLabel;
    Label14: TLabel;
    DataSource1: TDataSource;
    DataSource2: TDataSource;
    TableRFID1: TADOTable;
    TableRFID2: TADOTable;
    Button1: TButton;
    Button2: TButton;
    procedure NewMemberClick(Sender: TObject);
    procedure EditMemberClick(Sender: TObject);

    procedure NewBookClick(Sender: TObject);
    procedure EditBookClick(Sender: TObject);
    procedure Button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);

  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form3: TForm3;

```

```

implementation

uses Unit4, Unit5;

{$R *.dfm}

procedure TForm3.NewMemberClick(Sender: TObject);
begin
TableRFID1.Append;
TableRFID1['No'] := TableRFID1.RecordCount + 1;
TableRFID1['ID']:= Edit1.Text;
TableRFID1['Nama Anggota']:= Edit2.Text;
TableRFID1['Jurusan']:= Edit3.Text;
TableRFID1['NPM/NIP']:= Edit4.Text;
TableRFID1['Jabatan']:= Edit5.Text;
TableRFID1['alamat rumah']:= Edit6.Text;
TableRFID1['no telp']:= Edit7.Text;
TableRFID1['email']:= Edit8.Text;
TableRFID1.Post;
ShowMessage('Data Member Telah Selesai Ditambahkan');
end;

procedure TForm3.EditMemberClick(Sender: TObject);
begin
TableRFID1.Edit;
TableRFID1['No'] := TableRFID1.RecordCount + 1;
TableRFID1['ID']:= Edit1.Text;
TableRFID1['Nama Anggota']:= Edit2.Text;
TableRFID1['Jurusan']:= Edit3.Text;
TableRFID1['NPM/NIP']:= Edit4.Text;
TableRFID1['Jabatan']:= Edit5.Text;
TableRFID1['alamat rumah']:= Edit6.Text;
TableRFID1['no telp']:= Edit7.Text;
TableRFID1['email']:= Edit8.Text;
TableRFID1.Post;
ShowMessage('Data Member Telah Selesai Diedit');
end;

procedure TForm3.NewBookClick(Sender: TObject);
begin
TableRFID2.Append;
TableRFID2['No'] := TableRFID2.RecordCount + 1;
TableRFID2['ID Buku']:= Edit9.Text;
TableRFID2['Judul Buku']:= Edit10.Text;
TableRFID2['Pengarang']:= Edit11.Text;
TableRFID2['Penerbit']:= Edit12.Text;
TableRFID2.Post;
ShowMessage('Data Buku Telah Selesai Ditambah');
end;

procedure TForm3.EditBookClick(Sender: TObject);
begin
TableRFID2.Edit;
TableRFID2['No'] := TableRFID2.RecordCount + 1;
TableRFID2['ID']:= Edit9.Text;
TableRFID2['Judul Buku']:= Edit10.Text;
TableRFID2['Pengarang']:= Edit11.Text;
TableRFID2['Penerbit']:= Edit12.Text;
TableRFID2.Post;
ShowMessage('Data Buku Telah Selesai Diedit');
end;

procedure TForm3.Button1Click(Sender: TObject);
begin
Form4.Show;
end;

procedure TForm3.Button2Click(Sender: TObject);

```

```

begin
Form5.Show;
end;
end.

unit Unit4;

interface

uses
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, StdCtrls, Grids, DBGrids;

type
TForm4 = class(TForm)
DBGrid1: TDBGrid;
Label1: TLabel;
private
{ Private declarations }
public
{ Public declarations }
end;

var
Form4: TForm4;
implementation
{$R *.dfm}
end.

unit Unit5;
interface
uses
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, StdCtrls, Grids, DBGrids;
type
TForm5 = class(TForm)
DBGrid1: TDBGrid;
Label1: TLabel;
private
{ Private declarations }
public
{ Public declarations }
end;
var
Form5: TForm5;

implementation
{$R *.dfm}
end.

```


SUB PENGEMBALIAN

```
unit Unit1;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ComCtrls, CPort, DB, ADODB, ExtCtrls, jpeg;

type
  TForm1 = class(TForm)
    DateTimePicker1: TDateTimePicker;
    Button1: TButton;
    Edit1: TEdit;
    DateTimePicker2: TDateTimePicker;
    Button2: TButton;
    Label1: TLabel;
    Label2: TLabel;
    DateTimePicker3: TDateTimePicker;
    Label3: TLabel;
    Edit2: TEdit;
    Label4: TLabel;
    Label5: TLabel;
    DataSource1: TDataSource;
    DataSource2: TDataSource;
    ADOTable1: TADOTable;
    ADOTable2: TADOTable;
    OpenDialog1: TOpenDialog;
    C1: TComPort;
    Button4: TButton;
    Button5: TButton;
    Timer1: TTimer;
    Edit3: TEdit;
    Edit4: TEdit;
    Edit5: TEdit;
    Edit6: TEdit;
    Image1: TImage;
    procedure Button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure Button3Click(Sender: TObject);
    procedure Button4Click(Sender: TObject);
    procedure Button5Click(Sender: TObject);
    procedure Timer1Timer(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
    waktu : TDateTime;
  end;

var
  Form1: TForm1;

implementation

uses Unit2;

