



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB II

TELAAH LITERATUR

Bab ini berisi tentang teori-teori dan definisi para ahli yang digunakan dalam pembahasan penelitian, mulai dari objek penelitian, hingga sistem dan atau teknologi yang digunakan dalam penelitian ini.

2.1 Jalan Tol.

Menurut Peraturan Pemerintah Republik Indonesia Nomor 15 Tahun 2005 tentang Jalan Tol, yang dimaksud dengan jalan tol adalah jalan umum yang merupakan bagian sistem jaringan jalan dan sebagian jalan nasional yang penggunaannya diwajibkan membayar tol. Sedangkan tol adalah sejumlah uang tertentu yang dibayarkan untuk penggunaan jalan tol.

Jalan tol diatur oleh Badan Pengatur Jalan Tol, yang disingkat BPJT, adalah badan yang dibentuk oleh menteri, ada di bawah, dan bertanggung jawab kepada menteri.

Jalan tol memiliki maksud, tujuan dan lingkup tersendiri. Penyelenggaraan jalan tol dimaksudkan untuk mewujudkan pemerataan pembangunan serta menjaga keseimbangan dalam pengembangan wilayah secara adil.

Sedangkan penyelenggaraan jalan tol bertujuan untuk meningkatkan efisiensi pelayanan jasa distribusi guna menunjang peningkatan pertumbuhan ekonomi terutama di wilayah yang sudah padat atau tinggi tingkat

pertumbuhannya, yang berarti, jalan tol membantu memperlancar lalu lintas di daerah yang padat, dan yang terakhir adalah meringankan beban dana pemerintah dalam melakukan perawatan dan pemeliharaan jalan, melalui partisipasi pengguna jalan tol. Lingkup Peraturan Pemerintah ini mencakup pengaturan penyelenggaraan jalan tol, BPJT, serta hak dan kewajiban Badan Usaha dan pengguna jalan tol itu sendiri.

Saat ini, jalan tol di Indonesia dikelola oleh 27 Badan Usaha Jalan Tol, yang dibawah oleh BPJT, dan diatur oleh Undang Undang RI No. 38 Tahun 2004 Tentang Jalan, serta diperkuat oleh 5 Peraturan Pemerintah, 3 Peraturan Presiden, 10 Peraturan Menteri, 7 Keputusan Menteri, 2 Keputusan Kepala BPJT, dan 1 Keputusan Gubernur.

Tabel 2.1. Tabel Daftar Badan Usaha Jalan Tol di Bawah BPJT

Daftar Badan Usaha Jalan Tol di Bawah BPJT
<u>PT. BINTARO SERPONG DAMAI</u>
<u>PT. BOSOWA MARGA NUSANTARA</u>
<u>PT. CITRA MARGA NUSAPHALA PERSADA TBK</u>
<u>PT. CITRA MARGATAMA SURABAYA</u>
<u>PT. JALAN TOL SEKSI IV</u>
<u>PT. JASA MARGA</u>
<u>PT. CITRA MARGATAMA SURABAYA</u>
<u>PT. CITRA WASPPHUTOWA</u>
<u>PT. JAKARTA LINGKAR BARAT SATU</u>
<u>PT. KRESNA KUSUMA DYANDRA MARGA</u>
<u>PT. LINTAS MARGA SEDAYA</u>
<u>PT. MARGABUMI ADHIKARAYA</u>
<u>PT. MARGA HANURATA INTRINSIC</u>
<u>PT. MARGA NUJYASUMO AGUNG</u>
<u>PT. MARGA SARANA JABAR</u>
<u>PT. MARGA SETIAPURITAMA</u>
<u>PT. MARGA TRANS NUSANTARA</u>
<u>PT. MTD CTP EXPRESSWAY</u>
<u>PT. PEJAGAN PEMALANG TOL ROAD</u>

PT. PEMALANG BATANG TOL ROAD
PT. SEMESTA MARGA RAYA
PT. TRANS JABAR TOL
PT. TRANS-JAWA PAS PRO JALAN TOL
PT. TRANSLINGKAR KITA JAYA

2.2 Alat Pembayaran.

Menurut Bank Indonesia, alat pembayaran di Indonesia dibagi menjadi 2, yaitu tunai dan non tunai. Pada alat pembayaran tunai, instrumen pembayaran yang digunakan adalah uang kartal. Uang kartal berupa uang kertas dan uang logam.

