



**UNIVERSITAS INDONESIA**

**IMPLEMENTASI DAN EVALUASI KINERJA SISTEM  
INFORMASI SIMULASI PERPARKIRAN BERBASIS WEB  
DAN RFID MENGGUNAKAN ANTARMUKA JAVA DAN JSP  
DENGAN BASIS DATA MYSQL**

**SKRIPSI**

**EKI RAHMADIAN  
04 05 03 0311**

**FAKULTAS TEKNIK  
PROGRAM STUDI TEKNIK ELEKTRO  
DEPOK  
DESEMBER 2009**



**UNIVERSITAS INDONESIA**

**IMPLEMENTASI DAN EVALUASI KINERJA SISTEM  
INFORMASI SIMULASI PERPARKIRAN BERBASIS WEB  
DAN RFID MENGGUNAKAN ANTARMUKA JAVA DAN JSP  
DENGAN BASIS DATA MYSQL**

**SKRIPSI**

**Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik**

**EKI RAHMADIAN  
04 05 03 0311**

**FAKULTAS TEKNIK  
PROGRAM STUDI TEKNIK ELEKTRO  
DEPOK  
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 : Eki Rahmadian**

**NPM : 0405030311**

**Tanda Tangan : **

**Tanggal : 30 Desember 2009**

## HALAMAN PENGESAHAN

Skripsi ini diajukan oleh :  
Nama : Eki Rahmadian  
NPM : 0405030311  
Program Studi : Teknik Elektro  
Judul Skripsi : Implementasi dan Evaluasi Kinerja Sistem Informasi Simulasi Perparkiran Berbasis Web dan RFID Menggunakan Antarmuka Java dan JSP dengan Basis Data MySQL

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

### DEWAN PENGUJI

Pembimbing : Prof. Dr. Ir. Kalamullah Ramli, M.Eng (Kalamullah)

Penguji : Prima Dewi Purnamasari, ST, MT, M.Sc (Prima Dewi Purnamasari)

Penguji : Prof. Dr. Ir. Bagio Budiardjo, MSc (Bagio Budiardjo)

Ditetapkan di : Depok

Tanggal : 30 Desember 2009

## KATA PENGANTAR/UCAPAN TERIMA KASIH

Puji syukur saya panjatkan kepada Allah SWT atas segala rahmat dan hidayah-Nya sehingga saya dapat menyelesaikan skripsi ini. Penulisan skripsi ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gelar Sarjana Teknik. Saya menyadari bahwa, tanpa bantuan dan bimbingan dari berbagai pihak, dari masa perkuliahan sampai pada penyusunan skripsi ini, tidaklah mudah bagi saya untuk menyelesaikan skripsi ini. Oleh karena itu, saya mengucapkan terima kasih kepada :

- [1] **Prof. Dr. Ir. Kalamullah Ramli, M.Eng**, selaku dosen pembimbing yang telah menyediakan waktu, tenaga, dan pikiran untuk mengarahkan saya dalam penyusunan skripsi ini;
- [2] Lembaga pendidikan **LP2M Aray** yang telah banyak membantu dalam teknis pembuatan program;
- [3] Orang tua dan keluarga saya yang telah memberikan bantuan dukungan material dan moral; dan
- [4] Kakak saya **Melinda Savitri**, kemudian **Cindyta Septiana** dan **Manyang Panjerrino** yang telah memberikan dukungan dan motivasinya selama penyusunan skripsi ini.

Akhir kata, semoga Allah SWT melimpahkan rahmat kepada semua pihak yang telah membantu yang tidak bisa saya sebutkan satu per satu di sini. Semoga skripsi ini membawa manfaat bagi pengembangan ilmu.

Depok, Desember 2009

Penulis

**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS  
AKHIR UNTUK KEPENTINGAN AKADEMIS**

---

Sebagai sivitas akademik Universitas Indonesia, saya yang bertanda tangan di bawah ini:

Nama : Eki Rahmadian  
NPM : 0405030311  
Program Studi : Teknik Elektro  
Departemen : Teknik Elektro  
Fakultas : Teknik  
Jenis Karya : Skripsi

demikian 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 :

IMPLEMENTASI DAN EVALUASI KINERJA SISTEM INFORMASI DAN SIMULASI PERPARKIRAN BERBASIS WEB DAN RFID MENGGUNAKAN ANTARMUKA JAVA DAN JSP DENGAN BASIS DATA MYSQL

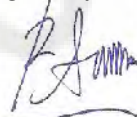
beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Universitas Indonesia berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (database), merawat, dan memublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok

Pada Tanggal : 30 Desember 2009

Yang Menyatakan



(Eki Rahmadian)

## ABSTRAK

Nama : Eki Rahmadian  
Program Studi : Teknik Elektro  
Judul : IMPLEMENTASI DAN EVALUASI KINERJA SISTEM  
INFORMASI SIMULASI PERPARKIRAN BERBASIS WEB  
DAN RFID MENGGUNAKAN ANTARMUKA JAVA DAN  
JSP DENGAN BASISDATA MYSQL

Skripsi ini membahas tentang perancangan sistem informasi untuk diaplikasikan pada sistem perpajakan. Sistem informasi ini terhubung dengan alat-alat tambahan, yaitu modul *Radio Frequency Identification (RFID)* dan kamera web. RFID berperan sebagai tanda pengenalan otomatis, dan kamera web dipergunakan untuk sistem pengawasan sederhana.

Pada sistem ini, *RFID reader* sebagai terminal pembaca *RFID tag* melakukan komunikasi secara *serial* dengan *PC*, dengan standar *RS-232*. Dengan menggunakan *Java API* untuk komunikasi serial, maka *RFID* terhubung dengan *PC*. Selanjutnya dengan *Java Database Connectivity (JDBC)*, aplikasi berbasis bahasa pemrograman *Java* tersebut dapat terhubung ke sebuah sistem basis data, yang dalam skripsi ini digunakan basis data *MySQL*.

Untuk menampilkan informasi-informasi yang dibutuhkan oleh pengguna dan *operator* parkir, dipergunakan antarmuka web yang dibangun dengan bahasa pemrograman web dinamis *Java Server Pages (JSP)*. Sistem informasi ini diharapkan mampu memberikan informasi yang cukup bagi setiap individu yang mengaksesnya, serta dapat membantu operator dalam hal pendataan dan keamanan.

Dari sistem yang telah dibuat, dilakukan tiga jenis pengujian. Pengujian ini dilakukan pada perangkat keras dan perangkat lunak yang digunakan. Pengujian fungsional sistem menunjukkan bahwa modul RFID mampu membaca tag kartu dengan baik, kemudian modul kamera membutuhkan jeda waktu minimal 0,5 detik untuk mengambil foto berdasarkan pergerakan objek, dan semakin besar objeknya, semakin sensitif kamera dalam mendeteksi pergerakan objek.

Pengujian kompatibilitas menunjukkan bahwa dari kelima web browser yang digunakan, dua di antaranya kurang kompatibel dengan sistem informasi yang dibuat. Selain itu dari pengujian performa halaman web diperoleh rata-rata waktu yang dibutuhkan untuk melakukan *load* setiap halamannya, yaitu berkisar pada 6000 sampai 18000 mikrosekon bergantung pada browser yang digunakan.

Kata Kunci:  
Sistem informasi parkir, *Java*, *RFID*, *MySQL*, *JSP*

## ABSTRACT

Name : Eki Rahmadian  
Study Program : Electrical Engineering  
Title : IMPLEMENTATION AND PERFORMANCE EVALUATION  
OF WEB BASED PARKING SIMULATION INFORMATION  
SYSTEM USING JAVA AND JSP INTERFACE WITH  
MYSQL DATABASE

This final assignment discusses the design of an information system to be applied in parking system. This information system is connected with several devices including Radio Frequency Identification (RFID) module and web cameras. The RFID module is used as an automatic identification whereas the web cameras are used for simple surveillance system.

The RFID reader which is used as the reader for RFID tag communicates with the PC using the serial port with an RS-232 standard. Using Java API for serial communication the RFID is connected to PC. Furthermore, Java Database Connectivity (JDBC) connects application to the database.

To deliver informations which are needed by its user, a web interface built with dynamic web programming language -Java Server Pages (JSP) in this case- is developed. This information system is expected to deliver any kind of information required by anyone who is accessing it, and could help the operators and users in terms of physical security.

Three kinds of tests have been performed on hardware and software part of the system. System's functionality tests showed that the RFID module run well and read the tag properly. The cameras need minimum 0.5 seconds to capture photos based on motion detection. The bigger the object, the more sensitive the camera in detecting the motion of object.

System's compatibility tests show that among five web browsers, two of them are not compatible with the information system. System's performance tests show that the average time which is needed to load every page is ranging from 6000 to 18000 microseconds subject the type of the browsers.

Keywords:

Parking information system, *Java*, *RFID*, *MySQL*, *JSP*



## DAFTAR ISI

HALAMAN JUDUL.....	i
HALAMAN PERNYATAAN ORISINALITAS.....	ii
HALAMAN PENGESAHAN.....	iii
UCAPAN TERIMA KASIH.....	iv
HALAMAN PERSETUJUAN PUBLIKASI SKRIPSI.....	v
ABSTRAK.....	vi
DAFTAR ISI.....	vii
DAFTAR TABEL.....	x
DAFTAR GAMBAR.....	xi
<b>1. PENDAHULUAN.....</b>	<b>1</b>
1.1 Latar Belakang Skripsi.....	1
1.2 Perumusan Masalah.....	2
1.3 Batasan Masalah.....	2
1.4 Sistematika Penulisan Skripsi.....	3
<b>2. TEORI PENDUKUNG.....</b>	<b>5</b>
2.1 Sistem Informasi.....	5
2.2 Entity Relationship Diagram (ERD).....	7
2.3 Unified Modelling Language (UML).....	9
2.3.1 Use Case Diagram.....	11
2.3.2 Sequence Diagram.....	12
2.3.3 Activity Diagram.....	13
2.4 Web.....	13
2.4.1 Web Database.....	15
2.5 Hyper Text Markup Language(HTML).....	15
2.6 Java Server Pages (JSP).....	16
2.6.1 Keuntungan JSP bagi Pengembang.....	17
2.6.2 Teknologi JSP dan Java Servlets.....	17
2.6.3 Latar Belakang Komunitas.....	18
2.7 MySQL.....	19
2.7.1 Relational Database Management System (RDBMS).....	19
2.4 Surveillance Camera.....	22
2.4 Radio Frequency Identification (RFID).....	23
<b>3. PEMODELAN SISTEM DAN RENCANA UJI COBA.....</b>	<b>24</b>
3.1 Skenario.....	24
3.2 Arsitektur Sistem Secara Global.....	26
3.3 Perancangan Perangkat Keras.....	29
3.3.1 Peralatan.....	29

3.3.2 Konfigurasi.....	31
3.4 Perancangan Aplikasi Sistem Informasi Perparkiran Berbasis Web.....	32
3.5 Database .....	33
3.6 Permodelan Aplikasi Sistem Informasi Perparkiran Berbasis Web .....	34
3.6.1 Use Case Diagram.....	35
3.6.2 Activity Diagram.....	37
3.6.3 Sequence Diagram.....	40
3.7 Desain Tampilan User Interface.....	44
3.8 Rencana Uji Coba Dan Analisa Kinerja Sistem.....	45
<b>4. PENGUJIAN DAN EVALUASI KINERJA SISTEM.....</b>	<b>46</b>
4.1 Pengujian Fungsional Sistem .....	46
4.1.1 Pengujian Fungsional Dasar dari Kamera.....	46
4.1.2 Pengujian Sensitivitas dan Delay Pengambilan Foto oleh Kamera .	47
4.1.2.1 Pengujian Sensitivitas Kamera .....	47
4.1.2.2 Uji Delay Pengambilan Foto .....	51
4.1.3 Pengujian Fungsionalitas RFID Reader dan RFID Tag .....	52
4.1.4 Pengujian Fungsionalitas Dasar Perangkat Lunak .....	54
4.2 Pengujian Kompatibilitas dan Performa Halaman Web.....	54
4.3 Perbandingan dengan Sistem RFID yang Telah Ada .....	61
<b>5. KESIMPULAN.....</b>	<b>64</b>
<b>DAFTAR ACUAN.....</b>	<b>65</b>
<b>DAFTAR PUSTAKA .....</b>	<b>66</b>
<b>LAMPIRAN.....</b>	<b>67</b>

## DAFTAR TABEL

Tabel 4.1 Pengujian Fungsi Dasar Kamera.....	46
Tabel 4.2 Pengujian Sensitivitas Kamera A Arah Horizontal .....	48
Tabel 4.3 Pengujian Sensitivitas Kamera B Arah Horizontal.....	48
Tabel 4.4 Pengujian Sensitivitas Kamera C Arah Horizontal.....	49
Tabel 4.5 Pengujian Sensitivitas Kamera A Arah Vertikal .....	49
Tabel 4.6 Pengujian Sensitivitas Kamera B Arah Vertikal.....	50
Tabel 4.7 Pengujian Sensitivitas Kamera C Arah Vertikal.....	50
Tabel 4.8 Uji Delay Pengambilan Foto.....	51
Tabel 4.9 Uji Fungsionalitas RFID Reader dan RFID Tag.....	52
Tabel 4.10 Pengujian Fungsionalitas Dasar Perangkat Lunak.....	54
Tabel 4.11 Pengujian Sistem pada Mozilla Firefox .....	56
Tabel 4.12 Pengujian Sistem pada Internet Explorer.....	57
Tabel 4.13 Pengujian Sistem pada Opera .....	58
Tabel 4.14 Pengujian Sistem pada Safari.....	59
Tabel 4.15 Pengujian Sistem pada Google Chrome.....	60
Tabel 4.16 Perbandingan dengan Sistem RFID yang Telah Ada.....	62

## DAFTAR GAMBAR

Gambar 2.1 Entitas Kendaraan .....	7
Gambar 2.2 One to one relationship .....	8
Gambar 2.3 One to many relationship .....	8
Gambar 2.4 Many to many relationship.....	9
Gambar 2.5 Sembilan Diagram UML.....	10
Gambar 2.6 Use Case Diagram.....	11
Gambar 2.7 Sequence Diagram.....	12
Gambar 2.8 Activity Diagram.....	13
Gambar 3.1 Denah Lahan Parkir.....	24
Gambar 3.2 Arsitektur sistem secara global .....	27
Gambar 3.3 Rencana perancangan sistem.....	28
Gambar 3.4 RFID Starter-Kit.....	29
Gambar 3.5 Tag RFID Pasif Class-1 berbentuk kartu .....	30
Gambar 3.6 Kamera Web.....	30
Gambar 3.7 Setting Jumper Pada RFID Starter-Kit.....	31
Gambar 3.8 Aplikasi Kamera Web Yawcam.....	32
Gambar 3.9 Diagram database .....	33
Gambar 3.10 Use Case Diagram Sistem Informasi Perparkiran.....	35
Gambar 3.11 Activity Diagram Untuk Capture Foto.....	37
Gambar 3.12 Activity Diagram Slot Parkir dan Log Kendaraan.....	38
Gambar 3.13 Activity Pencarian Letak Kendaraan.....	39
Gambar 3.14 Sequence Diagram Registrasi dan Input Kendaraan.....	41
Gambar 3.15 Sequence Diagram Letak Kendaraan .....	42
Gambar 3.16 Sequence Diagram Arsip Foto, Log dan Denah Parkir.....	43
Gambar 3.17 Desain Halaman Depan User Interface .....	44
Gambar 4.1 Output Pembacaan RFID Tag pada Hyper Terminal.....	53
Gambar 4.2 Output Pembacaan RFID Tag pada Program Desktop.....	53

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang Skripsi

Sistem parkir dalam suatu instansi, institusi ataupun pusat perbelanjaan merupakan bagian yang dipandang cukup penting dalam menunjang aktivitas pada masing-masing lokasi tersebut, khususnya bagi para pengguna kendaraan bermotor. Kemudahan dalam proses pemasukan data menjadi tolak ukur yang penting dalam proses transaksi parkir, yang tentunya akan membantu dalam hal penghematan waktu, biaya dan tenaga. Dewasa ini, kita telah mengenal suatu teknologi pengenalan suatu objek yang memanfaatkan frekuensi radio dalam pembacaannya, teknologi ini selanjutnya kita sebut RFID atau *Radio Frequency Identification*. Teknologi ini dapat diimplementasikan untuk kemudahan dalam sistem parkir.

Dengan sistem RFID, proses manipulasi data dari kendaraan yang akan masuk ataupun keluar dari lahan parkir menjadi lebih mudah dilakukan karena prosesnya yang lebih sederhana, yaitu hanya dengan mendekatkan tag atau penanda atau pengenal ke suatu *reader* dalam jangkauan tertentu, data dari kendaraan di dalam database sistem parkir ini akan secara otomatis berubah.

Kemudian bagi petugas parkir dan pengguna parkir, tentunya akan lebih baik jika disediakan juga suatu sistem informasi parkir pada sistem parkir ini. Sebuah sistem informasi harus dibuat seinformatif mungkin sehingga pengguna yang melakukan akses dapat dengan mudah mengetahui informasi yang dicari. Informasi seperti berapa banyak kendaraan yang telah masuk, sistem pengawasan dan pemantauan keadaan lokasi parkir, kemudian peta lokasi parkir dirasa cukup penting untuk diinformasikan kepada pengguna lahan parkir dan petugas parkir. Bagi petugas parkir, informasi keadaan lokasi parkir akan memudahkan petugas untuk memantau lokasi parkir, sehingga dapat mencegah tindak kejahatan khususnya pencurian, lebih jauh lagi, apabila terjadi

tindak kejahatan pencurian, informasi ini akan membantu proses pelacakan pelaku kejahatan, karena adanya rekaman kejadian-kejadian pada selang waktu tertentu. Informasi kendaraan yang telah masuk akan memudahkan petugas parkir untuk memberi tahu pengguna lahan parkir yang baru tiba apakah masih ada lahan yang masih kosong untuk bisa di isi lagi dengan kendaraan pengguna. Kemudian peta lokasi parkir akan memudahkan pengguna parkir mengingat di lokasi mana pengguna parkir tersebut meletakkan kendaraannya.

### **1.2 Perumusan Masalah**

Pada penyambungan dan pembentukkan komunikasi antara peripheral atau alat – alat perangkat luar di luar komputer (PC) dengan PC terdapat banyak pilihan, dimana salah satunya adalah menggunakan komunikasi serial melalui port COM pada PC. Pemilihan cara komunikasi tersebut harus tepat guna agar tidak menimbulkan masalah di kemudian harinya.

Untuk menjembatani antara pengguna dan operator parkir dengan sistem informasi dan divais-divais yang terhubung dengan PC dibutuhkan suatu antarmuka, antarmuka yang bisa dipergunakan salah satunya yaitu antarmuka web (*web interface*). Antarmuka web ini dapat dibangun dengan bahasa pemrograman web HTML (*Hypertext Markup Language*) dan JSP (*Java Server Pages*) yang sudah mendukung *Multiplatform*.

Sedangkan untuk melakukan penyimpanan data – data parkir, dibutuhkan suatu sistem basis data. Basis data ini mau tidak mau harus tergabung dengan antarmuka web yang dibangun. Dari sini timbul masalah baru lagi tentang bagaimana menghubungkan program berupa antarmuka web berbasis JSP dengan basis data yang digunakan.

### **1.3 Batasan Masalah**

Laporan skripsi yang penulis buat terbatas pada :

1. Sistem informasi dengan antarmuka web yang dibangun dengan HTML dan JSP.
2. Hubungan antara antarmuka web dengan RFID *receiver (reader)* dan RFID transponder (RFID *tag*).
3. Komunikasi antara RFID receiver dengan PC hanya menggunakan sistem komunikasi serial dengan standar arsitektur RS232.
4. Proses penyimpanan foto oleh kamera dan pemasukan foto ke dalam antarmuka web.
5. Keterkaitan antara Sistem RFID dengan Kamera dalam basis data.
6. Proses pengendalian dan komunikasi antara PC dengan RFID receiver menggunakan bahasa pemrograman JSP.
7. Penggunaan sistem basis data MySQL, dengan konfigurasi-nya menggunakan PHPMyAdmin.
8. Koneksi dengan basis data MySQL menggunakan bahasa pemrograman JSP dengan JDBC (*Java Database Connectivity*)

#### **1.4 Sistematika Penulisan Skripsi**

Bab I adalah pendahuluan, yang menjelaskan mengenai latar belakang dibuatnya skripsi ini, perumusan masalah yang akan dihadapi dalam pembuatan skripsi dengan tema yang diambil, tujuan dibuatnya skripsi ini, batasan masalah yang akan dibahas pada skripsi ini, dan sistematika penulisan laporan Skripsi.

Bab II adalah landasan teori yang menjelaskan tentang sistem informasi parkir, RFID, sistem pengawasan dan teori pendukung dalam membangun sistem informasi seperti UML (*Unified Modelling Language*), *Database*, *Web programming* dan sebagainya.

Bab III adalah penjelasan tentang skema dan permodelan sistem informasi dan pengawasan sederhana berbasis web dan RFID menggunakan antarmuka web JSP, dan sistem basis data MySQL. Pada bab ini juga dibahas pemodelan dan

konsep sistem informasi parkir, pembagian sistem – sistem yang digunakan, alur kerja sistem, rencana implementasi dan rencana uji coba.

Bab IV adalah penjelasan rangkaian implementasi dan uji coba skripsi. Pada bab ini juga terdapat analisa dan pembahasan atas data – data yang telah di uji cobakan.

Bab V adalah penutup dan kesimpulan dari perancangan dan implementasi skripsi ini dan peluang penerapan teknologi ini di masa depan.