{$R *.dfm}

procedure TForm1.Button1Click(Sender: TObject);
var
  x : real;
  y : string;
  z : real;
begin
  x := Int(DateTimePicker2.Date-DateTimePicker1.Date);
  //z := x * 1000;
  //edit2.Text := FloatToStr(round(z));
  if x<=0 then
```

```

begin
MessageDlg('Tidak kena denda',mtConfirmation,[mbOK],0);
end

else
z := x*1000;
edit2.Text := FloatToStr(round(z));
y := FloatToStr(round(x));
edit1.Text := y;
end;

procedure TForm1.Button2Click(Sender: TObject);
begin
form1.close;
end;

procedure TForm1.Button3Click(Sender: TObject);
begin
DateTimePicker1.Date:= DateTimePicker3.Date+7
end;

procedure TForm1.Button4Click(Sender: TObject);
begin
Form2.show;
end;

procedure TForm1.Button5Click(Sender: TObject);
begin
if OpenFileDialog1.Execute then
begin
ADOTable1.ConnectionString := 'Provider=Microsoft.Jet.OLEDB.4.0;Data Source=' +
opendialog1.FileName + ';Persist Security Info=False';
ADOTable1.TableName := 'Tabel Transaksi Peminjaman';

ADOTable2.ConnectionString := 'Provider=Microsoft.Jet.OLEDB.4.0;Data Source=' +
opendialog1.FileName + ';Persist Security Info=False';
ADOTable2.TableName := 'Tabel Transaksi Pengembalian';

end;

ADOTable1.Active := true;
DataSource1.DataSet := ADOTable1;

ADOTable2.Active := true;
DataSource2.DataSet := ADOTable2;
form2.DBGrid1.DataSource := DataSource2;
C1.ShowSetupDialog;
C1.Connected:=true;
C1.FlowControl.FlowControl:=fcHardware;

waktu := now;
timer1.Enabled := True;
end;

procedure TForm1.Timer1Timer(Sender: TObject);
Var
kata,katal,hitung,y: string;
nomor,i,a: integer;
dendamod2,totaldenda,x : real;
adadb : Boolean;
optsdb : TlocateOptions;

begin
C1.ReadStr(kata,nomor);
Edit3.Text:= kata;

if length(kata) > 10 then
begin

```

```

for a := 1 to 10 do
begin
katal:= katal + kata[a];

//for i:=1 to ADOTable1.recordcount+1 do
for i:=1 to 5 do
begin
adadb := ADOTable1.Locate('No',i,optsdb);
if adadb then
begin
if ADOTable1.FieldValues['ID Buku'] = katal then
begin
Edit5.Text := ADOTable1.FieldValues['Nama Anggota'];
Edit6.Text := ADOTable1.FieldValues['Judul Buku'];
Edit4.Text := DateToStr(now);
DateTimePicker3.Date := ADOTable1.FieldValues['tgl peminjaman'];
DateTimePicker2.Date := StrToDate(edit4.Text);
DateTimePicker1.Date := ADOTable1.FieldValues['tgl pengembalian default'];
dendamod2:= datetimepicker2.date-datetimepicker1.date;

if dendamod2 > 0 then
x := Int(dendamod2);
totaldenda:= x*1000;
edit2.Text := FloatToStr(round(totaldenda));
y := FloatToStr(round(x));
edit1.Text := y;
MessageDlg('Denda yang harus dibayarkan adalah '+edit2.Text+',mtConfirmation,[mbOK],0);

ADOTable2.Append;
ADOTable2['No']:= ADOTable2.RecordCount +1;
ADOTable2['ID']:= ADOTable1.FieldValues['ID'];
ADOTable2['Nama Anggota']:= ADOTable1.FieldValues['Nama Anggota'];
ADOTable2['email']:= ADOTable1.FieldValues['email'];
ADOTable2['ID Buku']:= ADOTable1.FieldValues['ID Buku'];
ADOTable2['Judul Buku']:= ADOTable1.FieldValues['Judul Buku'];
ADOTable2['tgl peminjaman']:= ADOTable1.FieldValues['tgl peminjaman'];
ADOTable2['tgl pengembalian default']:= ADOTable1.FieldValues['tgl pengembalian default'];
ADOTable2['tgl pengembalian real']:= DateToStr(now);
ADOTable2['total denda'] := edit2.Text;
ADOTable2.Post;

end;
end;
end;
end;
end;

timer1.Enabled := True;
end;
end.

unit Unit2;
interface
uses
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, StdCtrls, Grids, DBGrids;
type
TForm2 = class(TForm)
DBGrid1: TDBGrid;
Label1: TLabel;
private
{ Private declarations }
public
{ Public declarations }
end;
var
Form2: TForm2;
implementation
{$R *.dfm}
end.

```