Uang kertas yang diakui saat ini berjumlah 11 jenis, mulai dari pecahan seribu rupiah, dua ribu rupiah, lima ribu rupiah, sepuluh ribu rupiah, dua puluh ribu rupiah desain lama, dua puluh ribu rupiah desain baru, lima puluh ribu rupiah desain lama, lima puluh ribu rupiah desain baru, serta seratus ribu rupiah desain lama dan seratus ribu rupiah desain baru.



Gambar 2.1 Contoh Uang Kertas

Sumber : <http://www.bi.go.id/web/id/>

Sistem+Pembayaran/Instrumen+Pembayaran+Tunai/Gambar+Uang/

Selain uang kertas, Bank Indonesia juga mengakui 8 jenis uang logam yang berlaku, yaitu pecahan koin satu rupiah, lima puluh rupiah, seratus rupiah, dua ratus rupiah, lima ratus rupiah desain lama, lima ratus rupiah desain baru, seribu rupiah desain lama, dan seribu rupiah desain baru.



Gambar 2.2 Contoh Uang Logam

Sumber : <http://www.bi.go.id/web/id/>

Sistem+Pembayaran/Instrumen+Pembayaran+Tunai/Gambar+Uang/

Sedangkan pada alat pembayaran non tunai, Bank Indonesia mengakui 4 instrumen yang berlaku, yaitu kartu, cek, bilyet giro, dan nota debit.

2.3 Database.

Database adalah sekelompok file yang terkait secara logikal, dan menyimpan data serta asosiasi di antara mereka (Rainer & Turban , 2009: 31). Hirarki database dimulai dari *bit (binary digit)*, yang merepresentasikan unit terkecil dari data yang dapat diproses oleh komputer. Nilai dari *bit* adalah 0 dan 1.

Kumpulan dari 8 *bit* disebut *byte*, dimana *byte* merepresentasikan sebuah karakter tunggal. *Byte* bisa berupa karakter huruf, angka, maupun sebuah simbol. Kumpulan *byte* disebut *field*, dimana *field* dapat berupa kumpulan huruf dan atau

angka yang membentuk sebuah kata, yang juga dapat disertai oleh simbol dan memiliki suatu arti identifikasi tertentu.

Sedangkan *record* adalah kumpulan dari beberapa *field* yang membentuk suatu baris informasi yang saling terhubung, sehingga dapat dilihat sebagai suatu kesatuan. Dan tabel atau *file* merupakan kumpulan dari *record* yang saling terkait. Tabel atau *file* yang saling terhubung inilah yang merupakan bagian penyusun atau kerangka dari sebuah *database*.

Dalam penggunaannya, *database* membutuhkan perintah-perintah yang dapat melakukan pemilihan data yang diinginkan, maupun manipulasi data dan nilai di dalamnya. Perintah-perintah yang dapat melakukan pemilihan data, maupun manipulasi ini disebut *query*.

2.4 Prototype.

Menurut O'Brien dan Marakas (2007 : 374). dalam "*Enterprise Information Systems*", *prototype* adalah sebuah model kerja, khususnya model kerja dari suatu sistem informasi yang mencakup versi tentatif dari masukan dan keluaran pengguna, *database* dan *file*, metode pengendalian, serta pengolahan rutinitas.