## BAB II TEORI PENDUKUNG

### 2.1 Sistem Informasi

Sistem Informasi Terdiri dari dua kata yaitu Sistem dan Informasi. Sistem sendiri berarti gabungan dari beberapa sub sistem yang bertujuan untuk mencapai satu tujuan. Informasi berarti sesuatu yang mudah dipahami oleh si penerima. Sistem Informasi memiliki makna sistem yang bertujuan menampilkan informasi. Pada jaman dahulu sebelum sistem komputer ada maka sistem informasi ini telah lebih dahulu ada dan berjalan dengan baik.

Syarat-syarat dari sistem informasi yang baik dapat diuraikan sebagai berikut [1]:

1. Ketersediaan

Tentunya ini merupakan syarat yang mendasar bagi suatu informasi, yaitu tersedianya informasi itu sendiri. Informasi harus dapat diperoleh bagi orang yang ingin memanfaatkannya.

2. Mudah dipahami

Informasi harus mudah dipahami oleh para pengguna

3. Relevan

Informasi yang ditampilkan benar-benar relevan dengan permasalahan, misi, dan tujuan.

4. Bermanfaat

Sebagai konsekuensi dari syarat relevansi, informasi juga harus bermanfaat bagi perseorangan maupun organisasi

5. Keandalan

Informasi harus diperoleh dari sumber-sumber yang dapat diandalkan kebenarannya. Pengolah data atau pemberi informasi harus dapat menjamin tingkat kepercayaan yang tinggi atas informasi yang disajikan.

6. Akurat

Syarat ini mengharuskan bahwa informasi yang disampaikan harus bersih dari kesalahan dan kekeliruan. Ini juga berarti bahwa informasi harus jelas dan secara akurat mencerminkan makna yang terkandung dari data pendukungnya.

Untuk mewujudkan hal tersebut digunakan komputer sebagai komponen yang sangat penting di dalam proses tersebut. Alasan pemakaian computer sebagai alat bantu adalah karena kemampuan komputer untuk mengolah informasi.

Sistem informasi berbasis komputer mengandung unsur-unsur sebagai berikut [1]:

1. Manusia

Setiap informasi berbasis komputer harus memperhatikan unsur manusia karena merupakan penentu dari keberhasilan serta manusialah yang pada akhirnya akan memanfaatkan informasi yang dihasilkan.

2. Perangkat keras

Perangkat keras terdiri dari komputer itu sendiri beserta semua perangkat pendukungnya.

3. Perangkat lunak

Perangkat lunak terdiri dari program-program komputer beserta petunjuk-petunjuk pendukungnya.

4. Data

Yaitu fakta-fakta yang akan dibuat menjadi informasi yang bermanfaat. Data inilah yang akan dipilah, dimodifikasi, atau diperbarui oleh program-program agar dapat dihasilkan informasi yang akurat.

5. Prosedur

Peraturan-peraturan yang digunakan untuk menentukan operasi sistem komputer sehingga dapat berjalan dengan baik.

## 2.2 ENTITY RELATIONSHIP DIAGRAM (ERD)

*Entity Relationship Diagram* (ERD) adalah diagram yang digunakan untuk mendokumentasikan data yang direpresentasikan dalam bentuk :

- *Entity*

Entity dapat berupa *environmental element* (elemen di sekitar sistem yang berhubungan dengan sistem tersebut) atau berupa *resource* (sumber daya yang berhubungan dengan sistem yang ada). Contoh *entity* misalnya kendaraan. *Entity* digambarkan dengan sebuah kotak persegi empat dengan sebuah nama *entity* berupa kata benda tunggal di dalamnya, seperti terlihat pada gambar berikut :

Kendaraan	
PK	<u>NomorKendaraan</u>
	JenisKendaraan

**Gambar 2.1 – Entitas Kendaraan**

- *Attribute*

*Attribute* suatu *entity* adalah karakteristik *entity* tersebut. Misalnya *entity* kendaraan memiliki *attribute* NomorKendaraan dan JenisKendaraan.

- *Identifier dan Primary Key*

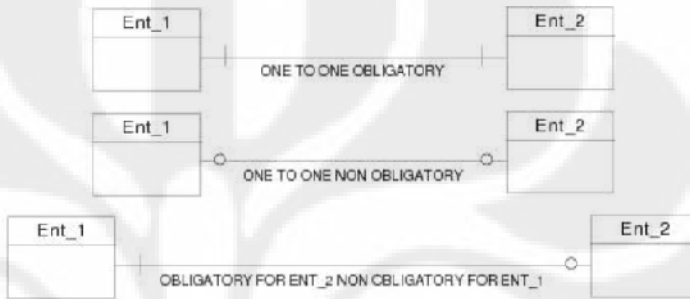
Merupakan *Attribute* yang mengidentifikasi suatu *entity* secara unit dan tidak boleh kembar. Misalnya pada *entity* Kendaraan, NomorKendaraan merupakan *Primary Key*-nya.

- *Relationship*

Merupakan hubungan yang terjadi antara dua buah *entity* atau lebih. *Cardinality* dan *relationship* dipakai untuk mengidentifikasi struktur dari *relationship*, yaitu :

- *One to One Relationship* (1:1)

Ini berarti entity\_1 berhubungan dengan entity\_2, demikian sebaliknya. Hubungan ini dibedakan menjadi dua, yaitu *obligatory* dan *non-obligatory*. *Obligatory* bila anggota dari suatu entity harus berpartisipasi atau berhubungan dengan entity lain. Non Obligatory bila semua anggota entity tidak harus mempunyai hubungan dengan anggota entity yang lain. Notasi one to one dapat dilihat pada gambar berikut.



**Gambar 2.2 – One to one relationship**

o *One to Many Relationship (1:M)*

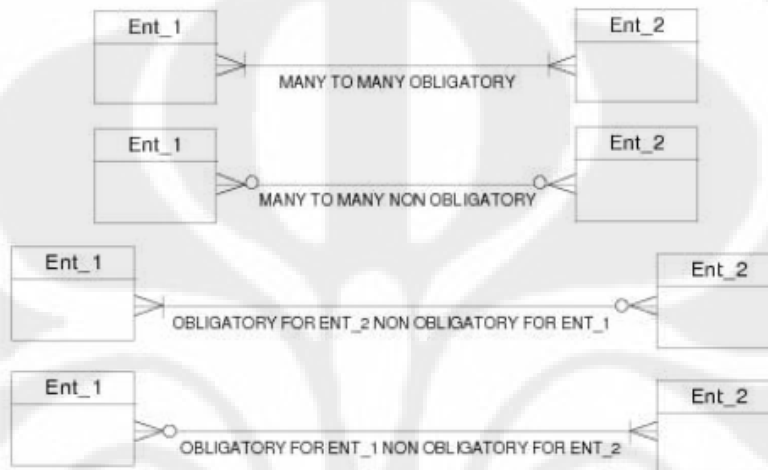
Berarti anggota *entity* dapat mempunyai hubungan dengan beberapa anggota *entity* lain. Notasi ini dapat dilihat pada gambar berikut :



**Gambar 2.3 – One to many relationship**

- o *Many to Many Relationship (M:N)*

Berarti beberapa anggota *entity* lain mempunyai hubungan dengan *entity* lain yang berbeda. Jadi kedua belah pihak dapat memiliki hubungan dengan lebih dari satu anggota *entity* yang lain. Notasi ini dapat dilihat pada gambar berikut :



**Gambar 2.4 – Many to many relationship**

### 2.3 UNIFIED MODELLING LANGUAGE (UML)

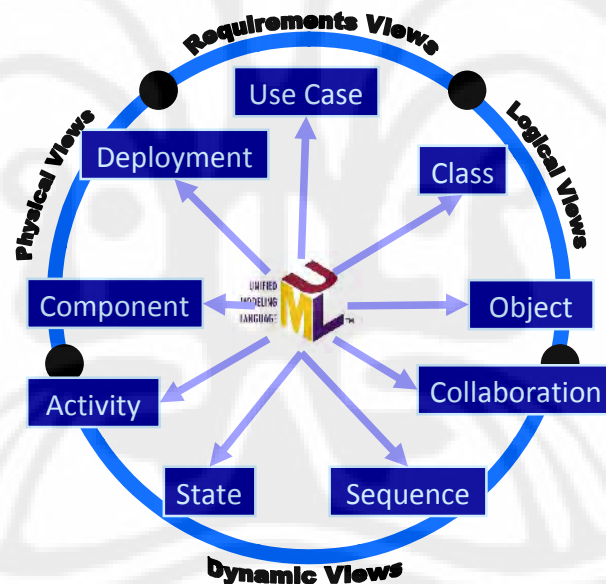
Pendesainan sistem dapat dilakukan dengan menggunakan UML. *Unified Modelling Language* (UML) adalah sebuah "bahasa" yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak.

Dengan menggunakan UML kita dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Tetapi karena UML juga menggunakan *class* dan *operation* dalam konsep dasarnya, maka ia lebih cocok untuk penulisan piranti lunak dalam bahasa - bahasa berorientasi objek seperti C++, Java, C# atau VB.NET. Walaupun

demikian, UML tetap dapat digunakan untuk pemodelan aplikasi prosedural dalam VB atau C.

Seperti bahasa-bahasa lainnya, UML mendefinisikan notasi dan *syntax* / semantik. Notasi UML merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram piranti lunak. Setiap bentuk memiliki makna tertentu, dan UML *syntax* mendefinisikan bagaimana bentuk-bentuk tersebut dapat dikombinasikan. Notasi UML terutama diturunkan dari 3 notasi yang telah ada sebelumnya: Grady Booch OOD (*Object-Oriented Design*), Jim Rumbaugh OMT (*Object Modeling Technique*), dan Ivar Jacobson OOSE (*Object-Oriented Software Engineering*).

Dari berbagai penjelasan rumit yang terdapat di dokumen dan buku-buku UML. Sebenarnya konsepsi dasar UML bisa kita rangkumkan dalam gambar berikut.

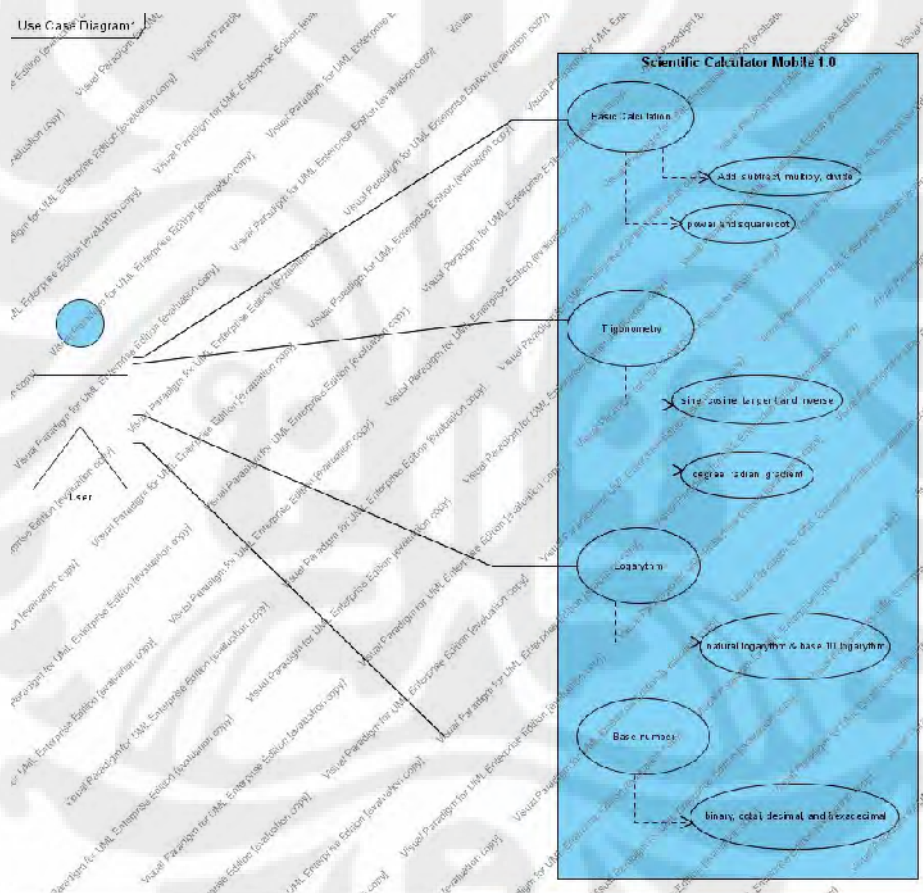


**Gambar 2.5 – Sembilan Diagram UML**

Dari sembilan diagram tersebut, tiga diagram cukup mudah dipahami, di antaranya *Use Case*, *Activity*, dan *Sequence Diagram*.

### 2.3.1 Use Case Diagram

*Use case diagram* menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use case* merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, meng-*create* sebuah daftar belanja, dan sebagainya. Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan system untuk melakukan pekerjaan-pekerjaan tertentu [5].



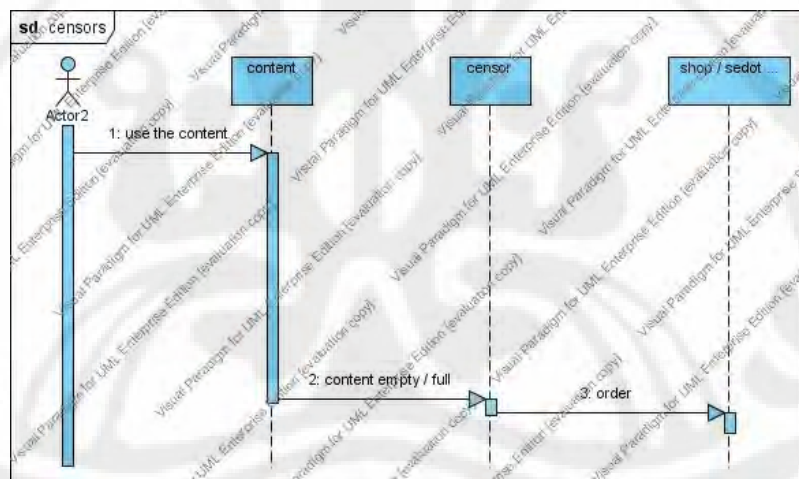
Gambar 2.6 – Use Case Diagram

### 2.3.2 Sequence Diagram

*Sequence diagram* menggambarkan interaksi antar objek di dalam dan di sekitar sistem berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait) [2].

*Sequence diagram* biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah event untuk menghasilkan *output* tertentu. Diawali dari apa yang men-*trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan *output* apa yang dihasilkan. Masing-masing objek, termasuk aktor, memiliki *lifeline* vertikal.

*Message* digambarkan sebagai garis berpanah dari satu objek ke objek lainnya. Pada fase desain berikutnya, *message* akan dipetakan menjadi operasi/metoda dari *class*. *Activation bar* menunjukkan lamanya eksekusi sebuah proses, biasanya diawali dengan diterimanya sebuah *message*.



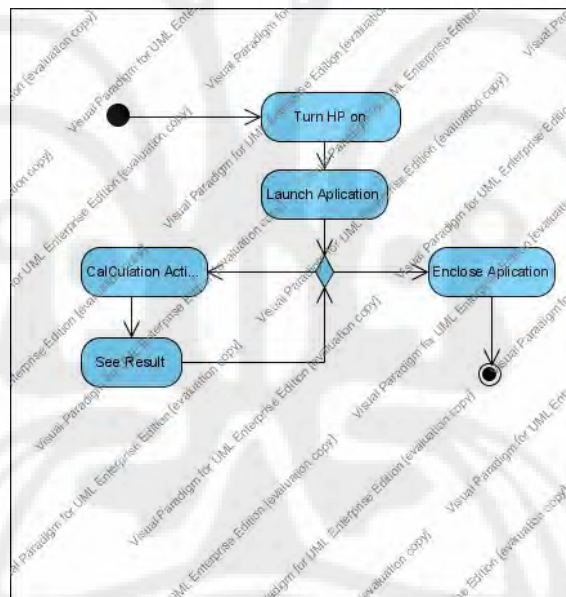
Gambar 2.7 – Sequence Diagram



### 2.3.3 Activity Diagram

*Activity diagram* menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi [2].

*Activity diagram* merupakan *state diagram* khusus, di mana sebagian besar *state* adalah *action* dan sebagian besar transisi di-*trigger* oleh selesainya *state* sebelumnya (*internal processing*). Oleh karena itu *activity diagram* tidak menggambarkan *behaviour internal* sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum.



**Gambar 2.8 – Activity Diagram**

## 2.4 WEB

*Internet* sebenarnya mengacu pada istilah untuk menyebut sebuah jaringan, bukannya suatu aplikasi tertentu. Karenanya, internet tidaklah memiliki manfaat apa-apa tanpa adanya aplikasi yang sesuai. Internet menyediakan beragam aplikasi yang dapat digunakan untuk berbagai keperluan. Setiap aplikasi

berjalan di atas sebuah protokol tertentu. Istilah “protokol” di internet mengacu pada satu set aturan yang mengatur bagaimana sebuah aplikasi berkomunikasi dalam suatu jaringan.

Dewasa ini, *World Wide Web* (WWW) atau yang sering disebut sebagai “*web*” saja adalah merupakan aplikasi internet yang paling populer. Demikian populernya hingga banyak orang yang keliru mengidentikkan web dengan internet. Sehingga perlu diketahui bahwa *World Wide Web* (WWW) bukanlah internet, demikian pula sebaliknya. Internet merupakan suatu jaringan *computer global*, sedangkan WWW bukan sekedar jaringan tetapi di dalamnya terdapat suatu set aplikasi komunikasi dan sistem perangkat lunak.

Web merupakan sebuah database jalinan komputer di seluruh dunia yang menggunakan sebuah arsitektur pengambilan informasi yang umum. Web tidak terpisah dari internet melainkan interface untuk menggali informasi yang ada di dalam internet. Web terbentuk dari website individu. *Website* seperti buku yang memiliki satu atau lebih page atau dokumen web yang terhubung antara page yang satu dengan yang lainnya.

Secara teknis, web adalah “sebuah sistem di mana informasi dalam bentuk teks umumnya ditulis dalam bentuk teks, gambar, suara dan lain-lain yang tersimpan dalam sebuah internet *web server* dipresentasikan dalam bentuk *hypertext*”. Informasi di web dalam bentuk teks umumnya ditulis dalam format HTML (*Hypertext Markup Language*). Informasi lainnya disajikan dalam bentuk grafis (dalam format GIF, JPG, PNG), suara (dalam format AU, WAV), dan objek multimedia lainnya (seperti MIDI, Shockwave, Quicktime Movie, 3d World).

Web dapat diakses oleh perangkat lunak web client yang secara populer disebut sebagai *browser*. *Browser* membaca halaman-halaman web yang tersimpan dalam *web server* melalui protokol yang disebut HTTP (*Hypertext Transfer protocol*). Beberapa *browser* yang cukup populer digunakan saat ini diantaranya *Mozilla Firefox*, *Microsoft Internet Explorer*, *Opera* dan *Google Chrome*.

Sebagai dokumen *hypertext*, dokumen-dokumen di web dapat memiliki *link* (sambungan) dengan dokumen lain, baik yang tersimpan dalam *web server* yang sama maupun di *web server* lainnya. Link memudahkan para pengakses web “berkelana” dari satu server ke server lain. Kegiatan penelusuran halaman web ini biasa kita sebut *browsing* atau *surfing*.

#### **2.4.1 Web Database**

Web database merupakan sistem penyimpanan data yang dapat diakses oleh bahasa pemrograman tertentu. Suatu database biasa dapat menjadi *web database* jika suatu database dapat diakses *user* menggunakan *web browser*. *Web database* tidak seperti sistem database konvensional yang hanya ditujukan untuk platform tertentu, karena *web database* dapat diakses oleh aplikasi *web*. *Web database* dapat diakses oleh aplikasi-aplikasi web yang dikembangkan dengan *tag HTML*, *control ActiveX*, dan pemrograman yang bersifat *server-side* melalui CGI, *Microsoft IIS (Internet Information Server)*, atau *script* yang bersifat *server-side* seperti *Microsoft Active Server Pages (ASP)* dan *Java Server Pages (JSP)*.

Web database dapat digunakan untuk berbagai macam keperluan misalnya menyediakan data stok barang untuk situs-situs *e-commerce*, layanan pengisian IRS (Isian Rencana Studi) mahasiswa, dan lain-lain.

#### **2.5 Hyper Text Markup Language (HTML)**

HTML digunakan untuk membangun dan menampilkan informasi suatu halaman *web*. Sekalipun banyak orang menyebutnya sebagai suatu bahasa pemrograman, HTML sebenarnya sama sekali bukan bahasa pemrograman, karena seperti tercermin dari namanya, HTML adalah suatu bahasa *mark up*. HTML digunakan untuk melakukan *mark up* (penandaan) terhadap sebuah dokumen teks. Simbol *mark up* yang digunakan oleh HTML ditandai dengan tanda lebih kecil (<) dan tanda lebih besar (>) sebagaimana disebut dengan *tag*. Kedua tanda tersebut digunakan untuk menentukan format atau *style* dari teks

yang ditandai. Perlu diketahui bahwa HTML tidak bersifat *Case Sensitive*. Contoh struktur lengkap HTML dapat dilihat pada contoh berikut :

```
<HTML>
<HEAD>
<TITLE>A Really Basic Document</TITLE>
</HEAD>
<BODY>
This is a really basic document.
</BODY>
</HTML>
```

Struktur di atas diapit oleh *tag* `<HTML>` dan `</HTML>` yang berarti menandai awal dan akhir dari sebuah dokumen HTML. Dua bagian di dalamnya adalah bagian kepala beserta judul dokumen. Bagian kedua yang diapit oleh *tag* `<BODY>` dan `</BODY>` berisi dokumen atau informasi yang hendak disajikan.