Sedangkan Rainer dan Turban (2009 : 307) dalam "*Introduction To Information Systems*", menjelaskan bahwa *prototype* adalah versi kecil yang dibuat secara cepat, berdasarkan sistem yang sedang dikembangkan, dimana pengguna memberikan masukan untuk meningkatkan serta memperbaiki kinerja dari *prototype*.

Berdasarkan definisi-definisi dari para ahli yang dipaparkan sebelumnya, dapat disimpulkan, bahwa *prototype* merupakan seluruh atau sebagian dari sistem yang sedang dikerjakan, serta memuat sebagian dan atau seluruh fungsi-fungsi sistem informasi yang akan dibuat, dimana pengguna berperan untuk memberikan saran dan perbaikan atas model kerja tersebut.

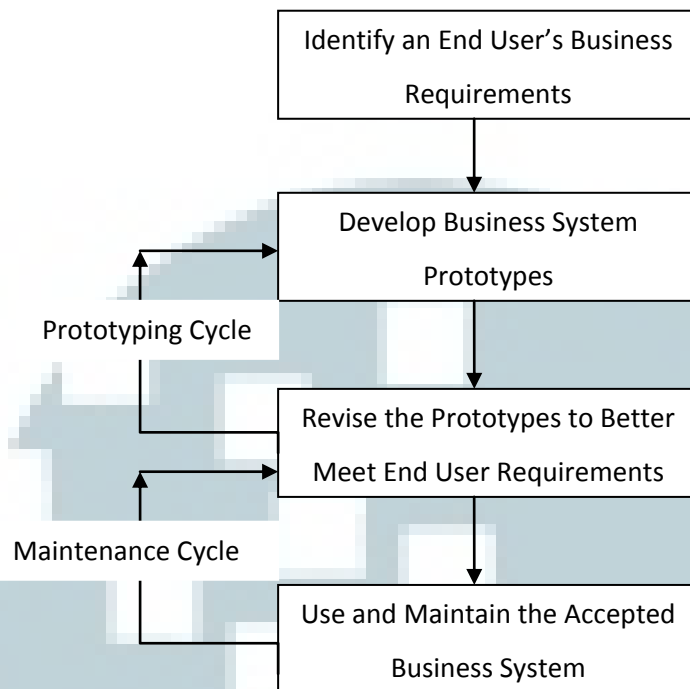
2.4.1. Prototyping.

Dalam pembuatan *prototype*, dikenal suatu metode bernama *prototyping*. *Prototyping* adalah pembuatan dan pengetesan secara cepat dari sebuah model kerja, atau yang disebut juga *prototype*, dari sebuah aplikasi baru, dalam sebuah proses yang interaktif dan iteratif yang dapat digunakan oleh pihak pembuat dan pengguna.

Prototyping berguna untuk membuat proses pembuatan menjadi lebih mudah dan lebih cepat, khususnya untuk proyek yang memiliki kebutuhan akhir pengguna yang sulit untuk didefinisikan.

Menurut O'Brien dan Marakas (2007 : 375), *prototyping* dibagi menjadi 4 tahapan, yaitu :

U
M
M
N



Gambar 2.3 Tahapan *Prototyping*

1. *Identify an End User's Business Requirements*

Pada tahap ini, dilakukan investigasi atau analisa kebutuhan bisnis dan melakukan penilaian terhadap kelayakan dari beberapa alternatif solusi sistem informasi.

2. *Develop Business System Prototypes*

Tahap kedua, dilanjutkan dengan melakukan analisa dan desain oleh pengguna, dan pihak pembuat sistem informasi menggunakan alat-alat pengembangan untuk melakukan desain secara interaktif dan dapat melakukan pengetesan *prototype* untuk memenuhi kebutuhan bisnis pengguna.

3. Revise the Prototypes to Better Meet End User Requirements

Tahapan ini melanjutkan desain, dan memperbaiki *prototype*, secara berulang, setelah dilakukan pengetesan serta evaluasi, sampai pengguna akhir dapat menerima model kerja tersebut.

4. Use and Maintain the Accepted Business System.

Pada tahapan terakhir ini, dilakukan penggunaan sistem yang sudah disetujui, serta dilakukannya perawatan dan pengendalian terhadap kemungkinan kegagalan sistem yang berjalan.

Dalam penelitian ini, penulis akan menggunakan tahapan *prototyping* hingga tahapan ketiga, yaitu *Revise the Prototypes to Better Meet End User Requirement*, karena penulis tidak membahas mengenai implementasi dan pemasangan alat, dan juga pemeliharaan sistem, aplikasi, dan peralatan yang digunakan, sehingga penelitian ini tidak akan sampai pada tahapan ke empat.

2.5 Radio Frequency Identification (RFID)

Radio Frequency Identification atau yang dikenal sebagai RFID, merupakan salah satu teknologi yang banyak dipakai di dunia saat ini. Sejak RFID *tag* dipatenkan oleh Mario W. Cardullo pada 23 Januari 1973, penggunaan RFID semakin melonjak, khususnya pada perusahaan-perusahaan yang mulai beralih ke

otomatisasi dan perusahaan tradisional yang mulai menggunakan teknologi mesin atau robot.

Menurut O'Brien dan Marakas (2007 : 103) dalam "*Enterprise Information Systems*", RFID adalah sistem untuk penandaan dan identifikasi objek bergerak secara otomatis, seperti paket pos, bahkan juga terkadang organism hidup, seperti hewan peliharaan.

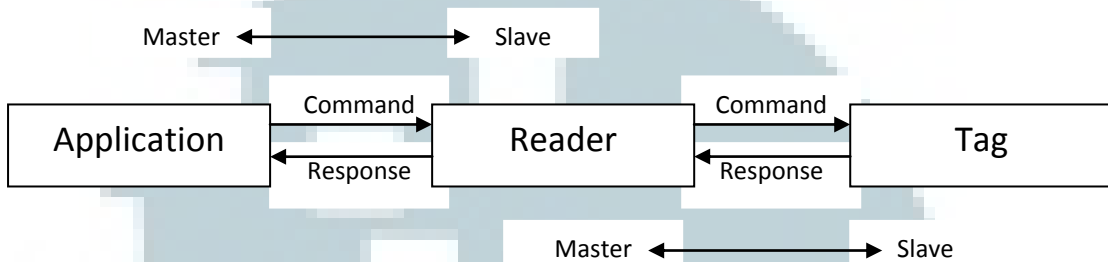
Sedangkan menurut Yan Zhang, Laurence T. Yang, dan Jiming Chen (2010 : 4). pada "*RFID and Sensor Networks*", RFID adalah sebuah teknologi identifikasi otomatis, yang tidak memerlukan garis komunikasi yang terlihat kasat mata.

Sistem RFID terbagi menjadi 2 bagian dasar, yaitu *reader* dan *tag*. *Tag* menyimpan informasi unik dan menempel pada suatu objek. Sedangkan *reader* melakukan prosedur pembacaan atau identifikasi dari suatu *tag* yang menempel pada suatu objek.

Terdapat 2 jenis *tag*, yaitu *tag* aktif dan *tag* pasif. Pada umumnya, *tag* bersifat pasif, dimana, *tag* tidak memiliki sumber daya fisikal, sehingga, *tag* hanya dapat berkomunikasi dengan *reader* pada jarak tertentu, tergantung kekuatan pembacaan dari *reader*. Sedangkan *tag* aktif memiliki sumber daya sendiri untuk melakukan kebutuhan pemrosesannya, dan mungkin memiliki kemampuan pengindraan atau pendeteksian untuk tekanan dan temperatur.

Sistem RFID memiliki suatu arsitektur, yang dinamakan *master-slave architecture*. *Master-slave architecture* menyatakan bahwa RFID *reader* merupakan *slave* dari suatu aplikasi, dan aplikasi merupakan *master* dari RFID

reader, selain itu, arsitektur tersebut juga menyatakan bahwa RFID *tag* merupakan *slave* dari RFID *reader*, dan RFID *reader* adalah *master* dari RFID *tag*.