## 2.6 *Java Server Pages (JSP)*

Teknologi *Java Server Pages (JSP)* memungkinkan desainer dan pengembang Web (*Web developers and designers*) untuk mengembangkan halaman Web dinamis (*dynamic Web Pages*) secara cepat dan mudah dirawat, serta kaya informasi yang dapat membantu sistem bisnis. Sebagai bagian dari famili teknologi Java, teknologi JSP memungkinkan pengembangan secara cepat dari aplikasi berbasis Web yang *platform independent*. Teknologi JSP memisahkan antarmuka pengguna (*user*) dari penghasil konten (*content generations*), memungkinkan desainer untuk mengubah keseluruhan susunan halaman tanpa mengubah bagian konten dinamisnya [4].

### 2.6.1 Keuntungan JSP bagi Pengembang

Seorang pengembang atau desainer halaman Web yang familiar dengan HTML, dapat :

- **Menggunakan teknologi JSP tanpa perlu belajar bahasa JAVA :** Teknologi JSP dapat digunakan tanpa perlu mempelajari bagaimana cara menulis Java *scriptlets*. Meskipun *scriptlets* tidak lagi diperlukan untuk menghasilkan konten dinamis, *scriptlets* tetap didukung untuk menyediakan *backward compatibility*.
- **Memperluas bahasa JSP:** Pengembang dan desainer Java *tag library* dapat memperluas bahasa JSP dengan “simple tag handler”, yang memanfaatkan *tag extension* API yang lebih baru, lebih sederhana dan lebih bersih. Ini memacu berkembangnya *tag librariestersedia* yang *pluggable dan reusable*, sehingga mengurangi jumlah kode yang dibutuhkan untuk menulis aplikasi Web yang tangguh.
- **Dengan mudah menulis dan merawat halaman :** Bahasa ekspresi JavaServer Pages Standard Tag Library (JSTL) sekarang terintegrasi ke dalam teknologi JSP dan telah di-*upgrade* untuk mendukung fungsi-fungsi. Bahasa ekspresi sekarang dapat digunakan disamping penggunaan ekspresi *scriptlet*.

### 2.6.2 Teknologi JSP dan Java Servlets

Teknologi JSP menggunakan *tag-tag* seperti XML yang meng-*encapsulate* logika yang menghasilkan konten untuk halaman. Aplikasi logika dapat berada di sumber berbasis server (seperti arsitektur komponen JavaBeans) yang mengakses halaman dengan tag tersebut. Setiap dan semua format *tag* (HTML atau XML) langsung dilewatkan kembali ke halaman respon. Dengan memisahkan halaman logika dari desain dan tampilan dan mendukung desain yang berbasis komponen *reusable*, teknologi JSP menjadikannya lebih cepat dan

lebih mudah untuk membangun aplikasi berbasis Web.

Teknologi *Java Server Pages* merupakan perpanjangan dari teknologi *Java servlet*. *Servlets* adalah platform-independen, modul *server-side* yang sesuai untuk *Web server framework* dan dapat digunakan untuk memperluas kemampuan dari server web dengan *overhead*, pemeliharaan, dan dukungan yang minimal. Tidak seperti bahasa scripting lainnya, *servlets* tidak melibatkan pertimbangan atau modifikasi *platform* khusus, mereka adalah komponen aplikasi yang di-*download*, sesuai permintaan, dengan bagian dari sistem yang memerlukan mereka. JSP dan teknologi *servlets* menyediakan alternatif yang menarik untuk jenis lain dari *Web scripting* / pemrograman dinamis dengan menawarkan: *platform independence*; peningkatan kinerja; pemisahan logika dari tampilan; kemudahan administrasi; kemungkinan diperpanjang menjadi perusahaan, dan yang terpenting, kemudahan penggunaan.

Saat ini *servlets* adalah pilihan yang populer untuk membangun aplikasi Web interaktif. *Third-party servlet container* tersedia untuk *Apache Web Server*, *Microsoft IIS*, dan lain-lain. *Servlet container* biasanya komponen dari server Web dan aplikasi, seperti aplikasi *BEA WebLogic Server*, *IBM WebSphere*, *Sun Java System Web Server*, *Sun Java System Application Server*, dan lain-lain.

### 2.6.3 Latar Belakang Komunitas

Spesifikasi JSP adalah produk dari kolaborasi industri yang luas dengan pemimpin industri di dalam perusahaan perangkat lunak dan pasar perangkat, yang dipimpin oleh Sun Microsystems. Sun menjadikan spesifikasi JSP bebas tersedia untuk komunitas pengembang, dengan tujuan agar setiap server aplikasi dan server Web akan mendukung antarmuka JSP. Halaman JSP berbagi keunggulan teknologi Java "*Write Once, Run Anywhere*". Teknologi JSP merupakan komponen kunci dalam Java 2 Platform, Enterprise Edition, Arsitektur yang sangat *scalable* buatan Sun untuk aplikasi *enterprise*.

## 2.7 MySQL

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (*database management system*) atau DBMS yang *multithread*, *multi-user*, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis dibawah lisensi GNU General Public License (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL.

Tidak sama dengan proyek-proyek seperti Apache, dimana perangkat lunak dikembangkan oleh komunitas umum, dan hak cipta untuk kode sumber dimiliki oleh penulisnya masing-masing, MySQL dimiliki dan disponsori oleh sebuah perusahaan komersial Swedia MySQL AB, dimana memegang hak cipta hampir atas semua kode sumbernya. Kedua orang Swedia dan satu orang Finlandia yang mendirikan MySQL AB adalah: David Axmark, Allan Larsson, dan Michael "Monty" Widenius.

### 2.7.1 *Relational Database Management System (RDBMS)*

MySQL adalah Relational Database Management System (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (General Public License). Dimana setiap orang bebas untuk menggunakan MySQL, namun tidak boleh dijadikan produk turunan yang bersifat komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam database sejak lama, yaitu SQL (*Structured Query Language*). SQL adalah sebuah konsep pengoperasian database, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis.

Keandalan suatu sistem database (DBMS) dapat diketahui dari cara kerja optimizer-nya dalam melakukan proses perintah-perintah SQL, yang dibuat oleh user maupun program-program aplikasinya. Sebagai *database server*, MySQL dapat dikatakan lebih unggul dibandingkan *database server* lainnya dalam *query*

*data*. Hal ini terbukti untuk *query* yang dilakukan oleh *single user*, kecepatan *query* MySQL bisa sepuluh kali lebih cepat dari PostgreSQL dan lima kali lebih cepat dibandingkan Interbase.

MySQL memiliki beberapa keistimewaan, antara lain :

1. **Portabilitas.** MySQL dapat berjalan stabil pada berbagai sistem operasi seperti Windows, Linux, FreeBSD, Mac Os X Server, Solaris, Amiga, dan masih banyak lagi.
2. **Open Source.** MySQL didistribusikan secara *open source*, dibawah lisensi GPL sehingga dapat digunakan secara cuma-cuma.
3. **'Multiuser'.** MySQL dapat digunakan oleh beberapa user dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.
4. **'Performance tuning'.** MySQL memiliki kecepatan yang menakjubkan dalam menangani *query* sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.
5. **Jenis Kolom.** MySQL memiliki tipe kolom yang sangat kompleks, seperti signed / unsigned integer, float, double, char, text, date, timestamp, dan lain-lain.
6. **Perintah dan Fungsi.** MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah Select dan Where dalam perintah (*query*).
7. **Keamanan.** MySQL memiliki beberapa lapisan sekuritas seperti level subnetmask, nama host, dan izin akses *user* dengan sistem perizinan yang mendetail serta sandi terenkripsi.
8. **Skalabilitas dan Pembatasan.** MySQL mampu menangani basis data dalam skala besar, dengan jumlah rekaman (records) lebih dari 50 juta dan 60 ribu tabel serta 5 milyar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.



9. **Konektivitas.** MySQL dapat melakukan koneksi dengan klien menggunakan protokol TCP/IP, Unix socket (UNIX), atau Named Pipes (NT).
10. **Lokalisasi.** MySQL dapat mendeteksi pesan kesalahan pada klien dengan menggunakan lebih dari dua puluh bahasa. Meski pun demikian, bahasa Indonesia belum termasuk di dalamnya.
11. **Antar Muka.** MySQL memiliki interface (antar muka) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (Application Programming Interface).
12. **Klien dan Peralatan.** MySQL dilengkapi dengan berbagai peralatan (tool) yang dapat digunakan untuk administrasi basis data, dan pada setiap peralatan yang ada disertakan petunjuk online.
13. **Struktur tabel.** MySQL memiliki struktur tabel yang lebih fleksibel dalam menangani ALTER TABLE, dibandingkan basis data lainnya semacam PostgreSQL ataupun Oracle.

Terdapat beberapa API tersedia yang memungkinkan aplikasi-aplikasi komputer yang ditulis dalam berbagai bahasa pemrograman untuk dapat mengakses basis data MySQL antara lain: bahasa pemrograman C, C++, C#, bahasa pemrograman Eiffel, bahasa pemrograman Smalltalk, bahasa pemrograman Java, bahasa pemrograman Lisp, Perl, PHP, bahasa pemrograman Python, Ruby, REALbasic dan Tcl. Sebuah antarmuka ODBC memanggil MyODBC yang memungkinkan setiap bahasa pemrograman yang mendukung ODBC untuk berkomunikasi dengan basis data MySQL. Kebanyakan kode sumber MySQL dalam ANSI C.

MySQL sangat populer dalam aplikasi web seperti MediaWiki (perangkat lunak yang dipakai Wikipedia dan proyek-proyek sejenis) dan PHP-Nuke dan berfungsi sebagai komponen basis data dalam LAMP. Popularitas sebagai aplikasi

web dikarenakan kedekatannya dengan popularitas PHP, sehingga seringkali disebut sebagai *Dynamic Duo*.

Untuk melakukan administrasi dalam basis data MySQL, dapat menggunakan modul yang sudah termasuk yaitu command-line (perintah: mysql dan mysqladmin). Juga dapat diunduh dari situs MySQL yaitu sebuah modul berbasis grafik (GUI): MySQL Administrator dan MySQL Query Browser. Selain itu terdapat juga sebuah perangkat lunak gratis untuk administrasi basis data MySQL berbasis web yang sangat populer yaitu phpMyAdmin. Untuk perangkat lunak untuk administrasi basis data MySQL yang dijual secara komersial antara lain: MySQL front, Navicat dan EMS SQL Manager for MySQL.

## 2.8 SURVEILLANCE CAMERA

Kata *surveillance* umumnya digunakan untuk menjelaskan pengamatan dari jauh dengan peralatan elektronik (seperti kamera CCTV), atau penangkapan yang dikirimkan informasi elektronik (seperti Internet atau lalu lintas panggilan telepon). Namun, pengawasan juga dapat merujuk ke metode yang sederhana atau teknologi rendah seperti *human intelligence agents* dan *postal interception*.

Kamera *Surveillance* adalah video kamera yang digunakan untuk tujuan mengamati suatu daerah. Kamera ini sering terhubung ke perangkat rekaman, jaringan IP, dan / atau dipantau oleh seorang penjaga keamanan / aparat penegak hukum. Kamera dan peralatan yang digunakan untuk merekam awalnya relatif mahal dan diperlukan personil manusia untuk memantaunya. Sekarang dengan teknik produksi yang lebih murah, membuatnya cukup sederhana dan tidak mahal untuk digunakan dalam sistem keamanan rumah, dan untuk pengawasan sehari-hari. Analisa *video footage* pun saat ini telah dipermudah dengan adanya otomasi sistem kamera *Surveillance* seperti ini telah terpasang di jutaan wilayah di berbagai negara, dan saat ini terus dipantau oleh program komputer otomatis bukan manusia.

## 2.9 RADIO FREQUENCY IDENTIFICATION (RFID)

RFID adalah sebuah istilah umum yang digunakan untuk menjelaskan sebuah alat yang mentransmisikan identitas (dalam bentuk deretan nomor) dari sebuah objek atau manusia secara nirkabel, menggunakan gelombang radio. RFID dikelompokkan sebagai teknologi *automatic identification* (identifikasi otomatis).

Teknologi identifikasi otomatis melingkupi *barcode*, pembaca karakter optikal dan beberapa teknologi *biometric*, seperti pemindai retina. Teknologi identifikasi otomatis telah digunakan untuk mengurangi jumlah waktu dan tenaga kerja yang dibutuhkan untuk memasukkan data secara manual dan meningkatkan akurasi data.

RFID tag terdiri dari sebuah microchip yang tertanam pada sebuah antenna gelombang radio yang ditempelkan pada sebuah substrat (biasanya silikon). Chip tersebut dapat menyimpan data seukuran 2 kilobyte. Sebagai contoh, informasi tentang sebuah barang atau tanggal pengkapalan pada sebuah pabrik, dan sebagainya, dapat dituliskan pada sebuah *tag*.

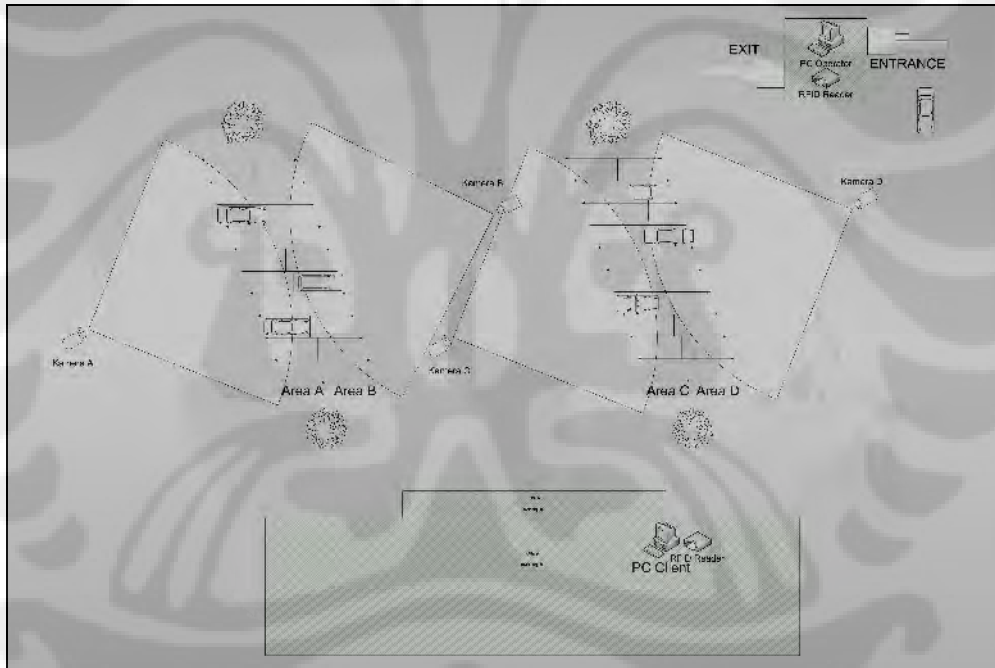
Untuk mengambil data yang tersimpan pada RFID tag, dibutuhkan sebuah *reader*. *Reader* ini adalah sebuah alat yang memiliki satu atau banyak antenna yang memancarkan gelombang radio dan menerima sinyal kembali dari tag. *Reader* kemudian melewati informasi tersebut dalam bentuk *digital* ke sistem computer.

### BAB III

## PEMODELAN SISTEM DAN RENCANA UJI COBA

### 3.1 Skenario

Suatu lahan parkir terbuka dengan 4 area parkir dengan 10 petak parkir di setiap areanya, dilengkapi dengan sistem informasi perparkiran yang terintegrasi dengan sistem RFID dan perangkat pengawasan berupa 4 buah kamera yang diposisikan dengan sudut pandang menyorot pada seluruh petak yang ada di setiap area, sehingga keadaan di setiap area bisa dipantau dengan baik, dan dengan demikian keamanan lahan parkir ini lebih terjaga. Ilustrasi lahan parkir tersebut ditunjukkan pada gambar berikut :



**Gambar 3.1 - Denah Lahan Parkir**

Pada lahan parkir tersebut, terdapat operator parkir yang berada di gerbang masuk dan keluar lahan parkir. Operator parkir ini bertugas melakukan entri data pengguna parkir. Di dalam PC operator terdapat sistem informasi berbasis web yang memiliki beberapa fitur dan fungsi di setiap halamannya. Sistem informasi

ini berfungsi sebagai penyedia informasi yang dibutuhkan pemilik kendaraan dan juga operator. Selain itu sistem informasi ini juga berguna membantu proses keamanan dengan adanya fitur pengawasan sederhana yang dimilikinya.

Halaman "**Home**" menunjukkan sisa slot parkir yang masih tersedia, sehingga dapat membantu operator untuk mengetahui apakah lahan parkir sudah penuh atau belum. Halaman "**Letak Kendaraan**" dapat digunakan oleh pemilik kendaraan untuk melihat di mana kendaraannya berada dan bagaimana keadaannya saat ini, untuk melihatnya, pemilik kendaraan bisa secara manual memasukkan nomor kendaraannya ke dalam formulir yang tersedia di dalam halaman tersebut atau bisa juga menggunakan kartu RFID dengan cara mendekatkannya ke pembaca RFID, kemudian sistem akan memetakan di mana kendaraannya berada saat ini. Halaman ini bisa diakses oleh pemilik kendaraan melalui PC Client yang terletak di dalam gedung seperti yang diilustrasikan pada gambar 3.1.

Halaman "**Input Kendaraan**" dapat digunakan oleh operator untuk mendata kendaraan yang masuk ataupun keluar lahan parkir, operator bertugas memilihkan petak parkir yang masih kosong untuk kendaraan yang masuk, kemudian petugas parkir yang berada di area parkir akan mengarahkan pemilik kendaraan menuju petak parkir yang telah ditentukan oleh operator.

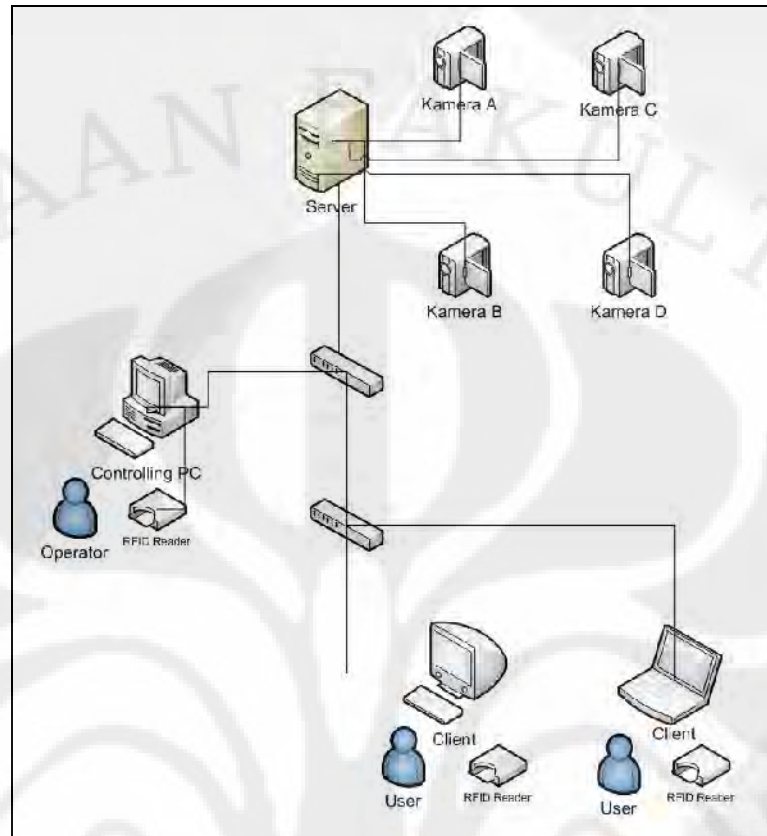
Bagi pemilik kendaraan yang belum terdaftar di area parkir tersebut dan belum memiliki kartu RFID sebagai pengenalan, akan diarahkan oleh operator untuk melakukan registrasi, proses registrasi ini dibantu langsung oleh operator, operator akan membuka halaman "**Registrasi**" dan akan mengisi data pribadi pemilik kendaraan kemudian menyerahkan kartu RFID baru kepada pemilik kendaraan. Kemudian di dalam sistem informasi ini terdapat halaman "**Arsip**" yang berisi informasi yang dibutuhkan operator dan petugas parkir untuk memantau lokasi parkir tanpa harus melihat langsung di lokasi sebenarnya. Ada 4 sub-halaman pada halaman arsip ini, yaitu halaman "**Data Parking In**" dan "**Data Parking Out**", kedua halaman ini akan menunjukkan *log* dari kendaraan yang telah masuk

dan keluar area parkir. Sub-halaman berikutnya adalah halaman “*Time Based Capturing*” dan “*Motion Based Capturing*”, kedua halaman ini akan menunjukkan arsip foto-foto area parkir yang disimpan berdasarkan interval waktu tertentu dan juga berdasarkan pergerakan objek yang disorot.

Bagi petugas parkir, halaman ini akan membantu dalam hal keamanan, karena proses pengawasan sederhana ini dapat memudahkan petugas parkir dalam memantau lokasi parkir. Selain itu, dengan adanya arsip ini, dapat membantu proses pelacakan jika memang terjadi proses pencurian. Kemudian halaman terakhir adalah halaman denah parkir yang menunjukkan peta lokasi parkir tersebut.