Gambar 2.4 Master-Slave Architecture
Sumber : Zhang, Yang, Chen (2010 : 4)

Seperti terlihat pada Gambar 2.4, aplikasi memberikan perintah kepada *reader*, untuk melakukan pembacaan, atau identifikasi dari *tag*, dan *reader* akan memberikan perintah kepada *tag* untuk memberikan atau mengambil data dan informasi yang terdapat pada *tag*. Setelah *tag* diperintahkan oleh *reader*, maka *tag* akan memberikan respon berupa informasi atau data yang diminta oleh *reader*, lalu *reader* akan memberikan respon kepada aplikasi, dengan menyampaikan data yang telah diambil dari *tag*, kepada aplikasi.

U
M
M
N

2.6 Microsoft Visual Basic 2008



Gambar 2.5 Microsoft Visual Basic 2008

Sumber : <http://antivirus-softwares.com/2012/05/visual-basic-2008-express-edition-free-downloads/>

Visual Basic adalah sebuah bahasa pemrograman yang dikembangkan oleh Microsoft, yang pada awalnya diciptakan untuk mempermudah *programmer* dalam melakukan penulisan bahasa pemrograman untuk sistem operasi Windows. Tetapi, pada perkembangannya, Visual Basic memungkinkan *programmer* untuk membuat aplikasi pada sistem operasi Windows, Web, dan aplikasi pada perangkat *mobile*.

Menurut Diane Zak (2010 : 3) dalam “*Programming With Microsoft® Visual Basic® 2008*”, Visual Basic 2008 adalah sebuah bahasa pemrograman yang mendukung pemrograman berbasis objek, yang memungkinkan *programmer* dalam menggunakan objek untuk mencapai tujuan suatu program.

Microsoft Visual Basic 2008 dapat berupa aplikasi produk yang berdiri sendiri (*stand alone application*), dengan nama Microsoft Visual Basic 2008 Express Edition, maupun menempel sebagai bagian dari Microsoft Visual Studio 2008.


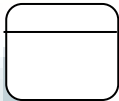






2.7 Data Flow Diagram

Pemodelan proses secara grafis melibatkan tentang perwakilan suatu fungsi, atau proses yang menangkap, memanipulasi, menyimpan, dan mendistribusikan data antara sistem dan lingkungan sistem itu sendiri serta antar komponen di dalam sistem. Dalam proses inilah, *Data Flow Diagram* digunakan.

Menurut Hoffer, George, dan Valacich (2008: 206-207) dalam *Modern Systems Analysis and Design*, *data flow diagram* adalah sebuah gambaran dari pergerakan data antara entitas eksternal dan proses dan *data stores* di dalam sebuah sistem.

Terdapat 4 buah simbol dalam *data flow diagram*, dimana terdapat 2 buah versi berbeda, menurut DeMarco dan Yourdon, serta menurut Gane dan Sarson, yaitu :

Tabel 2.2. Simbol-Simbol Dalam *Data Flow Diagram*

DeMarco dan Yourdon	Simbol	Gane dan Sarson
	Proses	
	<i>Data store</i>	
	<i>Source/sink</i>	
	<i>Data flow</i>	

Pada pembahasan ini, penulis menggunakan simbol berdasarkan versi DeMarco dan Yourdon.