### **3.2 Arsitektur Sistem Secara Global**

Pada bab berikut ini, akan dibuat rancangan awal berupa prototipe untuk sistem informasi perparkiran berbasis Web. Arsitektur sistem secara global artinya kita melihat suatu bentuk perancangan yang melibatkan berbagai sistem. Perancangan sistem-sistem tersebut meliputi pemilihan sistem jaringan yang tepat, perancangan perangkat lunak untuk sisi *Client* dan *Server*, serta pemilihan *hardware* yang cocok sebagai prototipe. Perancangan sistem secara global dapat dilihat dari gambar berikut :

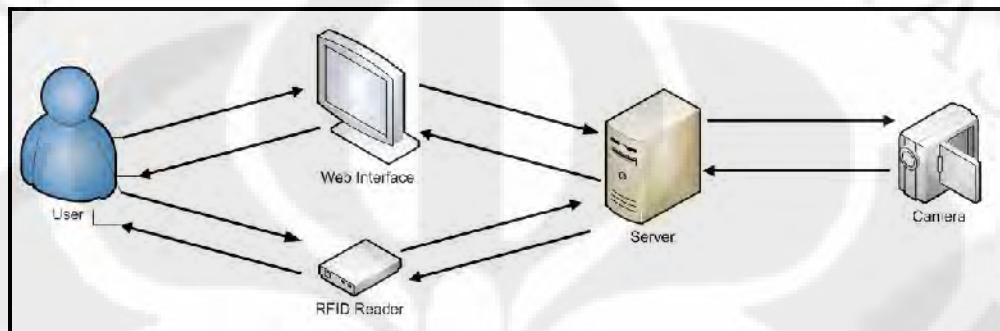


**Gambar 3.2 – Arsitektur sistem secara global**

Sistem informasi ini dirancang untuk sistem parkir berbasis RFID. Pada Controlling PC, terdapat aplikasi untuk membaca RFID Reader yang nantinya juga akan terhubung ke suatu database yang akan dipakai bersama-sama dengan sistem informasi yang akan dirancang. Terdapat 4 buah Kamera untuk *surveillance* yang ditempatkan dengan sudut yang sesuai pada empat area parkir, yaitu area A, area B, area C dan area D. Kamera-kamera ini terhubung langsung ke server, sehingga ketika gambar lokasi parkir diambil, gambar bisa langsung disimpan di server untuk kemudian dipergunakan dan ditampilkan pada *Web interface*. Pada sistem ini, *user* dan *operator* dapat melakukan pencarian lokasi parkir yang sesuai dengan bantuan RFID *tag* sebagai pengenalan identitas, ataupun melakukan *manual input* berupa nomor kendaraan untuk melakukan pencarian

lokasi parkir. Akses web dapat dilakukan menggunakan bermacam-macam terminal yang mendukung antarmuka web, tentunya yang dapat menampilkan web interface dengan optimal adalah PC dan Laptop.

Kemudian gambaran aplikasi yang akan dirancang dan nantinya yang akan diimplementasikan dapat dilihat dari gambar berikut ini :



**Gambar 3.3 – Rencana perancangan sistem**

Dari gambar diatas dapat dilihat bahwa ada 5 elemen yang terlibat di dalam perancangan prototipe dari sistem informasi perparkiran berbasis web ini. Yaitu *User* (pengguna), *web interface*, *RFID Reader* berserta *tag*-nya, *server* dan kamera. Aplikasi ini akan dijalankan oleh pengguna yang berinteraksi dengan *web interface*. Pengguna dapat menggunakan *web browser* seperti *Mozilla Firefox*, *Safari*, *Opera*, dan *Internet Explorer*. Apabila situs ini dibuka, maka *browser* akan mengunduh tampilan situs yang berisi gambar-gambar hasil capture lokasi parkir yang tersimpan pada *server*. *Server* di sini bertindak sebagai *Web server* dan juga *database server*. Program dengan tampilan *web* inilah yang berinteraksi secara interaktif dengan pengguna. Apabila pengguna melakukan suatu perintah, eksekusinya akan diproses di *web server*. Sedangkan bila terdapat permintaan untuk mengakses *database*, maka *database* tersebut akan dipanggil ke dalam program. Aplikasi ini memiliki tiga fitur utama yaitu, pencarian lokasi parkir bagi *user* yang menempati *slot* parkir tertentu, kemudian menampilkan foto hasil *capture* kamera sesuai lokasi parkirnya, pemuatannya secara otomatis diambil dari



folder tempat foto disimpan, terakhir yaitu menampilkan sisa slot parkir yang tersedia. Lalu terdapat fitur tambahan, yaitu menampilkan denah lokasi parkir.

### 3.3 Perancangan Perangkat Keras

Sisi perangkat keras sistem terdiri atas tiga unit yaitu tag RFID, pembaca RFID dan kamera web. Susunannya seperti yang digambarkan di skema desain sistem pada Gambar 3.2 di mana masing-masing kamera akan menyorot ke area tertentu.

#### 3.3.1 Peralatan



**Gambar 3. 4 RFID Starter-Kit**

Bagian yang pertama adalah pembaca RFID. Di sini divais pembaca yang digunakan adalah RFID Starter-Kit yang berbasis reader tipe ID-12. Pembaca ini mampu membaca tag RFID pasif hingga jarak 12 centimeter. Namun karena pengerjaan ini sifatnya menghasilkan prototipe sistem, maka sebagai permulaan digunakan pembaca ini. Antarmuka keluaran RFID Starter-Kit adalah port RJ11 dengan format data UART RS-232. Selain format data tersebut dapat juga diatur menjadi protokol UART TTL, Wiegand26, dan Magnet Emulation. Dalam tugas akhir ini dipakai format data keluaran UART RS-232 juga untuk kompatibilitas modul berikutnya.



**Gambar 3.5 Tag RFID Pasif Class-1 berbentuk kartu**

Kemudian perangkat berikutnya adalah Tag RFID yang digunakan sebagai tanda pengenal, setiap tag mempunyai nomor identifikasi yang unik yang tidak mungkin sama untuk setiap kartunya. Jenis tag yang digunakan adalah tag pasif Kelas-1. Tag ini mendapatkan sumber daya dari pembaca RFID kemudian mentransmisikan nomor identifikasinya. Ukuran tag yang digunakan berukuran seperti kartu kredit biasa. Karena tag ini memerlukan sumber daya dari pembaca RFID, ini menyebabkan jarak baca tag dengan pembacanya tidak bisa tepat sesuai spesifikasi dari pembaca RFID yang digunakan, yaitu maksimal 12 cm. Tag aktif yang menggunakan sumber daya sendiri akan lebih cocok digunakan jika diaplikasikan ke lingkungan parkir yang sebenarnya. Untuk sistem yang sedang dirancang tidak menggunakan tag aktif karena saat ini ketersediaannya masih kosong dan harganya pun relatif mahal, penulis merasa belum perlu menggunakan tag aktif untuk sistem yang masih prototipe ini.

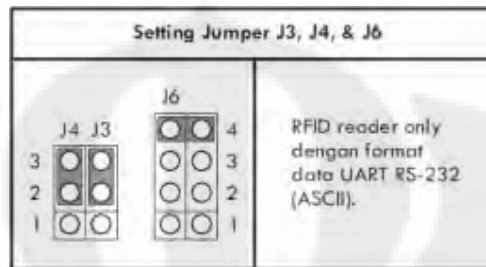


**Gambar 3.6 Kamera Web**

Modul ketiga yang digunakan dalam sisi perangkat keras adalah tiga buah kamera web (*webcam*), Masing-masing dengan spesifikasi yang berbeda. Yang pertama adalah *webcam* Prolink 1310 dengan resolusi 1.3 MegaPixels, kemudian *webcam* Mediatech dengan resolusi 1.3 MegaPixels, terakhir adalah *webcam* HP yang terintegrasi di dalam laptop dengan resolusi 2 MegaPixels. Divais ini

dipasang dengan konektivitas USB dan terhubung langsung dengan program desktop pendukung *webcam* yaitu *Yawcam*.

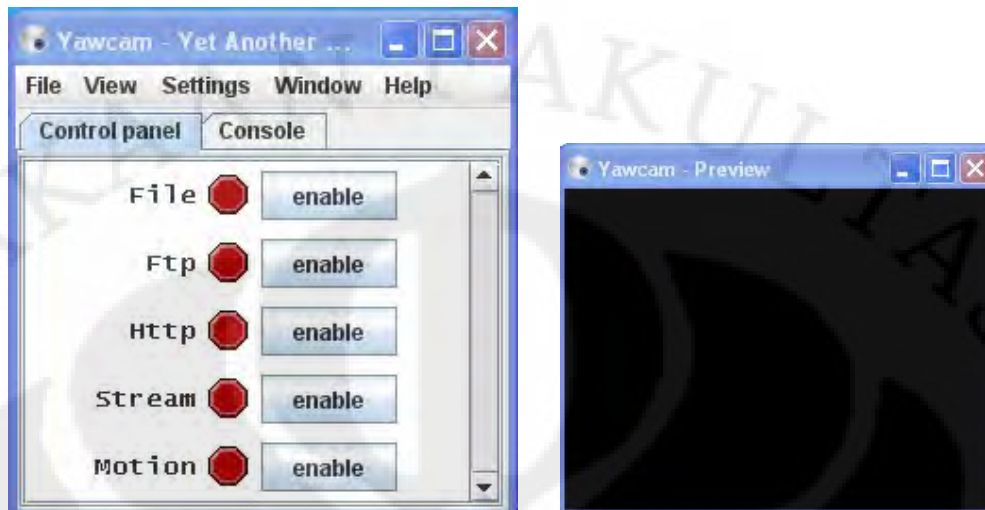
### 3.3.2 Konfigurasi



**Gambar 3. 7 Setting Jumper Pada RFID Starter-Kit**

Modul yang pertama kali dikonfigurasi adalah pembaca RFID. Pada modul ini tidak banyak dilakukan konfigurasi. Hanya memastikan bahwa format data keluarannya adalah UART RS-232, yaitu dengan mengkonfigurasi jumper pada RFID reader seperti pada gambar 3.5. Penentuan ini harus disesuaikan dengan parameter serial yang dimiliki oleh pembaca RFID. Jika bernilai beda maka koneksi data tidak akan sinkron dengan program *desktop* yang dirancang. Karena pembaca-RFID yang dipakai sudah mempunyai parameter tetap maka kita samakan nilai tersebut pada aplikasi *desktop* pembaca RFID. Parameter serial yang dipakai adalah speed: **9600** bps, databit: **8**, parity: **none**, stop bits: **1**, flow control: **none**.

Modul kedua yang dikonfigurasi adalah modul kamera web, kamera web ini dihubungkan dengan port USB dan diaktifkan menggunakan aplikasi tambahan *Yawcam*, aplikasi *Yawcam* ini memiliki beberapa macam fitur, di antara sekian fitur yang ada, fitur yang digunakan untuk sistem ini yaitu pengambilan foto secara otomatis dalam interval waktu yang ditentukan, dan fitur pengambilan foto secara otomatis apabila ada objek yang di sorot oleh kamera berubah atau bergerak. Berikut urutan konfigurasi yang harus dilakukan untuk menjalankan kamera untuk pengambilan foto otomatis dengan interval waktu, setelah membuka aplikasi *Yawcam* akan muncul *window* seperti ini :



**Gambar 3.8 Aplikasi Kamera Web Yawcam**

Dengan meng-klik *enable* pada modul *File*, kamera akan secara otomatis mengambil foto dalam interval waktu tertentu, untuk menentukan berapa lama interval waktu yang dibutuhkan untuk mengambil foto berikutnya, bisa dilakukan konfigurasi pada menu *settings*. Pada Menu tersebut juga bisa diatur di mana kita akan menyimpan foto hasil capture kamera tadi. Kemudian untuk memungkinkan pengambilan foto otomatis jika objek berubah atau bergerak, bisa dilakukan dengan meng-klik *enable* pada modul *Motion*.

### 3.4 Perancangan Aplikasi Sistem Informasi Perparkiran Berbasis Web

Untuk membangun antarmuka web (*web interface*) pada aplikasi ini dipergunakan bahasa pemrograman JSP (*Java Server Pages*) dan HTML. JSP dipilih karena memiliki beberapa keunggulan, yaitu bagi programmer yang sudah familiar dengan bahasa pemrograman Java, untuk mempelajari cara membangun web dengan JSP tidak akan terlalu sulit, karena dasar dari JSP adalah bahasa Java.

Kemudian JSP juga sudah mendukung *multiplatform*, ini artinya JSP bisa berjalan di berbagai sistem operasi. Selain itu JSP juga dinilai lebih tahan dari ancaman dan gangguan keamanan, karena komunitas pengguna JSP belum

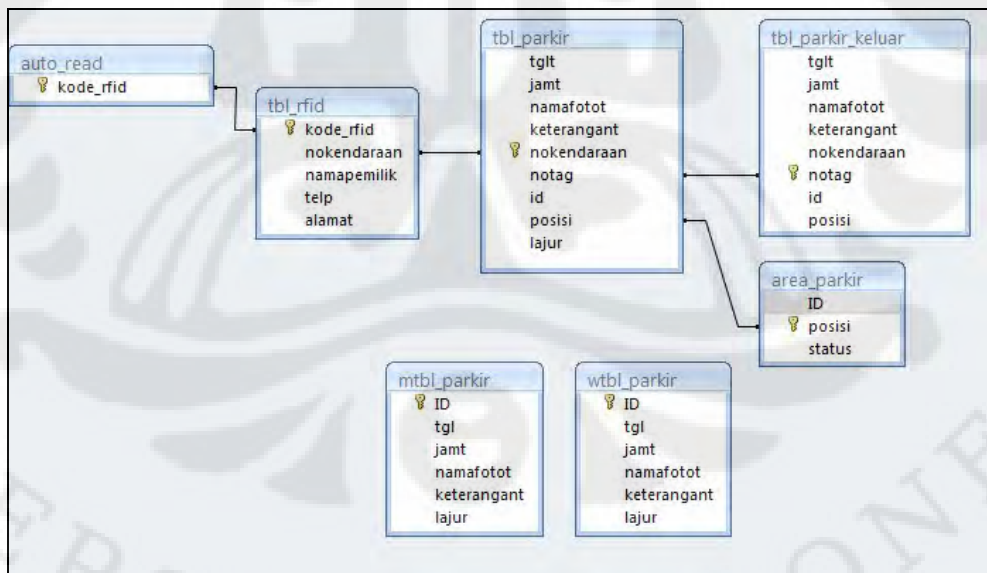
sebanyak komunitas pengguna bahasa pemrograman web dinamis lainnya seperti PHP.

Untuk mengkomunikasikan antara Web, RFID Reader dan User, diperlukan aplikasi *desktop* tambahan yang berbasis JAVA, aplikasi ini terhubung ke database yang sama, sehingga *web interface* dapat menangkap Inputan dari RFID tag yang melalui RFID Reader.

Database yang dipergunakan adalah MySQL. MySQL dipilih karena merupakan program yang Open Source, kemudian pengoperasiannya cukup mudah dan bahasa yang digunakan juga cukup familiar. Selain itu juga MySQL bisa dikoneksikan dengan Java atau JSP dengan menggunakan JDBC (*Java Database Connectivity*).

### 3.5 Database

Database terdiri atas tujuh tabel utama yaitu tabel *auto\_read*, *tbl\_rfid*, *area\_parkir*, *tbl\_parkir*, *wtbl\_parkir*, *mtbl\_parkir*, dan *tbl\_parkir\_keluar*. Di dalam database ini, ada beberapa tabel yang tidak memiliki keterkaitan satu sama lain, karena memang digunakan untuk fungsi tersendiri, sehingga tabelnya pun tidak memiliki relasi dengan tabel-tabel lainnya. Tabel-tabel tersebut di antaranya :



**Gambar 3.9 Diagram Relasi database**

Tabel `auto_read` merupakan tabel bantu yang berisikan nomor RFID tag, tabel ini akan terus di-*update* sesuai dengan inputan yang berasal dari RFID reader, tabel ini digunakan oleh aplikasi Reader yang berbasis JAVA dan juga aplikasi web yang berbasis JSP.

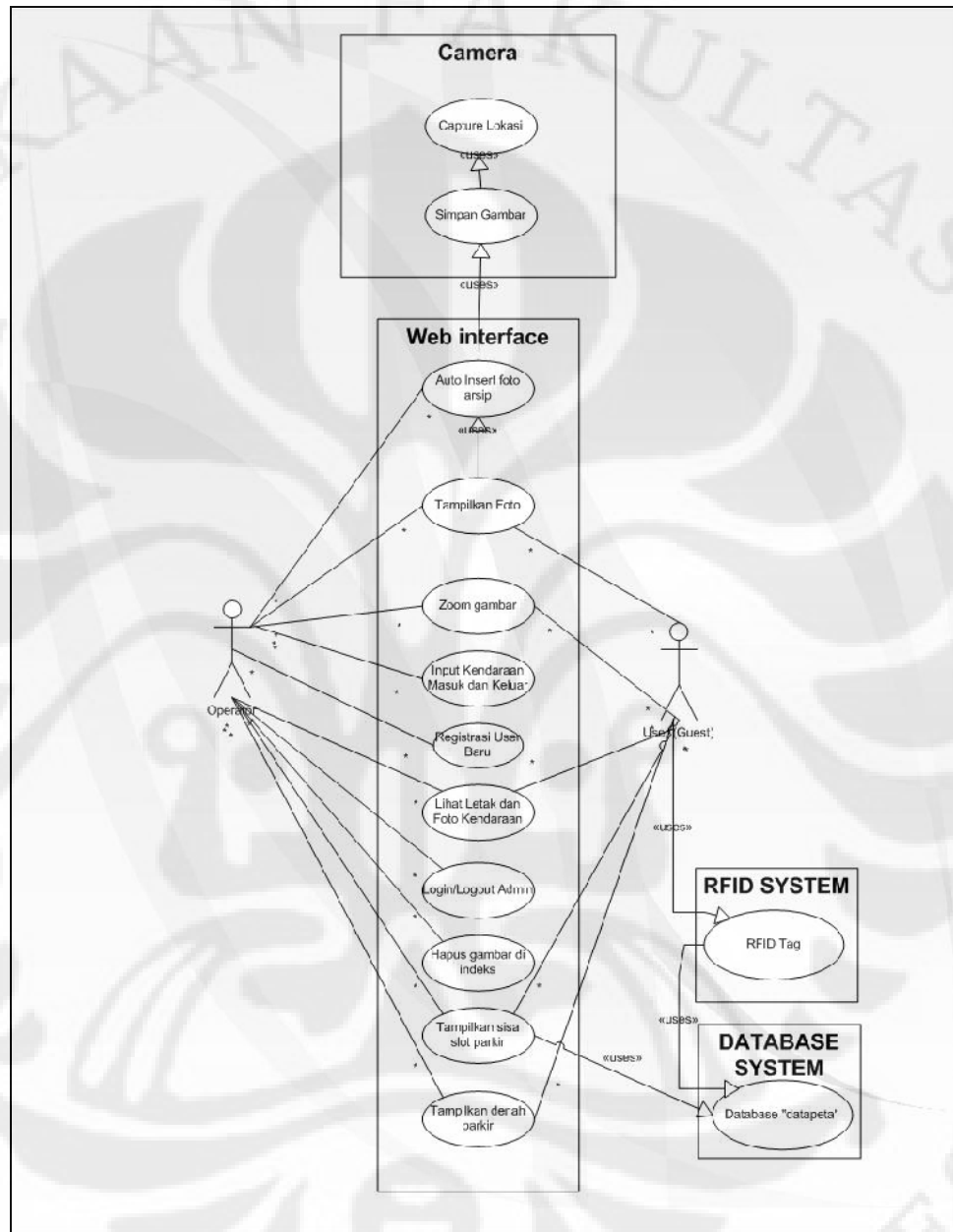
Tabel “`tbl_rfid`” digunakan oleh halaman registrasi untuk menyimpan data-data pemilik kendaraan yang telah terdaftar pada sistem ini. *Field* “**RFID**” yang menjadi Primary Key, tipe datanya berupa *varchar*, field ini terhubung dengan tabel `auto_read` dengan relasi *one-to-one*. Tabel “`area_parkir`” memiliki *field* “**posisi**” sebagai Primary Key yang terhubung dengan tabel “`tbl_parkir`”, tabel ini digunakan sistem untuk menandakan posisi kendaraan yang menggunakan lahan parkir. Kemudian tabel “`tbl_parkir`” digunakan oleh halaman Data Parking In untuk menyimpan *log* dari kendaraan-kendaraan yang menggunakan lahan parkir tersebut. Tabel “`tbl_parkir_keluar`” digunakan oleh halaman Data Parking Out untuk menyimpan *log* dari kendaraan-kendaraan yang telah keluar dari lahan parkir.

Tabel “`wtbl_parkir`” dan “`mtbl_parkir`” tidak memiliki relasi dengan tabel-tabel lainnya, karena kedua tabel ini digunakan untuk fungsi tersendiri. Kedua tabel ini digunakan oleh halaman arsip untuk menyimpan arsip-arsip foto yang telah ditangkap kamera di waktu lampau.

### 3.6 Pemodelan Aplikasi Sistem Informasi Perparkiran Berbasis Web

Untuk pemodelan aplikasi sistem informasi perparkiran berbasis web ini, digunakan bahasa pemodelan *Unified Modelling Language* (UML). Ada 3 diagram UML yang dipergunakan dalam pemodelan aplikasi ini, yaitu *Use Case diagram*, *Activity Diagram* dan *Sequence Diagram*.

### 3.6.1 USE CASE DIAGRAM



Gambar 3.10 – Use Case Diagram Sistem Informasi Perparkiran

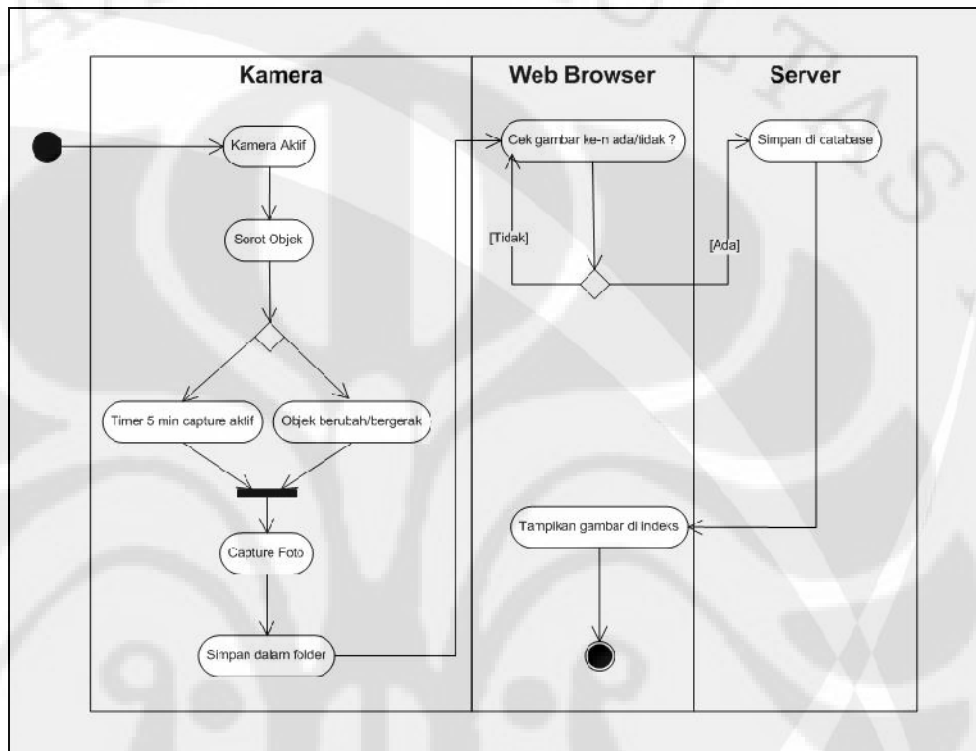
Gambar di atas merupakan diagram *use case* yang menunjukkan fungsionalitas dari aplikasi. Pada sistem ini, diskenariokan ada dua jenis user yang dapat mengakses halaman web, yaitu *Operator* dan *User (Guest)*. *Operator* memiliki hak penuh dalam penggunaan semua fitur yang terdapat pada aplikasi ini, sedangkan *User (Guest)* hanya memiliki porsi yang terbatas dalam penggunaan fiturnya. *Operator* dapat mengakses halaman Registrasi, yaitu untuk registrasi kendaraan yang belum terdaftar, kemudian juga bisa mengakses halaman input kendaraan yang digunakan ketika kendaraan yang sudah diregistrasi masuk ke area parkir dan menempati lokasi yang telah ditunjuk oleh *operator* parkir. *Operator* dan *user* dapat melihat berapa banyak *slot* parkir yang masih tersedia, sehingga bisa diketahui apakah lahan parkir masih bisa diisi oleh kendaraan lagi atau tidak. Selain itu, operator juga bisa melihat arsip-arsip foto yang diambil oleh kamera pada waktu lampau. Apabila kamera menangkap foto kemudian menyimpannya di media penyimpanan di server, aplikasi web akan secara otomatis menampilkan foto yang baru saja diambil di halaman arsip.

Dalam sistem ini, *user* yang sudah terdaftar dapat melihat letak kendaraannya dengan menggunakan RFID tag yang dimilikinya. Nomor RFID yang terbaca oleh pembaca RFID terhubung ke *database*, kemudian sistem akan menampilkan posisi kendaraan sesuai dengan posisi yang sejak awal sudah di daftarkan ketika pemilik kendaraan masuk ke area parkir.



### 3.6.2 Activity Diagram

#### Activity diagram untuk melihat halaman arsip foto berdasarkan waktu dan berdasarkan perubahan objek



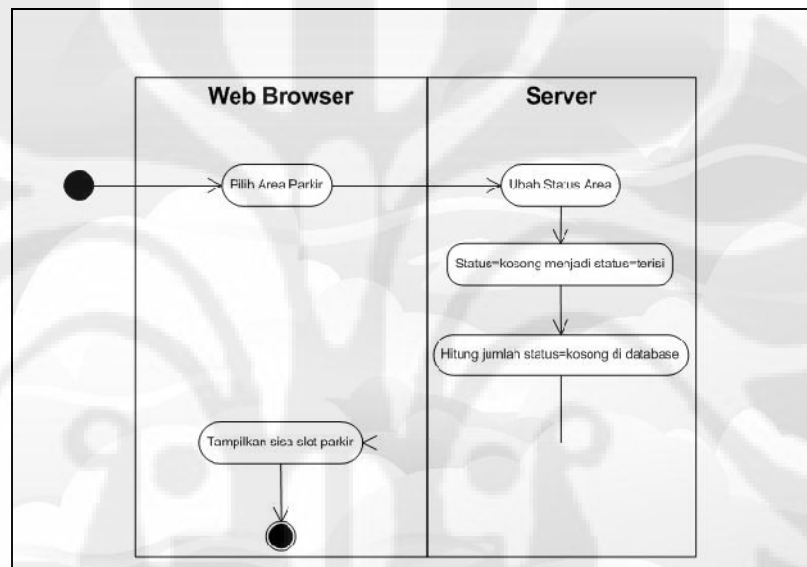
**Gambar 3.11 – Activity Diagram Untuk Capture Foto**

Untuk menampilkan hasil *capture* foto objek (lokasi parkir) dari kamera pada halaman *website*, kamera harus aktif terlebih dahulu, kemudian aplikasi pengontrol kamera juga harus diaktifkan agar proses *Auto-Capturing* objek dapat dilakukan, *Auto-Capturing* ini bisa terjadi melalui 2 cara, yang pertama dengan interval waktu tertentu, di mana interval waktunya bisa diatur dalam satuan detik. Kemudian cara yang kedua adalah dengan mendeteksi perubahan objek yang disorot oleh kamera, bila objek mengalami perubahan atau pergerakan, kamera akan secara otomatis melakukan *capture*.

Apabila objek telah berhasil di-*capture* maka akan tersimpan ke dalam folder pada media penyimpanan dalam bentuk *digital image* dengan format JPEG.

Kemudian apabila aplikasi web dijalankan, aplikasi web ini akan menjalankan fungsi *auto insert* gambar ke dalam halaman web dengan mengecek terlebih dahulu apakah di *folder* yang dimaksud ada gambar ke-n atau tidak, jika tidak maka fungsi gambar tidak akan di-*insert*, jika ada maka data dari gambar akan di simpan ke dalam *database*, kemudian aplikasi web ini juga akan memasukkan gambar ke halaman web untuk kemudian ditampilkan.

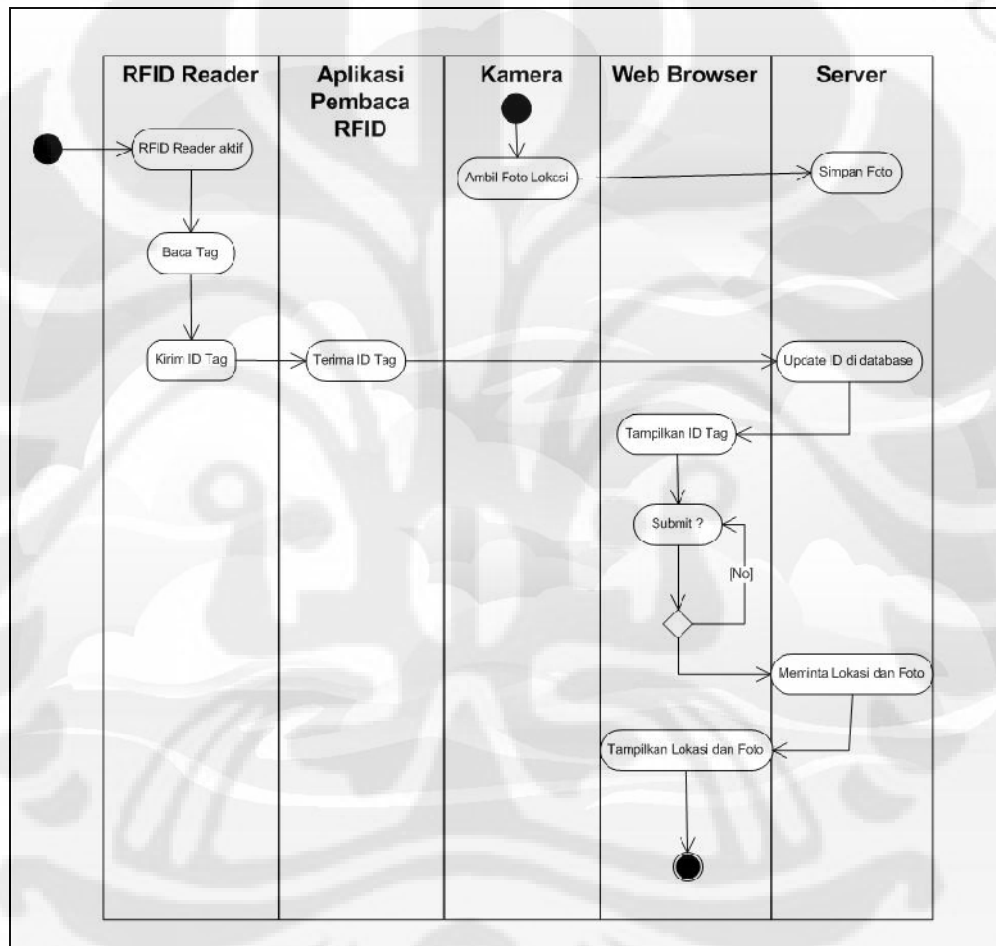
**Activity diagram untuk menampilkan sisa slot parkir**



**Gambar 3.12 – Activity Diagram Slot Parkir dan Log Kendaraan**

Untuk menampilkan berapa banyak parkir yang tersedia dari kendaraan yang masuk ke area parkir, Ketika modul Input Kendaraan dijalankan, modul akan meminta pilihan lokasi di mana kendaraan akan diparkirkan, ketika pilihan telah di-*submit*, *browser* akan melakukan permintaan perubahan status area yang kosong pada database menjadi terisi, kemudian database akan melakukan *query* penghitungan jumlah field yang memiliki status “kosong”. Setelah jumlah field tersebut didapat, maka web browser akan memuat tampilan jumlah *slot* parkir yang masih kosong.

**Activity Diagram untuk Proses Pencarian Letak Kendaraan Menggunakan RFID Tag**



**Gambar 3.13 – Activity Pencarian Letak Kendaraan**

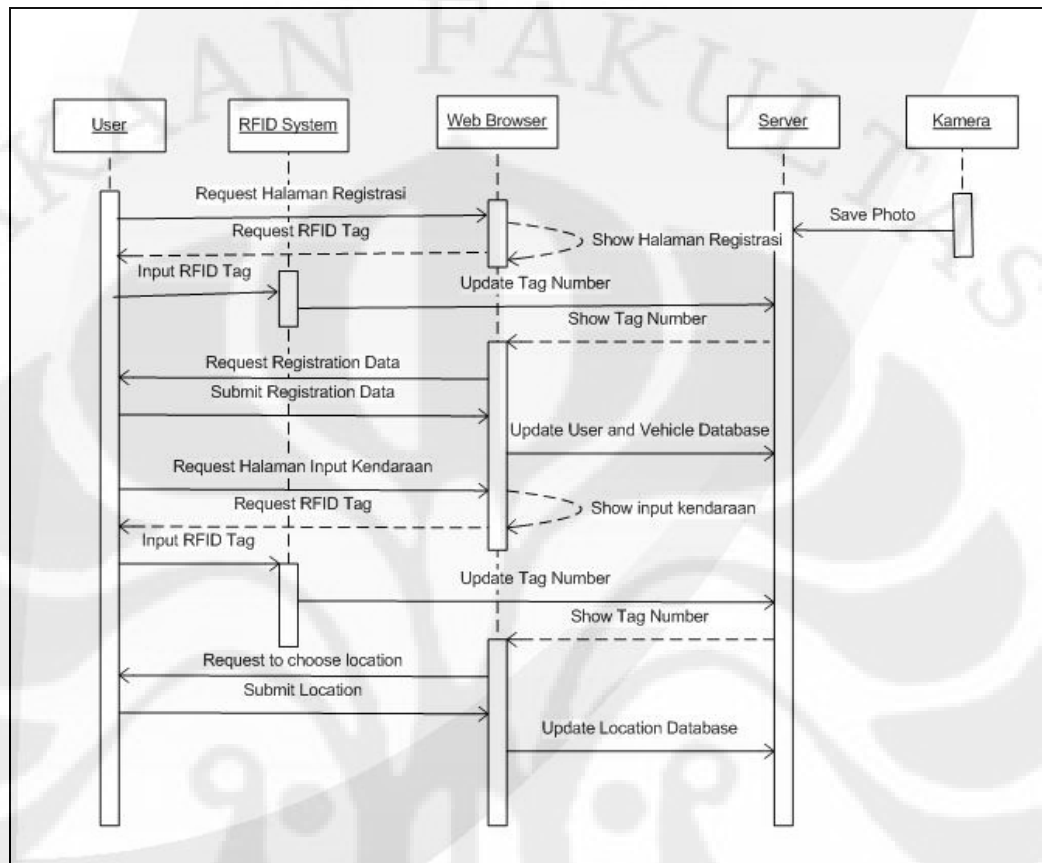
*User* yang ingin melakukan pencarian letak kendaraan, harus terlebih dahulu terdaftar dalam sistem parkir ini, dan memiliki kartu RFID sebagai alat bantu

penelitian. Proses ini diawali dengan aktifnya pembaca RFID dan kamera yang secara terus menerus mengambil foto lokasi dan menyimpannya di server. Untuk mencari lokasi kendaraan saat ini, pembaca RFID akan meminta inputan ID *tag* dari kartu RFID, kemudian mengirim ID tag tersebut ke aplikasi pembaca RFID melalui komunikasi serial. Setelah itu aplikasi akan melakukan cek dalam database apakah sebelumnya terdapat ID tag atau tidak, jika ada, aplikasi akan melakukan perintah menghapus data ID tag pada tabel di database, kemudian akan menuliskan inputan yang terbaru ke dalam tabel. Kemudian *web browser* akan terus menerus melakukan cek pada tabel tersebut, jika terdapat data ID *tag* pada tabel, *web browser* akan menampilkan nomor ID *tag* tersebut. Setelah itu web browser meminta *user* untuk melakukan *submit*, jika tidak ada respon maka foto lokasi parkir tidak akan dimuat di halaman *web*. Jika dilakukan *submit*, browser akan meminta lokasi parkir dan foto yang sesuai dengan inputan RFID tag ke server, untuk kemudian menampilkannya di halaman *web*.

### 3.6.3 Sequence Diagram

*Sequence diagram* sebetulnya tidak jauh berbeda dengan *activity diagram*, hanya saja pada diagram ini, urutan-urutan modul-modul yang dijalankan digambarkan jelas sesuai waktunya. Dari keseluruhan sistem ini, penulis membagi sistem ke dalam tiga *sequence diagram*. Berikut gambar masing-masing bagian beserta penjelasannya.

### Sequence Diagram untuk Registrasi dan Input Kendaraan



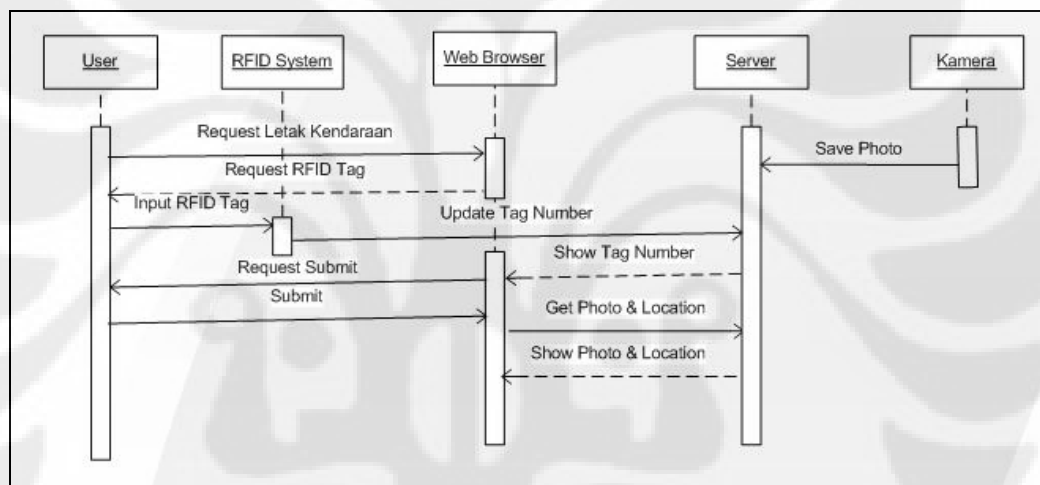
**Gambar 3.14 – Sequence Diagram Registrasi dan Input Kendaraan**

Pengguna parkir yang belum terdaftar dalam sistem parkir ini diharuskan melakukan registrasi terlebih dahulu dengan mengisi data pribadi berupa Nama, Alamat, nomor kendaraan, dan nomor telepon. Urutan registrasi ini diilustrasikan pada gambar 3.x. Pertama, *user* (dalam hal ini *operator*) yang melakukan *request* untuk membuka halaman registrasi akan dimintai inputan RFID Tag. Apabila *user* melakukan input RFID *tag*, sistem RFID akan melakukan *update* database pada tabel yang berisi *field* nomor RFID. Kemudian server akan memerintahkan *web browser* untuk menampilkan Nomor RFID yang sebelumnya telah diinput. Jika nomor RFID telah ditampilkan, *user* akan diminta untuk mengisi data pada *form*

registrasi. Setelah *form* diisi dan di-*submit*, *web browser* akan meminta *server* untuk melakukan *update* data pribadi *user* ke database.

*Operator* akan mengarahkan pengguna parkir yang akan masuk ke area parkir untuk memilih lokasi yang masih kosong. Pertama, *operator* meminta halaman “**Input Kendaraan**”, kemudian *web browser* akan meminta lagi inputan dari RFID tag, lalu sistem memerintahkan *web browser* menampilkan nomor RFID. Setelah itu *web browser* akan meminta *operator* memilih area yang akan diisi oleh kendaraan. Area yang dipilih tersebut akan di-*update* ke dalam database.

### Sequence Diagram untuk Pencarian Letak Kendaraan

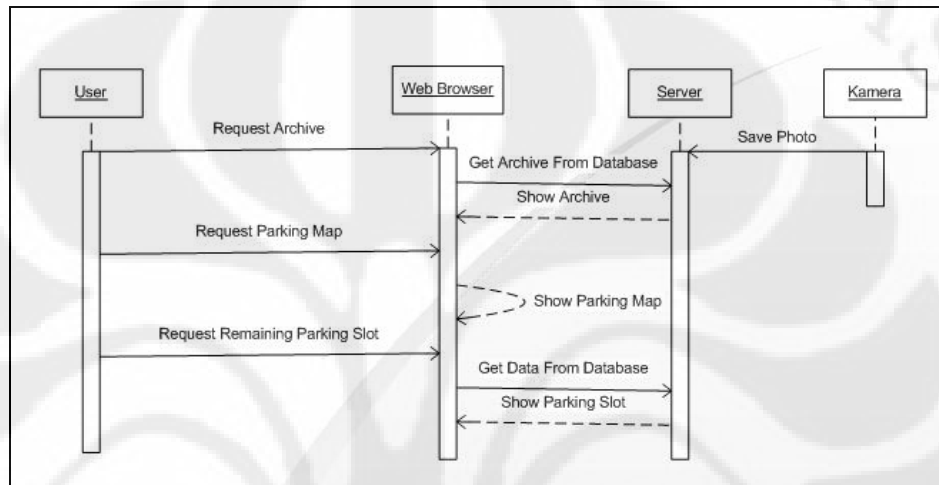


**Gambar 3.15 – Sequence Diagram Letak Kendaraan**

Untuk mencari letak kendaraan, sistem informasi ini akan melalui tahapan-tahapan berikut ini. Pertama, *user* yang ingin mengetahui di mana letak kendaraannya saat ini melakukan *request* halaman “**Letak Kendaraan**”, kemudian *web browser* akan membalas dengan permintaan inputan nomor RFID tag. Setelah *user* melakukan input nomor RFID, aplikasi pembaca RFID akan melakukan update nomor RFID yang dibaca ke dalam database. Setelah itu *server* akan memerintahkan *web browser* untuk menampilkan nomor Tag yang tadi diinput, kemudian *web browser* meminta persetujuan *user* untuk melakukan

pencarian data foto dan lokasi kendaraan saat ini. Jika *user* setuju dan menekan tombol “*submit*”, maka foto lokasi dan deskripsi lokasi kendaraan akan ditampilkan di dalam *web browser*.

**Diagram untuk Melihat Arsip Foto, Log Kendaraan dan Denah Parkir**



**Gambar 3.16 – Sequence Diagram Arsip Foto, Log dan Denah Parkir**

Untuk melihat arsip foto, log dan denah parkir, setiap permintaan ini melalui tahapan yang tidak jauh berbeda, yakni pertama *user* melakukan permintaan halaman, kemudian *web browser* akan memuat data-data yang diperlukan dengan mengambilnya di dalam database, kemudian barulah *request* dari *user* ditampilkan di dalam *web browser*.