Terdapat 26 aturan yang tidak boleh dilanggar dalam pembuatan *data flow diagram*, dimana ke-26 aturan tersebut terbagi menjadi 4 sub pokok, yaitu :

Tabel 2.3. Aturan dan Larangan Pada *Data Flow Diagram*

<i>Process</i>
1. Tidak boleh ada proses yang hanya memiliki <i>output</i>
2. Tidak boleh ada proses yang hanya memiliki <i>input (black hole)</i>
3. Sebuah proses harus memiliki frasa kata kerja dalam penamaannya.
<i>Data Store</i>
1. Data tidak dapat berpindah secara langsung dari satu <i>data store</i> ke <i>data store</i> lainnya. Data harus berpindah melalui sebuah proses.
2. Data tidak dapat berpindah langsung dari sebuah <i>source</i> eksternal menuju sebuah <i>data store</i> . <i>Source</i> eksternal harus memasukkan input data ke dalam proses, dan proses akan melanjutkan perpindahan data ke <i>data store</i> .
3. Data tidak dapat berpindah langsung dari <i>data store</i> ke <i>sink/ source</i> . Data harus berpindah melalui proses, dan proses akan meneruskan data berupa <i>output</i> kepada <i>source / sink</i> .
4. sebuah data store harus dinamakan menggunakan frasa kata benda
<i>Source/Sink</i>
1. Data tidak dapat berpindah langsung dari <i>source</i> ke <i>source</i> , atau dari <i>sink</i> ke <i>sink</i> , atau dari <i>source</i> ke <i>sink</i> dan sebaliknya.
2. Sebuah <i>source/sink</i> harus dinamakan menggunakan frasa kata benda.

Data Flow

1. Sebuah *data flow* hanya boleh memiliki 1 arah tujuan ke simbol-simbol yang ada
2. Sebuah aliran yang berbentuk garpu, atau cabang di *data flow* berarti bahwa data yang identik berasal dari satu lokasi yang sama, menuju dua atau lebih proses yang berbeda.
3. Penggabungan beberapa *data flow* berarti ada lebih dari satu data yang identik berasal dari dua atau lebih proses yang berbeda, data stores, atau sources/sinks menuju lokasi yang sama
4. Sebuah *data flow* tidak dapat bergerak kembali ke proses awal yang sama secara langsung. Harus ada minimal satu proses lain yang disinggahi, baru dapat kembali ke proses awal tempat *data flow* tersebut berasal.
5. Sebuah *data flow* menuju *data stores* berarti adanya perubahan data.
6. Sebuah *data flow* dari *data store* berarti sedang digunakan atau sedang di akses.
7. Sebuah *data flow* memiliki penamaan berupa frasa kata benda.

2.8 Entity Relationship Diagram

Setiap aplikasi yang dirancang, pasti memiliki hubungan dengan entitas-entitas akan menggunakan aplikasi. Setiap entitas pasti memiliki hubungan satu dengan yang lainnya, entah itu hubungan secara langsung, maupun tidak langsung.

Menurut Hoffer, George, dan Valacich dalam *Modern Systems Analysis and Design*, entitas adalah orang, tempat, objek, even atau konsep yang tercakup dalam lingkungan pengguna, dimana organisasi berharap agar dapat dilakukan pengaturan data organisasi (2008:290). Sedangkan, *Entity Relationship Model* adalah sebuah representasi logis secara rinci dari data untuk sebuah organisasi ataupun untuk area bisnis tertentu. (2008:290). *Entity Relationship Diagram* sendiri adalah representasi grafis dari sebuah *Entity Relationship Model* (2008:290).

Sedangkan menurut Rainer dan Turban dalam *Introduction to Information Systems* edisi ke dua, *Entity Relationship Diagram* adalah dokumen yang menunjukkan data entitas dan atribut, serta relasi diantara keduanya (2009:109).

Tabel 2.4. Contoh-contoh Entitas

Entitas	Contoh Entitas
Orang (<i>Person</i>)	EMPLOYEE, STUDENT, PATIENT
Tempat (<i>Place</i>)	STORE, WAREHOUSE
Objek (<i>Object</i>)	MACHINE, PRODUCT, BUILDING
Even (<i>Event</i>)	SALE, REGISTRATION, RENEWAL
Konsep (<i>Concept</i>)	ACCOUNT, COURSE, WORK CENTER