### 3.7 Desain Tampilan *User Interface*



**Gambar 3.17 – Desain halaman depan *User Interface***

Gambar di atas menunjukkan desain tampilan *User Interface* dari aplikasi yang dirancang. Gambar yang ditunjukkan di atas merupakan halaman index yang menampilkan fungsi pertama dari sistem informasi perparkiran ini, yaitu berapa banyak lahan parkir yang masih tersedia. Setiap halaman dari *user interface* ini menggunakan *header* yang sama, pada bagian atas terdapat judul, kemudian tulisan-tulisan yang menjadi *link* untuk halaman-halaman yang lain. *Background* tampilan yang digunakan berwarna dominan putih, karena menurut penulis, warna ini adalah warna yang cerah dan tidak menimbulkan kesan yang macam-macam bagi penggunaannya. Desain ini akan terus dikembangkan agar tampilan yang terlihat bisa lebih menarik dan lebih mudah dipergunakan oleh *user*. Kemudahan



disini berarti bahwa pengguna harus dapat dengan mudah memanfaatkan sistem ini walaupun belum pernah menggunakannya sekalipun, untuk itu penempatan-penempatan tampilan halaman dirasa sangat penting dalam mendesain tampilan halaman web.

### 3.8 Rencana Uji Coba Dan Analisa Kinerja Sistem

Sistem ini akan dirancang dengan menggunakan komponen-komponen sebagai berikut : 1 Set PC yang memiliki USB *ports* sebagai *server*, 1 Set PC atau Laptop sebagai *Client*. Kamera dengan konektivitas USB beserta aplikasi kamera dan *driver*-nya, kemudian Tools-tools pemrograman untuk Java seperti Jcreator, aplikasi Web Server yang mendukung JSP, aplikasi *database server*, *web browser* dan *tools* lainnya untuk pengujian aplikasi yang dirancang ini.

Pengujian Sistem ini dibagi menjadi 2 kategori utama, yang pertama yaitu pengujian perangkat keras untuk kamera dan pembaca RFID beserta *tag*-nya, kemudian pengujian perangkat lunak yang melibatkan aplikasi *desktop*, aplikasi web dan juga *database*-nya. Pada pengujian perangkat keras akan dilakukan uji fungsionalitas, mulai dari fungsi dasar perangkat tersebut hingga fungsi yang melibatkan sistem secara keseluruhan. Pada pengujian perangkat lunak, juga akan dilakukan uji fungsionalitas untuk masing-masing modulnya. Selain itu juga akan dilakukan uji kompatibilitas dan juga performa perangkat lunak tersebut.

## BAB 4

### PENGUJIAN DAN EVALUASI KINERJA SISTEM

#### 4.1 Pengujian Fungsional Sistem

Sebelum keseluruhan sistem dapat dijalankan, perlu dilakukan pengujian terhadap setiap aspek penting yang dipergunakan untuk membangun satu sistem ini secara utuh. Yang pertama dilakukan adalah dengan melakukan uji fungsional sistem, ini dilakukan untuk mengetahui apakah fungsi-fungsi dari setiap perangkat yang digunakan bekerja sesuai dengan desain yang telah dibuat. Pengujian ini dilakukan kepada perangkat-perangkat keras dan perangkat-perangkat lunak yang digunakan untuk membangun sistem ini dengan kategori pengujian yang berbeda-beda, hasil yang didapat dari pengujian ini disajikan dalam bentuk tabel-tabel.

##### 4.1.1 Pengujian Fungsional Dasar dari Kamera

Pengujian fungsi-fungsi dasar dari kamera ini dilakukan untuk mengetahui apakah fungsi-fungsi yang dibutuhkan untuk membangun sistem ini bekerja dengan baik atau tidak. Hasil pengujian ini disajikan dalam tabel berikut :

Jenis Kamera	Pengambilan dengan interval	Menyimpan di folder <i>Time Based</i>	Pengambilan berdasarkan pergerakan	Menyimpan di folder <i>Motion Based</i>
Kamera A (1.3 MP) - Prolink PCC1310	Ya	Ya	Ya	Ya
Kamera B (1.3 MP) - Mediatech webcam	Ya	Ya	Ya	Ya
Kamera C (2 MP) - HP Compaq Webcam	Ya	Ya	Ya	Ya

Tabel 4.1 Pengujian Fungsi Dasar Kamera

Terlihat bahwa seluruh fungsi dari kamera yang dibutuhkan untuk menjalankan sistem ini bekerja dengan baik. Meskipun semua fungsi-fungsinya bekerja dengan baik, namun penulis menemukan ketidaknormalan pada kamera B (Mediatech Webcam 1.3 MP). Kamera B tidak dapat menampilkan *output* yang fokus, *output* dari kamera ini terlihat lebih gelap, ini mungkin saja terjadi karena kamera B ini sudah berumur dan lensanya tidak lagi dapat bekerja dengan baik, atau dengan kata lain mengalami kerusakan.

#### **4.1.2 Pengujian Sensitivitas dan Delay Pengambilan Foto oleh Kamera**

Pengujian sensitivitas dan *delay* ini dilakukan untuk mengetahui seberapa besar pengaruh objek yang ditangkap kamera terhadap kemampuan kamera dalam menjalankan fungsinya.

##### **4.1.2.1 Pengujian Sensitivitas Kamera**

Pengujian sensitivitas kamera dilakukan dengan dua objek yang berbeda, objek yang pertama dengan gerakan bolpoin dan yang kedua adalah gerakan tangan, keduanya dipilih karena ukurannya masing-masing berbeda, ini dilakukan untuk mengetahui apakah perbedaan ukuran objek yang bergerak akan mempengaruhi kamera dalam pengambilan foto otomatis berdasarkan pergerakan atau perubahan objek. Kemudian pengujian ini dilakukan dengan tiga cara, yaitu dengan gerakan cepat, lambat dan sangat lambat. Ketiga cara ini dilakukan untuk mengetahui bagaimana respon kamera terhadap kecepatan pergerakan objek. Pengujian ini juga dilakukan pada tiga jarak yang berbeda, ini dilakukan untuk mengetahui bagaimana respon kamera terhadap jauh dekatnya objek yang ditangkap. Pengujian ini dilakukan dalam dua arah pergerakan yang berbeda, yaitu vertikal dan horizontal. Masing-masing pengujian dilakukan sebanyak 10 kali uji gerakan. Hasilnya disajikan dalam table-tabel berikut :

## Arah Horizontal

**Kamera A (Prolink PCC1310 - 1.3 MP)**

Jarak	Bolpoin			Tangan		
	Cepat	Lambat	Sangat lambat	Cepat	Lambat	Sangat lambat
5 cm	0	3	10	4	10	10
10 cm	0	4	10	5	10	10
30 cm	0	0	0	1	10	10
<b>% Berhasil</b>	<b>0%</b>	<b>23.33%</b>	<b>66.67%</b>	<b>33.33%</b>	<b>100%</b>	<b>100%</b>

**Tabel 4.2 Pengujian Sensitivitas Kamera A Arah Horizontal**

Dari hasil pengujian ini terlihat bahwa bolpoin yang merupakan objek berukuran kecil dengan jarak yang jauh dari lensa (30 cm) tidak mampu ditangkap gerakannya oleh kamera, objek berukuran kecil yang bergerak pun tidak mampu ditangkap perubahan gerakannya oleh kamera. Sedangkan tangan yang merupakan objek berukuran besar, gerakannya lebih mampu ditangkap oleh kamera, meskipun sensitivitasnya tidak terlalu baik pada saat mendeteksi gerakan objek yang cepat.

**Kamera B (Mediatech Webcam - 1.3 MP)**

Jarak	Bolpoin			Tangan		
	Cepat	Lambat	Sangat lambat	Cepat	Lambat	Sangat lambat
5 cm	0	0	5	3	10	10
10 cm	0	0	0	0	10	10
30 cm	0	0	0	0	10	10
<b>% Berhasil</b>	<b>0%</b>	<b>0%</b>	<b>20%</b>	<b>10%</b>	<b>100%</b>	<b>100%</b>

**Tabel 4.3 Pengujian Sensitivitas Kamera B Arah Horizontal**

Kamera B memiliki resolusi yang sama dengan Kamera A, hanya saja fokus dari kamera ini sudah tidak berfungsi dengan baik. Sensitivitas kamera ini bisa dibilang cukup buruk, hampir semua gerakan objek berukuran kecil tidak mampu ditangkap oleh kamera ini. Namun demikian, objek berukuran besar masih mampu ditangkap oleh kamera ini pada kecepatan yang lambat.

**Kamera C (HP Compaq Webcam - 2 MP)**

Jarak	Bolpoin			Tangan		
	Cepat	Lambat	Sangat lambat	Cepat	Lambat	Sangat lambat
5 cm	0	5	10	4	9	10
10 cm	0	2	10	4	10	10
30 cm	0	0	0	3	10	10
<b>% Berhasil</b>	<b>0%</b>	<b>23.33%</b>	<b>66.67%</b>	<b>36.67%</b>	<b>96.67%</b>	<b>100%</b>

**Tabel 4.4 Pengujian Sensitivitas Kamera C Arah Horizontal**

Kamera ini memiliki resolusi yang lebih tinggi dengan dua kamera lainnya, namun hal ini tidak terlalu mempengaruhi kemampuannya dalam mendeteksi objek bergerak. Hasil pengujian pada kamera ini tidak berbeda jauh dengan Kamera A.

**Arah Vertikal****Kamera A (Prolink PCC1310 - 1.3 MP)**

Jarak	Bolpoin			Tangan		
	Cepat	Lambat	Sangat lambat	Cepat	Lambat	Sangat lambat
5 cm	0	10	10	4	10	10
10 cm	0	10	10	4	10	10
30 cm	0	0	0	2	10	10
<b>% Berhasil</b>	<b>0%</b>	<b>66.67%</b>	<b>66.67%</b>	<b>33.33%</b>	<b>100%</b>	<b>100%</b>

**Tabel 4.5 Pengujian Sensitivitas Kamera A Arah Vertikal**

Pada arah gerakan vertikal, fungsi kamera dalam mendeteksi gerakan ternyata tidak terlalu berbeda, bisa terlihat dari hasil pengujian, hasil yang didapat dari Kamera A pada gerakan horizontal tidak berbeda jauh dengan hasil pada gerakan vertikal.

**Kamera B (Mediatech Webcam - 1.3 MP)**

Jarak	Bolpoin			Tangan		
	Cepat	Lambat	Sangat lambat	Cepat	Lambat	Sangat lambat
5 cm	0	0	0	0	10	10
10 cm	0	0	0	0	0	2
30 cm	0	0	0	0	0	0
<b>% Berhasil</b>	<b>0%</b>	<b>0%</b>	<b>0%</b>	<b>0%</b>	<b>33.33%</b>	<b>40%</b>

**Tabel 4.6 Pengujian Sensitivitas Kamera B Arah Vertikal**

Sekali lagi dalam pengujian ini terlihat bahwa Kamera B yang fokusnya tidak berfungsi dengan baik, memiliki sensitivitas yang buruk dalam hal pendeteksian gerakan, gerakan objek berukuran kecil (bolpoin) sama sekali tidak terdeteksi pada gerakan vertikal.

**Kamera C (HP Compaq Webcam - 2 MP)**

Jarak	Bolpoin			Tangan		
	Cepat	Lambat	Sangat lambat	Cepat	Lambat	Sangat lambat
5 cm	0	1	10	2	10	10
10 cm	0	2	9	3	10	10
30 cm	0	0	0	5	10	10
<b>% Berhasil</b>	<b>0%</b>	<b>10%</b>	<b>63.33%</b>	<b>33.33%</b>	<b>100%</b>	<b>100%</b>

**Tabel 4.7 Pengujian Sensitivitas Kamera C Arah Vertikal**

Gerakan objek pada arah vertikal mampu ditangkap dengan baik oleh Kamera C, hasil yang ditunjukkan tidak jauh berbeda dengan hasil pergerakan objek pada arah horizontal.

Dari hasil pengujian pada ketiga kamera, dapat ditarik kesimpulan bahwa besar kecilnya ukuran objek, mempengaruhi fungsi kamera dalam pengambilan foto otomatis berdasarkan pergerakan atau perubahan objek. Semakin besar objek yang disorot, maka kamera akan semakin sensitif terhadap pergerakannya, ini dikarenakan cara kamera dalam mendeteksi pergerakan adalah dengan mendeteksi perubahan dari titik-titik objek. Semakin banyak titik-titik objek yang berubah, semakin besar peluang untuk mendeteksi gerakan objek, karena semakin besar objeknya, semakin banyak titik-titik yang dideteksi oleh kamera tersebut.

Perbedaan jarak juga mempengaruhi fungsi kamera dalam pengambilan foto, karena kamera memandang objek yang jauh sebagai titik-titik yang kecil. Selain itu juga semakin cepat gerakan objek, semakin sulit pergerakan tersebut untuk dideteksi oleh kamera, begitu juga sebaliknya, semakin lambat gerakan objek, semakin mudah kamera mendeteksi pergerakan objek tersebut.

Hal lain yang cukup berpengaruh adalah kemampuan fokus dari lensa, terlihat pada kamera B banyak pergerakan yang tidak mampu terdeteksi, karena kemampuan fokus lensa dari kamera B tidak lagi dapat bekerja sebagaimana mestinya, ini artinya semakin tidak fokus objek yang disorot oleh kamera, semakin sulit pergerakannya dideteksi oleh kamera. Kemudian arah pergerakan vertikal dan horizontal tidak terlalu berpengaruh terhadap kemampuan kamera mendeteksi pergerakan objek.

#### 4.1.2.2 Uji Delay Pengambilan Foto

Pengujian ini dilakukan untuk mengetahui berapa banyak waktu yang dibutuhkan oleh kamera untuk melakukan pengambilan foto otomatis berdasarkan pergerakan objek setelah pengambilan foto sebelumnya. Interval waktu yang diambil adalah kurang dari 1 detik dan lebih dari 1 detik, karena rata-rata kamera web dapat mengambil foto objek dengan interval waktu 0,5 detik dari pengambilan sebelumnya. Pengujian ini juga masing-masing dilakukan sebanyak 5 kali pergerakan objek.

Jenis Kamera	Waktu (detik)		% Berhasil
	< 1	> 1	
Kamera A (1.3 MP) (Prolink PCC1310)	5	5	100%
Kamera B (1.3 MP) (Mediatech Webcam)	5	5	100%
Kamera C (2 MP) (HP Compaq Webcam)	1	5	60%

**Tabel 4.8 Uji Delay Pengambilan Foto**

Dari hasil pengujian dapat disimpulkan bahwa, besarnya resolusi kamera dapat mempengaruhi *delay* dalam pengambilan foto otomatis, karena semakin besar resolusi kamera, objek yang ditangkap oleh kamera dipandang memiliki lebih banyak titik-titik, sehingga ketika deteksi gerakan terjadi, kamera membutuhkan waktu yang lebih lama untuk kembali ke kondisi awal sebelum melakukan pengambilan foto lagi.

#### 4.1.3 Pengujian Fungsionalitas RFID Reader dan RFID Tag

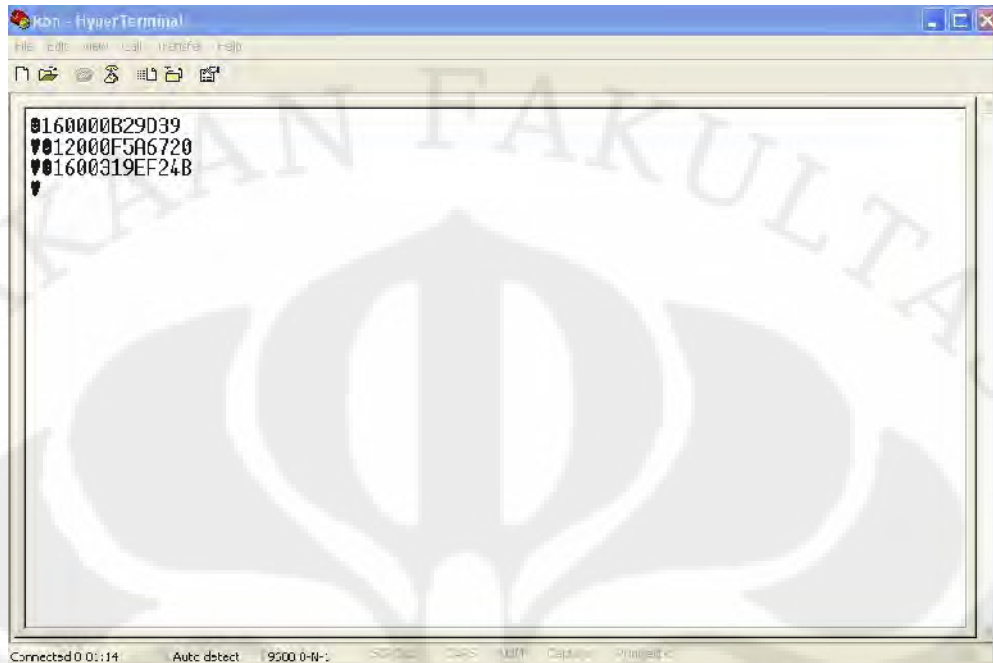
Pengujian ini dilakukan untuk mengetahui apakah kedua divais ini dapat menjalankan fungsi-fungsi yang dibutuhkan oleh sistem ini nantinya. Hasil pengujian ini disajikan dalam tabel berikut :

Deskripsi	Keterangan
Reader Membaca Tag	Ya
Modul Converter RS232 - USB berfungsi	Ya
Data RFID terbaca oleh Program Desktop	Ya
Data RFID tersimpan di database	Ya

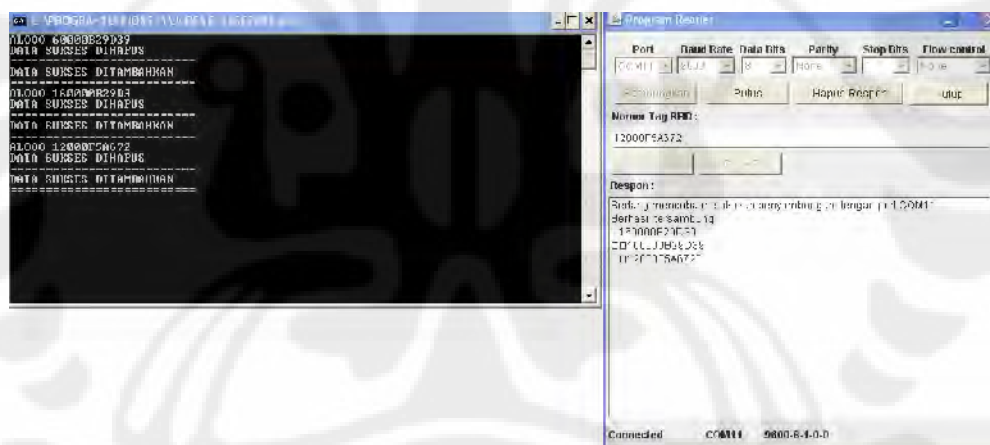
**Tabel 4.9 Uji Fungsionalitas RFID Reader dan RFID Tag**

Pada pengujian ini terdapat modul tambahan berupa *converter RS232 to USB*, *converter* ini diperlukan untuk komunikasi serial pada PC yang tidak memiliki port serial RS232 tetapi memiliki port USB. Jika pembaca RFID berhasil membaca RFID tag, pada *hyperterminal* akan didapat keluaran seperti pada gambar 4.1. Kemudian apabila Data RFID tag terbaca oleh program *desktop*, akan didapat keluaran seperti yang ditunjukkan pada gambar 4.2.





**Gambar 4.1 Output Pembacaan RFID Tag pada Hyper Terminal**



**Gambar 4.2 Output Pembacaan RFID Tag pada Program Desktop**

#### 4.1.4 Pengujian Fungsionalitas Dasar Perangkat Lunak

Pengujian fungsionalitas ini dilakukan untuk mengetahui apakah fungsi-fungsi dari perangkat lunak yang dibutuhkan untuk membangun sistem ini secara utuh berjalan baik atau tidak. Hasil pengujian ini disajikan dalam tabel berikut :

Deskripsi	Keterangan
Database berfungsi sesuai relasinya	Ya
Modul <i>slot</i> parkir dapat ditampilkan sesuai jumlahnya	Ya
Data nomor tag pada program desktop sesuai dengan yang ditampilkan pada halaman letak kendaraan, input kendaraan, dan registrasi	Ya
Output Foto area pada halaman letak kendaraan sesuai dengan inputan nomor kendaraan dan inputan nomor tag	Ya
Data parkir kendaraan yang telah masuk dan keluar ditampilkan sesuai keadaan	Ya
Foto arsip beserta tanggalnya dapat ditampilkan di halaman Time Based dan Motion Based	Ya

**Tabel 4.10 Pengujian Fungsionalitas Dasar Perangkat Lunak**

Dari hasil pengujian fungsionalitas ini, seluruh fungsi yang diinginkan berjalan dengan baik, artinya perangkat lunak ini siap untuk mendukung berjalannya sistem informasi ini.

#### 4.2 Pengujian Kompatibilitas dan Performa Halaman Web

Terkadang seorang pengguna aplikasi menginginkan aplikasi ini berjalan pada *platform* yang sesuai dengan seleranya, karenanya penulis merasa perlu untuk melakukan pengujian kompatibilitas, sehingga dapat diketahui apakah sistem informasi ini dapat berjalan dengan baik di berbagai *platform* atau tidak. Pengujian dilakukan pada tiga *platform hardware* yang berbeda dan lima *platform software* yang berbeda. Adapun spesifikasi dari masing-masing *platform hardware* yang digunakan yaitu sebagai berikut :

Server (Localhost) :	
Processor	: Athlon 64 X2 Dual-Core QL-64 2.1 GHz
RAM	: 1 GB DDR2 PC 5300
Storage Media	: 250GB SATA 5400 rpm
Screen Resolution	: 1366 x 768
NIC	: Gigabit Ethernet
Operating System	: Windows XP Professional SP2
Client 1 :	
Processor	: Intel Celeron M ULV 900 MHz
RAM	: 1 GB DDR2 PC 5300
Storage Media	: 4 GB SSD
Screen Resolution	: 800 x 600
NIC	: 10/100 Mbps
Operating System	: Windows XP Professional SP2
Client 2 :	
Processor	: Athlon 64 X2 5000+ Dual-Core 3.12 GHz
RAM	: 2 GB DDR2 PC 7000
Storage Media	: 500GB SATA 7200 rpm
Screen Resolution	: 1680x 1050
NIC	: 10/100 Mbps
Operating System	: Windows XP Professional SP2

Kemudian *platform software* yang digunakan dalam pengujian ini di antaranya, *web browser Mozilla Firefox, Internet Explorer, Opera, Safari dan Google Chrome*. Masing-masing merupakan *web browser* yang dikembangkan oleh *software developer* yang berbeda. Selain itu juga, uji performa dilakukan pada sistem ini dengan mengukur seberapa besar *load time* yang dibutuhkan untuk membuka suatu halaman. Untuk pengukuran halaman dipergunakan Fungsi berikut yang dimasukkan ke dalam *script* program di setiap halamannya :

```

<%
long t0 = System.currentTimeMillis();
long tn0 = System.nanoTime();
%>
<html>
    <!-- kode-kode JSP untuk halaman web -->
</html>
<%
long tn1 = (System.nanoTime() - tn0)/1000;
long t1 = System.currentTimeMillis() - t0;
%>
<p>t(mili) : <%=t1%><br />
t(micro): <%=tn1%></p>

```

Fungsi-fungsi tersebut diletakkan di setiap halaman pada bagian atas tag <html> dan bagian bawah tag </html>. *Load time* diukur dengan satuan milisekon dan mikrosekon. Namun karena hasil yang ditunjukkan satuan milisekon terdapat anomali, maka satuan mikrosekon yang dijadikan acuan. Hasil pengujian ini disajikan dalam tabel-tabel di bawah ini :

#### Mozilla Firefox (Versi 3.5.5)

Halaman	Susunan Halaman	Load Time					
		Localhost		Client-1		Client-2	
	Tersusun Rapi	ms	µs	ms	µs	ms	µs
Index	Ya	16	13537	15	13496	15	7964
Letak Kendaraan	Ya	16	16410	0	12277	0	8124
Input Kendaraan	Ya	31	22241	16	12256	0	8437
Registrasi	Ya	15	14335	0	10441	16	7001
Arsip	Ya	0	591	0	741	0	711
Data Parking In	Ya	0	13560	16	12095	0	7110
Data Parking out	Ya	16	11422	16	15047	0	10196
TBC A	Ya	0	8093	16	8010	16	7260
TBC B	Ya	16	7179	16	7852	0	7326
MBC A	Ya	16	7777	0	10375	0	8211
MBC B	Ya	0	7965	16	9952	15	8453
Lihat Denah	Ya	0	1663	0	623	0	702
Rata-rata		10.5	10397.75	9.25	9430.417	5.166667	6791.25

**Tabel 4.11 Pengujian Kompatibilitas dan Performa Halaman Web pada Mozilla Firefox**

*Browser Mozilla Firefox* dapat menjalankan sistem informasi ini dengan baik, susunan halaman tiap halaman web pun tersusun rapi, karena *browser* ini memang dijadikan referensi oleh penulis pada tahap desain halaman dan uji coba awal. Rata-rata *load time* yang paling baik adalah pada *client-2* hal ini dimungkinkan karena spesifikasi komputer yang digunakan memang lebih tinggi dibanding *server* dan *client 1*, sehingga transfer data yang terjadi bisa lebih cepat.

#### Internet Explorer (Versi 6.0)

Halaman	Susunan Halaman	Load Time					
		Localhost		Client-1		Client-2	
	Tersusun Rapi	ms	$\mu$ s	ms	$\mu$ s	ms	$\mu$ s
Index	Tidak	16	12193	16	13340	0	8290
Letak Kendaraan	Tidak	0	12944	16	12702	16	11930
Input Kendaraan	Tidak	16	8591	15	8319	16	8479
Registrasi	Tidak	0	6952	0	6817	0	6576
Arsip	Tidak	0	1009	15	688	0	664
Data Parking In	Tidak	15	11496	16	14626	16	7424
Data Parking out	Tidak	0	7535	16	11195	0	6539
TBC A	Tidak	141	136875	0	8394	16	15250
TBC B	Tidak	0	7511	15	12840	15	7616
MBC A	Tidak	0	7044	0	10530	15	9405
MBC B	Tidak	0	7470	0	11638	16	10317
Lihat Denah	Tidak	0	674	0	748	0	667
Rata-rata		15.6667	18357.83	9.333	9319.75	9.1667	7763.083

**Tabel 4.12 Pengujian Kompatibilitas dan Performa Halaman Web pada**

#### Internet Explorer

*Internet Explorer 6.0* ternyata tidak dapat menampilkan halaman web pada sistem informasi ini dengan baik, tidak ada satupun susunan halaman yang sesuai dengan desain yang telah dirancang penulis. Performanya pun lebih rendah dari web browser yang lain, terlihat pada sisi *server*, rata-rata *load time* yang

dibutuhkan sebesar 18537 mikrosekond, angka yang cukup besar untuk proses *load time* halaman web.

**Opera (Versi 10.10)**

Halaman	Susunan Halaman	Load Time					
		Localhost		Client-1		Client-2	
	Tersusun Rapi	ms	µs	ms	µs	ms	µs
Index	Tidak	16	8018	16	9957	0	8327
Letak Kendaraan	Ya	0	8496	15	7750	16	7935
Input Kendaraan	Ya	16	10513	0	12106	16	7143
Registrasi	Ya	0	6806	0	6157	15	6415
Arsip	Ya	16	4617	0	710	0	628
Data Parking In	Ya	16	13817	16	6602	15	6350
Data Parking out	Ya	0	6829	16	5926	0	6616
TBC A	Ya	0	7708	0	7038	15	13005
TBC B	Ya	15	11558	16	7164	0	7155
MBC A	Ya	16	8192	31	26516	16	8976
MBC B	Ya	0	7267	15	9255	15	8891
Lihat Denah	Ya	0	582	0	628	0	641
Rata-rata		7.9166667	7866.917	10.4167	8317.417	9	6840.167

**Tabel 4.13 Pengujian Kompatibilitas dan Performa Halaman Web pada**

**Opera**

Seperti halnya *Mozilla Firefox*, *Browser Opera* dapat menjalankan sistem informasi ini dengan baik, susunan halaman tiap halaman web pun tersusun rapi, namun ada salah satu bagian pada halaman index yang tampilannya tidak sesuai desain awal, yaitu pada bagian sisa slot parkir. Rata-rata *load time* yang paling baik juga ada pada *client-2* karena spesifikasi komputer yang digunakan memang lebih tinggi dibanding *server* dan *client 1*.

### Safari (Versi 4.04)

Halaman	Susunan Halaman	Load Time					
		Localhost		Client-1		Client-2	
	Tersusun Rapi	ms	$\mu$ s	ms	$\mu$ s	ms	$\mu$ s
Index	Ya	15	10438	16	8200	15	8119
Letak Kendaraan	Ya	0	7727	16	15436	15	7378
Input Kendaraan	Ya	15	15054	0	633	0	7798
Registrasi	Ya	0	8798	0	6334	0	6170
Arsip	Ya	0	472	16	6512	0	712
Data Parking In	Ya	16	6696	0	7029	0	7178
Data Parking out	Ya	0	6498	0	7214	15	6008
TBC A	Ya	16	7233	16	9005	16	14833
TBC B	Ya	15	7441	0	7696	0	7030
MBC A	Ya	0	7074	16	9123	15	8844
MBC B	Ya	15	13596	16	9400	15	8691
Lihat Denah	Ya	0	471	0	655	0	615
Rata-rata		7.6667	7624.833	8	7269.75	7.583	6948

**Tabel 4.14 Pengujian Kompatibilitas dan Performa Halaman Web pada**

#### Safari

Browser yang dibuat oleh Apple ini ternyata mampu menampilkan halaman web sistem informasi ini dengan baik tanpa ada kendala yang berarti. Rata-rata *load time* yang paling baik juga ada pada *client-2* karena spesifikasi komputer yang digunakan memang lebih tinggi dibanding *server* dan *client 1*, sehingga transfer data yang terjadi bisa lebih cepat.

### Google Chrome (Versi 4.0.249.30)

Halaman	Susunan Halaman	Load Time					
		Localhost		Client-1		Client-2	
	Tersusun Rapi	ms	$\mu$ s	ms	$\mu$ s	ms	$\mu$ s
Index	Ya	0	10912	15	9471	0	8054
Letak Kendaraan	Ya	16	8803	15	7712	0	7376
Input Kendaraan	Ya	16	18070	0	7519	15	7359
Registrasi	Ya	0	6536	0	12777	16	6211
Arsip	Ya	0	412	0	633	0	693
Data Parking In	Ya	15	9927	0	6938	0	6499
Data Parking out	Ya	0	6991	0	6484	0	6417
TBC A	Ya	0	7197	16	7304	16	13403
TBC B	Ya	16	7741	0	6954	0	6943
MBC A	Ya	0	6895	0	9329	0	8305
MBC B	Ya	15	18759	16	8825	16	8778
Lihat Denah	Ya	15	432	0	699	0	619
Rata-rata		7.75	8556.25	5.1666667	7053.75	5.25	6721.417

**Tabel 4.15 Pengujian Kompatibilitas dan Performa Halaman Web pada Google Chrome**

Browser yang dibuat oleh Google ini juga ternyata mampu menampilkan halaman web sistem informasi ini dengan baik. Rata-rata *load time* yang paling baik juga ada pada *client-2* karena spesifikasi komputer yang digunakan memang lebih tinggi dibanding *server* dan *client 1*, sehingga transfer data yang terjadi bisa lebih cepat.

Dari hasil pengujian yang diperoleh, diketahui bahwa ada dua *platform software* yang tingkat kompatibilitasnya kurang dengan sistem informasi ini. Yang paling terlihat yaitu pada *internet explorer*, sangat jelas terlihat ketidakteraturan halaman pada *platform software* ini, ini mungkin terjadi karena sistem informasi ini dibangun menggunakan *software* yang lebih *up-to-date*



daripada *platform software* yang digunakan untuk pengujian. *Internet Explorer* yang digunakan adalah versi 6.0 yang memang dikeluarkan pada sekitar tahun 2004, sedangkan *web browser* lain merupakan *software* keluaran terbaru. Selain pada *internet explorer*, *web browser Opera* juga kurang kompatibel dengan sistem informasi ini, walaupun tidak signifikan, karena hanya satu modul saja yang tidak berjalan sebagaimana mestinya.

Dalam pengujian performa (*load time*) halaman web, ditemukan kegagalan pada *load time* dengan satuan waktu milisekon, karenanya, waktu mikrosekon lah yang dijadikan acuan pada pengujian performa ini. Perbedaan *platform*, baik itu pada *platform software* maupun *platform hardware*, tidak menunjukkan perbedaan yang signifikan dalam hal performa. Jadi dari hasil pengujian ini, menunjukkan bahwa, perbedaan *platform* pengujian tidak mempengaruhi performa dari halaman web.

#### **4.3 Perbandingan dengan Sistem RFID yang Telah Ada**

Untuk membedakan sistem ini dengan sistem RFID lain yang sudah ada dilakukan studi evaluasi perbandingan. Perbandingan dilakukan dengan mempelajari secara singkat karya tulis mengenai sistem berbasis RFID yang penulis ketahui. Karya tulis yang diambil adalah karya tulis yang dibuat oleh beberapa mahasiswa Departemen Teknik Elektro FTUI yang terdokumentasi secara online.

Karya tulis yang dijadikan pembanding adalah:

**FHS2008** – Fadhly H. Saputra, “*Sistem Absensi Menggunakan Teknologi RFID*”.

**MAH2008** – Mahadhir, “*Rancang Bangun Sistem Identifikasi Kendaraan Pada Akses Masuk Menggunakan Teknologi RFID*”.

**AAD2009** – Adytiawan A. Dwitama, “*Perancangan dan Implementasi Sistem Parkir Berbasis RFID Dengan Menggunakan Antarmuka Java dan*

Universitas Indonesia

*Basis Data MySQL Untuk Diimplementasikan Pada Lingkungan Parkir FTUI”.*

**TAZ2009** – Tommy A. Zairi, “*Rancang Bangun Sistem Parkir Otomatis Berbasis Teknologi RFID Reader DL-910 dan Tag Pasif EPC Gen2 Dengan Fitur Sistem Debit Biaya Parkir Via SMS*”.

Sedangkan sistem ini mempunyai kode **EKR2009**.

	MAH2008	AAD2009	TAZ2009	FHS2008	EKR2009
Aplikasi Sistem	Identifikasi Kendaraan	Sistem Parkir	Sistem Parkir	Sistem Absensi	Sistem Parkir
Basis Komunikasi	Serial	Serial	Serial	Serial	Serial
Tipe Pembaca RFID	PF 5210	ID-12	PF5210	ID-12	ID-12
Tipe Tag	Tag Aktif PF-300	Tag pasif ukuran ID-1	Tag Pasif EPC Gen2	Passive Tag Mifare UL	Tag pasif ukuran ID-1
Hardware Tambahan	Mikrokontroler Atmega8535, LDR	-	Mikrokontroler Atmega8535, Motor Servo	Mikrokontroler AT89S52	Kamera Web
Bahasa Pemrograman	Visual Basic	Java	Delphi 7	Delphi 5	Java, JSP
Tipe Pemrograman	Aplikasi	Aplikasi	Aplikasi	Aplikasi	Aplikasi Web
Fitur Tambahan	Sensor	-	SMS	-	Motion Detection

**Tabel 4.16 Perbandingan dengan Sistem RFID yang telah ada**

Sistem ini adalah sistem yang menggunakan protokol komunikasi Serial. Pembaca RFID yang digunakan sederhana dengan kemampuan baca maksimal 12 cm. Tag yang dipakai adalah tag pasif berukuran ID-1 (berdimensi 85.60 × 53.98 mm). Program berupa program *desktop* yang dihubungkan dengan web melalui koneksi *database*.

Hal yang membuat sistem ini berbeda dengan sistem lain program utamanya berupa web yang dibangun dengan bahasa pemrograman JSP. Kemudian tipe program yang berupa halaman web dinamis yang dikoneksikan dengan program *desktop* Java melalui *database* yang sama. Selain itu sistem ini memiliki perangkat tambahan berupa kamera dengan fungsi uniknya yaitu *motion detection*.



Universitas Indonesia

## BAB 5 KESIMPULAN

1. Sistem terdiri atas dua bagian yaitu sisi perangkat keras dan sisi perangkat lunak.
2. Sisi perangkat keras berupa sistem RFID yang menggunakan komunikasi serial untuk pengiriman informasinya.
3. Sisi perangkat lunak dikembangkan dengan program *desktop* berbasis Java untuk komunikasi dan pemrosesan data serta JSP untuk halaman web dinamis dan MySQL untuk *database*.
4. Sensitivitas Kamera dalam mendeteksi objek yang bergerak dipengaruhi oleh beberapa hal di antaranya Besar kecilnya ukuran objek, perbedaan jarak, kecepatan gerak objek, dan tingkat fokus kamera. Dari 10 kali pengujian, hanya pada pengujian objek berukuran besar yang dapat 100% dideteksi gerakannya.
5. Semakin besar resolusi kamera, semakin besar juga delay yang dibutuhkan untuk melakukan pengambilan foto berikutnya. *Delay* minimum sebelum pengambilan gambar berikutnya adalah 0.5 detik.
6. Diperlukan *converter* usb to RS232 untuk komunikasi serial pada PC yang tidak memiliki port serial RS232 tetapi memiliki port USB.
7. Dari lima *web browser* yang digunakan dalam pengujian sistem ini, dua di antaranya kurang kompatibel dengan sistem ini.
8. Pada pemrograman Java, pengukuran waktu untuk mengukur kode program lebih baik menggunakan fungsi perhitungan waktu yang menggunakan satuan yang lebih teliti yaitu mikrosekond.
9. Waktu *load* halaman web dalam sistem ini berkisar antara 6000 sampai 18000 mikrosekond. Waktu *load* tidak bergantung pada *platform hardware* yang digunakan.

## DAFTAR ACUAN

- [1] Christian Hartono, “Perancangan dan pembuatan aplikasi sistem informasi sepak bola berbasis web” – Jurusan Teknik Informatika Fakultas Teknologi Industri Universitas Kristen Petra Surabaya, 2005. Diakses 16 Mei 2009.  
<http://digilib.petra.ac.id/jiunkpe/s1/info/2006/jiunkpe-ns-s1-2006-26401224-3879-football-chapter1.pdf>
- [2] Arie Wahyu Hidayat, “Proyek Rekayasa Perangkat Lunak – Smart Bathroom”, 2006.
- [3] Bernard Liandie, “Perancangan Aplikasi Mobile Learning Pada Sisi Client Dengan Pendekatan Suatu Model Perkuliahan Dalam Kelas Di Lingkungan Universitas Menggunakan Java 2 Micro Edition”, 2007
- [4] Sun Microsystems, “Java Server Pages Overview”. Diakses 15 Mei 2009 dari Sun Microsystems.  
<http://java.sun.com/products/jsp/overview.html#1>
- [5] Sri Dharwiyanti, “Pengantar Unified Modeling Language (UML)”, 2003. Diakses 10 Mei 2009.  
<http://setia.staff.gunadarma.ac.id/Downloads/files/6039/MateriSuplemenUml.pdf>

## DAFTAR PUSTAKA

Dwitama, Adytiawan Arga “Perancangan Dan Implementasi Sistem Parkir Berbasis Rfid Dengan Menggunakan Antarmuka Java Dan Basis Data Mysql Untuk Diimplementasikan Pada Lingkungan Parkir Ftui” Skripsi, Departemen Teknik Elektro Fakultas Teknik Universitas Indonesia, 2009.

Hartono, Christian. “Perancangan dan pembuatan aplikasi sistem informasi sepak bola berbasis web” Skripsi, Jurusan Teknik Informatika Fakultas Teknologi Industri Universitas Kristen Petra Surabaya, 2005. Diakses 16 Mei 2009.  
<http://digilib.petra.ac.id/jiunkpe/s1/info/2006/jiunkpe-ns-s1-2006-26401224-3879-football-chapter1.pdf>

Hijrah Saputra, Fadhly. “Sistem Absensi Menggunakan Teknologi RFID“ Tugas Akhir, Departemen Teknik Elektro Fakultas Teknik Universitas Indonesia, 2008.

Ichtiara, Cita “Implementasi Aplikasi Sistem Informasi Geografis (SIG) Universitas Indonesia (UI) Berbasis Web Dengan Menggunakan Google Maps” Skripsi, Departemen Teknik Elektro Fakultas Teknik Universitas Indonesia, 2008.

Liandie, Bernard “Perancangan Aplikasi Mobile Learning Pada Sisi Client Dengan Pendekatan Suatu Model Perkuliahan Dalam Kelas Di Lingkungan Universitas Menggunakan Java 2 Micro Edition”, Skripsi, Departemen Teknik Elektro Fakultas Teknik Universitas Indonesia, 2007.

Kadir, Abdul. , *Dasar Pemrograman Web Dinamis dengan JSP (Java Server Pages)*. (Bandung : Penerbit Andi, 2004).

## LAMPIRAN

### Lampiran 1: Source Code Halaman Web

=====koneksi ke database=====

```
<% @page import="java.sql.*"%>
<% @ page import="java.io.*,
java.util.Date,java.text.DateFormat,java.text.SimpleDateFormat,
java.text.DateFormatSymbols,java.util.Calendar,java.util.GregorianCalendar" %>

<%
    Class.forName("com.mysql.jdbc.Driver");
    Connection con =
    DriverManager.getConnection("jdbc:mysql://localhost:3306/datapeta","root","123
");
    Statement stm = con.createStatement();
%>
```

=====Halaman Index=====

```
<% @ page import="java.sql.*,java.util.Calendar,java.util.GregorianCalendar,
java.util.Date,java.text.DateFormat,java.text.SimpleDateFormat,
java.text.DateFormatSymbols" %>
    <font size="5"> <%! public String
infoWaktu(GregorianCalendar cal) {
    String hari[] = {"Dummy", "Minggu", "Senin", "Selasa", "Rabu", "Kamis",
"Jumat", "Sabtu"};
    String bulan[] = {"Januari", "Pebruari", "Maret", "April", "Mei", "Juni",
"Juli", "Agustus", "September", "Oktober", "November", "Desember"};
    return("<B><center><font color=green>" +
hari[cal.get(Calendar.DAY_OF_WEEK)] +" </font><font color=green> / "
```

```

+cal.get(Calendar.DATE) + " " +bulan[cal.get(Calendar.MONTH)] + " " +
cal.get(Calendar.YEAR)+"</B></font><HR>");}

%>
<%GregorianCalendar sekarang =new
GregorianCalendar();out.println(infoWaktu(sekarang) + "<BR>");%>
<font size="+1"><h2><a href="#" title=" " class="style4"
rel="bookmark">PARKIR TERSEDIA </a></h2></font>
<a class="thumb" href="#" rel="bookmark" title=""></a>

<% @ include file="koneksi.jsp" %>
<%
boolean cek=false;
String tglku = new SimpleDateFormat("yyyy/MM/dd").format(new Date());
String jamku = new SimpleDateFormat("hh:mm:ss").format(new Date());

ResultSet hasil=null;
try{hasil=stm.executeQuery("SELECT count(status) as jk
FROM area_parkir where status='kosong');}
catch(Exception err){out.println("Kesalahan="+err);cek =
true;}

if(!cek){
try {
while(hasil.next()){
String jken=hasil.getString("jk"); //ahiran t=data diambil dari tabel langsung
out.println("<div align = center> <font size =15><font color =red><h1> <B>"+
jken + " </font></font></h1></div>");
}
}
catch(Exception err){out.println("Kesalahan="+err);cek = true;}

```



```

    }
%>

=====Halaman Letak Kendaraan=====
adonly="1" value=<%= snotag %> >
    <%
cek=false;
hasil=null;

        try{
            if(snotag==""){hasil=stm.executeQuery("SELECT
* FROM tbl_parkir where notag like '"+snotag+"'");}
            else{hasil=stm.executeQuery("SELECT *
FROM tbl_parkir where notag like '%" +snotag+'%");}
        }
        catch(Exception err){out.println("Kesalahan="+err);cek =
true;}

        if(!cek){
            try {
                int no=0;
                int x;

                while(hasil.next()){
                    no=hasil.getInt("id");

                    strjam=hasil.getString("jamt");
                    strtgl=hasil.getString("tgl");
                    strnamafoto=hasil.getString("namafotot");
                    snotag=hasil.getString("notag");
                    nokendaraan=hasil.getString("nokendaraan");
                    keterangan=hasil.getString("keterangan");
                    posisi=hasil.getString("posisi");
                }
            } //milik try

```

```

        catch(Exception err){ cek = true;}
    }
%>
    </span></span>        </form></td>
</tr>
<tr>
    <td>Nomor Kendaraan </td>
    <td><form name="form2" id="form2" method="post"
action="Letak%20Kendaraan.jsp">
        <input name="nokendaraan" type="text" id="nokendaraan" value ="<%=
nokendaraan %>"/>
        <input type="submit" name="Submit" value="Submit" />
        <span class="meta"><span class="sidebar style1">
        <%

//-----
cek=false;
String alamat="";
hasil=null;
        try{
            hasil=stm.executeQuery("SELECT * FROM
tbl_parkir where nokendaraan='"+nokendaraan+"'");
        }
        catch(Exception err){out.println("Kesalahan="+err);cek =
true;}
        if(!cek){
            try {
                int no=0;
                int x;
                while(hasil.next()){

```

```

no=hasil.getInt("id");

strjam=hasil.getString("jamt");
strtgl=hasil.getString("tgl");
strnamafoto=hasil.getString("namafotot");
snotag=hasil.getString("notag");
nokendaraan=hasil.getString("nokendaraan");
keterangan=hasil.getString("keterangan");
posisi=hasil.getString("posisi");

if(posisi.substring(0,6).equalsIgnoreCase("Area
A")){alamat="Kamera_A/Foto_A.jpg";}
else if(posisi.substring(0,6).equalsIgnoreCase("Area
B")){alamat="Kamera_B/Foto_B.jpg";}
else if(posisi.substring(0,6).equalsIgnoreCase("Area
C")){alamat="Kamera_C/Foto_C.jpg";}
else if(posisi.substring(0,6).equalsIgnoreCase("Area
D")){alamat="Kamera_D/Foto_D.jpg";}
}
} //milik try
catch(Exception
err){out.println("Errs="+err);cek = true;}
}
try{stm.executeUpdate("delete from auto_read"); }
catch(Exception err){out.println("Kesalahan="+err);cek = true;}
%>
</span></span>
</form></td>
</tr>
<tr>
<td>Lokasi Parkir </td>

```

```

        <td><input name="textfield" type="text" size="33" value ="<%= posisi %>"
    /></td>
    </tr>
</table>
<p><span class="meta"><span class="sidebar style1">
    </span></span><span class="meta"><span class="sidebar style1">
</span></span> </p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<div class='single_post_photo'> <a href="" class='img' rel="bookmark"
title=""></a> </div>
<p class="small">&nbsp;</p>
</div>
</div>
</div> <!-- /header -->

```

=====Halaman Registrasi=====

```

<% @ include file="koneksi.jsp" %>
<%
boolean cek=false;
ResultSet hasil=null;
String kode_rfid="";
String notag="";

    try{
        hasil=stm.executeQuery("SELECT * FROM
auto_read");
    }
    catch(Exception err){out.println("Kesalahan="+err);cek =
true;}

```

```

if(!cek){
    try {
        while(hasil.next()){
            kode_rfid=hasil.getString("kode_rfid");
            //kode_rfid=kode_rfid.substring(2,10);
        }
    } //milik try
    catch(Exception
err){out.println("Kesalahan22="+err);cek = true;}
    }
    if (kode_rfid==""){notag="";}
    else {notag=kode_rfid;}
%>

</p>
<form name="form1" id="form1" method="post" action="saveRegistrasi.jsp">
<table width="381" border="1">
<tr>
<td width="80">Nomor Tag</td>
<td width="285"><input name="notag" type="text" id="notag" value
="<%= notag %>" /></td>
</tr>
<tr>
<td>Nomor Kendaraan </td>
<td><input name="nokendaraan" type="text" id="nokendaraan" /></td>
</tr>
<tr>
<td>Nama Pemilik </td>
<td><input name="namapemilik" type="text" id="namapemilik" size="50"
/></td>
</tr>

```

```

<tr>
  <td>No. Telp </td>
  <td><input name="notelp" type="text" id="notelp" size="30" /></td>
</tr>
<tr>
  <td>Alamat</td>
  <td><input name="alamat" type="text" id="alamat" size="75"
maxlength="255" /></td>
</tr>
</table>
<p>&nbsp; </p>
<p>
  <input type="submit" name="Submitted" value="Simpan" />
  <input type="reset" name="Reset" value="Reset" />

```

=====Halaman Input Kendaraan=====

```

<%
boolean cek=false;
ResultSet hasil=null;
String kode_rfid="";
String notag="";
    try{
        hasil=stm.executeQuery("SELECT * FROM
auto_read");
    }
    catch(Exception err){out.println("Kesalahan="+err);cek =
true;}
    if(!cek){
        try {

```

```

while(hasil.next()){

    kode_rfid=hasil.getString("kode_rfid");

}

} //milik try
catch(Exception
err){out.println("Kesalahan22="+err);cek = true;}

}

if (kode_rfid==""){notag=request.getParameter("notag");}
else {notag=kode_rfid;}
%>

<%
if (notag==""){notag=request.getParameter("notag");}
String nokendaraan=request.getParameter("nokendaraan");
String strketerangan="",strjam="",strtgl="",strnamafoto="",strposisi="";
String strnokendaraan="";
%>
<% cek=false;
if(notag==null){ notag=""; strnokendaraan="";}
else{
hasil=null;

    try{hasil=stm.executeQuery("SELECT * FROM tbl_rfid
where rfid="+notag+"");}
    catch(Exception e){out.println("Kesalahan="+e);cek =
true;}

    if(!cek){

        try {

            while(hasil.next()){

                strnokendaraan=hasil.getString("nokendaraan");

            }


```





```

<input name="notag" type="text" id="notag" value="<%= notag %>" />
    <label>
    <input type="submit" name="Submit" value="Cek" />
    </label>
</form>    </td>
</tr>
</table>
<%
String info="";
if(strketerangan==""){info="Masuk";
    %>
        <form name="form1" id="form1" method="post"
action="saveKendaraan.jsp">
<% }
else if(strketerangan=="elki"){info="Daftar";
    %>
        <form name="form1" id="form1" method="post"
action="Registrasi.jsp">
<% }else {info="Keluar";
    %>
        <form name="form1" id="form1" method="post"
action="outKendaraan.jsp">
<%
    }
%>
<table width="381" border="1">
    <tr>
        <td>Nomor_Kendaraan </td>
        <td><input name="nokendaraan" type="text" id="nokendaraan"
value="<%= strnokendaraan %>" readonly=1 /></td>
        <td>Lokasi Parkir </td>

```

```

<% if(strposisi==""){ %>
<td bgcolor=green><select name="select">
<% cek=false;
hasil=null;
        try{hasil=stm.executeQuery("SELECT *
FROM area_parkir where status='kosong' Order by id");}
        catch(Exception
e){out.println("Kesalahan="+e);cek = true;}
        if(!cek){
                try {
                        while(hasil.next()){
String s_posisi=hasil.getString("posisi");
out.println("<option
value=\"'+s_posisi+'\">"+s_posisi+"</option>");
                }
                } //milik try
        catch(Exception
e){out.println("Kesalahan =" +e);cek = true;}
        }
%>
</select></td>
<%
} else if(strposisi=="elki"){
%>
        <td bgcolor=yellow><input type =text name="select" readonly=1
>
<%
} else{
%>
<td bgcolor=red><input type =text name="select" readonly=1 value = "<%=
strposisi %>">

```

```

<%
}
%>
</tr>
</table>
<input type="hidden" name="notag" value="<%= notag %>" />
<input type="submit" name="Submitted" value="<%= info %>" />
<input type="reset" name="Reset" value="Reset" />
<label></label>
<br />
</form>
<p>&nbsp;</p>
<p>&nbsp;</p>
<div class='single_post_photo'> <a href="" class='img' rel="bookmark"
title="">" width="320" height="240"
class="photo_align_left_single" /></a>
<div class='excerpt'>
<p>&nbsp;</p>
</div>
</div>
<p class="small">&nbsp;</p>
</div>
</div>

```

=====Halaman MBC=====

```

<% @ include file="koneksi.jsp" %>
<%
String tglku = new SimpleDateFormat("yyyy/MM/dd").format(new Date());
String jamku = new SimpleDateFormat("hh:mm:ss").format(new Date());

```

```

        Date now =new Date();
SimpleDateFormat waktuku=new SimpleDateFormat("yyyy-MM-dd");
SimpleDateFormat waktuku2=new SimpleDateFormat("H:m:s");
        Calendar cal=Calendar.getInstance();
        cal.setTime(now);
boolean cek=false;
        ResultSet hasil;
        String sAlamatGambar;
        String data;
        String indexGambar;
        hasil=stm.executeQuery("select * from mtbl_parkir Order by id
desc");
        hasil.next();
try{ data=hasil.getString("namafotot"); }
catch(Exception e){ data="gambar0.jpg";}
int panjang=data.length();
data=data.substring(6,panjang-4);
int nomor=Integer.parseInt(data);
nomor=nomor+1;
indexGambar="gambar"+nomor+".jpg";
String strketerangan="Index kendaraan ke-"+nomor;
out.print("<h1>MotionBased_A/"+indexGambar+"</h1>");
sAlamatGambar="picture";
Date tanggal;
File f=f=new File("");
try{f=new File ("C:/Program Files/Apache Software Foundation/Tomcat
6.0/webapps/ROOT/ParkingArea/MotionBased_A/gambar.jpg");}
catch(Exception err){ out.println(err);}
if ((f.exists() ? "ADA":"TIDAK_ADA").equals("ADA")){

```

```

        f.renameTo(new File("C:/Program Files/Apache Software
Foundation/Tomcat
6.0/webapps/ROOT/ParkingArea/MotionBased_A/"+indexGambar));
        tanggal=new Date();
        tanggal.setTime(f.lastModified());
        out.println("==> Tanggal Pengambilan indexGambar
==>:"+tanggal);
        String alamat1=f.getAbsolutePath();
        sAlamatGambar=alamat1;
    try{
        stm.executeUpdate("insert into mtbl_parkir(tgl,jamt,namafotot,keterangan)
values (\"+waktuku.format(cal.getTime())
+\"'\,\'"+waktuku2.format(cal.getTime()) +\"'\,\'"+indexGambar
+\"'\,\'"+strketerangani+"'\");
        out.println("<a href=inputdata.jsp><font
color=pink>DATA SUKSES DI MASUKKAN....</a></font><hr>");
    }
    catch(Exception err){out.println("err:"+err);cek = true;
        out.println("<a href=inputdata.jsp><font
color=pink>DATA GAGAL DI MASUKKAN....</a></font><hr>");
    }
}
else{out.println("<H1><font color=red>DATA BELUM
TERSEDIA....</font></H1><hr>"); response.sendRedirect("lihatMBC.jsp");}
cek=false;
hasil=null;
        try{hasil=stm.executeQuery("SELECT * FROM
mtbl_parkir Order by id Desc");}
        catch(Exception err){out.println("Kesalahan="+err);cek =
true;}
        if(!cek){

```

```

        try {
            int no=0;
            int x;
            out.println("<table border=2><tr><TD><center><font
color=red><b>No.</td><center><font color=red><b>Tanggal
Pengambilan</td><center><font color=red><b>Jam Pengambilan</td><font
color=red><b><font color=red><CENTER>Link
Gambar</red></b></td><center><font color=red><b>Hasil
Capture</td><td><center><font color=red><b>Info_Parkir</td><tr>");
                while(hasil.next()){
                    no=no+1;
                                //no=hasil.getInt("id");
                    String idx=String.valueOf(hasil.getInt("id"));
                    String strjam=hasil.getString("jamt");
                    String strtgl=hasil.getString("tgl");
                    String strnamafoto=hasil.getString("namafotot");
                    String strketerangan=hasil.getString("keterangan");
                    String path="MotionBased_A/"+strnamafoto;
                    out.println("<tr><td><font color=red><center>"+no+" </td><td><font
color=red><center>"+strtgl+"</td><td><font color=red><center>"+strjam+"
WIB </td><td><b><font color=red><center><font color=red><center><a
href=detailGambarM.jsp?id="+strnamafoto+">"+strnamafoto+"</a></center></re
d></b></td><td><img src='"+path+"' width=160 height=60 alt="+ path
+"><td>"+strketerangan+"<td><tr>\n");
                                }
                    out.println("</table>");
                }
                catch(Exception
err){out.println("Kesalahan22="+err);cek = true;}
        }
    %>

```

```

<% @ include file="koneksi.jsp" %>
<%
boolean cek=false;
ResultSet hasil=null;
        try{hasil=stm.executeQuery("SELECT * FROM
mtbl_parkir Order by id Desc");}
        catch(Exception err){out.println("Kesalahan="+err);cek =
true;}

        if(!cek){
                try {
                        int no=0;
                        int x;
                        out.println("<table border=2><tr><TD><center><font
color=red><b>No.</td><center><font color=red><b>Tanggal
Pengambilan</td><center><font color=red><b>Jam Pengambilan</td><font
color=red><b></td><center><font color=red><CENTER>Link
Gambar</td></td><center><font color=red><b>Info_Parkir</td><tr>");
                                while(hasil.next()){
                                        no=no+1;
                                        //no=hasil.getInt("id");
                                        String idx=String.valueOf(hasil.getInt("id"));
                                        String strjam=hasil.getString("jamt");
                                        String strtgl=hasil.getString("tgl");
                                        String strnamafoto=hasil.getString("namafoto");
                                        String strketerangan=hasil.getString("keterangan");
                                        String path="MotionBased_A/"+strnamafoto;
                                        out.println("<tr><td><font color=red><center>"+no+"
</td><td><font color=red><center>"+strtgl+"</td><td><font
color=red><center>"+strjam+" WIB </td><td><b><font
color=red><center><font color=red><center><a

```

```
href=detailGambarM.jsp?id="+strnamafoto+">" +strnamafoto+"</a></center></re
d></b></td><td><img src=\"'+path+'\" width=160 height=60 alt="+ path
+"><td>" +strketerangan+"<td></tr>\n");
```

```

    }
    out.println("</table>");
}
catch(Exception
err){out.println("Kesalahan22="+err);cek = true;}
%>
```

=====Halaman TBC=====

```
<% @ include file="koneksi.jsp" %>
<%
String tgliku = new SimpleDateFormat("yyyy/MM/dd").format(new Date());
String jamku = new SimpleDateFormat("hh:mm:ss").format(new Date());
Date now =new Date();
SimpleDateFormat waktuku=new SimpleDateFormat("yyyy-MM-dd");
SimpleDateFormat waktuku2=new SimpleDateFormat("H:m:s");
Calendar cal=Calendar.getInstance();
cal.setTime(now);
boolean cek=false;
    ResultSet hasil;
    String sAlamatGambar;
    String data;
    String indexGambar;
    File f;
//do {
    hasil=stm.executeQuery("select * from wtbl_parkir Order by id
desc");
    hasil.next();
try{data=hasil.getString("namafotot"); }
catch(Exception e){data="gambar0.jpg";}
}
```



```

int panjang=data.length();
data=data.substring(6,panjang-4);
int nomor=Integer.parseInt(data);
nomor=nomor+1;
indexGambar="gambar"+nomor+".jpg";
String strketerangan="Index kendaraan ke-"+nomor;
try{
f=new File ("C:/Program Files/Apache Software Foundation/Tomcat
6.0/webapps/ROOT/ParkingArea/TimeBased_A/"+indexGambar);
}
catch(Exception err){
f=new File ("C:/Program Files/Apache Software Foundation/Tomcat
6.0/webapps/ROOT/ParkingArea/TimeBased_A/gambar0.jpg");
//f=new File ("TimeBased_A/gambar0.jpg");
}
out.print(f.getName() +" ==> " +(f.exists() ? "save data...":"searching data..."));
out.print("<h1>TimeBased_A/"+indexGambar+"</h1>");
sAlamatGambar="picture";
Date tanggal;
if ((f.exists() ? "ADA":"TIDAK_ADA").equals("ADA")){
    tanggal=new Date();
    tanggal.setTime(f.lastModified());
    out.println("==> Tanggal Pengambilan indexGambar
==>:"+tanggal);
    String alamat1=f.getAbsolutePath();
    sAlamatGambar=alamat1;
}
try{
stm.executeUpdate("insert into wtbl_parkir(tgl,jamt,namafotot,keterangan)
values ('"+waktuku.format(cal.getTime())
+"'\,'"+waktuku2.format(cal.getTime()) +'\'','"+indexGambar
+"'\,'"+strketerangan+"')");
}

```

```

out.println("<a href=inputdata.jsp><font color=pink>DATA SUKSES DI
MASUKKAN....</a></font><hr>");
    }
    catch(Exception err){out.println("err:"+err);cek = true;
    out.println("<a href=inputdata.jsp><font
color=pink>DATA GAGAL DI MASUKKAN....</a></font><hr>");
    }
}
else{out.println("<H1><font color=red>DATA BELUM
TERSEDIA....</font></H1><hr>"); response.sendRedirect("lihatTBC.jsp");}
//=====
cek=false;
hasil=null;
    try{hasil=stm.executeQuery("SELECT * FROM
wtbl_parkir Order by id Desc");}
    catch(Exception err){out.println("Kesalahan="+err);cek =
true;}
    if(!cek){
        try {
            int no=0;
            int x;
            out.println("<table border=2><tr><TD><center><font
color=red><b>No.</td><center><font color=red><b>Tanggal
Pengambilan</td><center><font color=red><b>Jam Pengambilan</td><font
color=red><b></font color=red><CENTER>Link
Gambar</red></b></td><center><font color=red><b>Hasil
Capture</td><td><center><font color=red><b>Info_Parkir</td><tr>");
                while(hasil.next()){
                    no=no+1;
                    //no=hasil.getInt("id");
                    String idx=String.valueOf(hasil.getInt("id"));

```

```

String strjam=hasil.getString("jamt");
String strtgl=hasil.getString("tgl");
String strnamafoto=hasil.getString("namafotot");
String strketerangan=hasil.getString("keterangan");
String path="TimeBased_A/"+strnamafoto;
out.println("<tr><td><font color=red><center>" +no+
</td><td><font color=red><center>" +strtgl+ "</td><td><font
color=red><center>" +strjam+ " WIB </td><td><b><font
color=red><center><font color=red><center><a
href=detailGambar.jsp?id="+strnamafoto+">" +strnamafoto+ "</a></center></red
></b></td><td><img src='\'+path+'\' width=160 height=60 alt="+ path
+"><td>" +strketerangan+"<td></tr>\n");
    }

    out.println("</table>");
    }
    catch(Exception
err){out.println("Kesalahan22="+err);cek = true;}
    }
    %>
=====Data Parking In dan Out=====
<%
boolean cek=false;
ResultSet hasil=null;
    try{hasil=stm.executeQuery("SELECT * FROM tbl_parkir
where keterangan='parking' Order by id Desc");}
    catch(Exception e){out.println("Kesalahan="+e);cek =
true;}
    if(!cek){
        try {
            int no=0;
            int x;

```

```

        out.println("<table border=2><tr><TD><center><font
color=red><b>No.<td><center><font color=red><b>Tanggal
Masuk<td><center><font color=red><b>Jam Masuk<td><center><font
color=red><b>Nomor Tag<td><center><font
color=red><b>No_Kendaraan<td><font color=red><b><font color=red>");
            while(hasil.next()){
                no=no+1;
                String idx=String.valueOf(hasil.getInt("id"));
                String strjam=hasil.getString("jamt");
                String strtgl=hasil.getString("tgl");
                String notag=hasil.getString("notag");
                String noken=hasil.getString("nokendaraan");
                String strnamafoto=hasil.getString("namafotot");
                String strketerangan=hasil.getString("keterangan");
                String path="MotionBased_A/"+strnamafoto;
                out.println("<tr><td><font color=red><center>"+no+"
</td><td><font color=red><center>"+strtgl+"</td><td><font
color=red><center>"+strjam+" WIB </td><td><font
color=red><center>"+notag+" </td><td><font
color=red><center>"+noken+"</td><td><b><font color=red><center><font
color=red><center><a href=detailkendaraan.jsp?id="+idx+">");
                    }
                out.println("</table>");
            }
        catch(Exception
e){out.println("Kesalahan22="+e);cek = true;}
    }
%>

```