

SISTEM INFORMASI PEMBAYARAN PARKIR BERBASIS WEB DENGAN MENGGUNAKAN MODEL WATERFALL

¹Achmad Maezar Bayu Aji; ²Biktra Rudianto

Sistem Informasi¹ Teknik Informatika²

Sekolah Tinggi Manajemen dan Ilmu Komputer Nusa Mandiri ^{1,2}

www.nusamandiri.ac.id

achmad.azb@nusamandiri.ac.id¹ biktra.brd@nusamandiri.ac.id²



Abstract - Simultaneously with the increasing population and increasing vehicle ownership will change the growth and development of human activities in it, especially in areas that have a high percentage of service activities, trade, and commercial. Restrictions arise when the system that runs on the parking process is still not computerized in processing vehicle data in and out as well as calculating how much time parking is still using a calculator to determine the amount of money to be paid. Therefore we need a good system to facilitate parking officers in carrying out their duties. The research method used by the writer is quantitative descriptive research method. Because quantitative descriptive research methods analyze and provide a clear and detailed picture based on the facts found in the field through interviews, data observations, looking for various literature related to research so that conclusions can be drawn for the model used by using the waterfall model supported by UML diagrams and languages web programming. The purpose of this study can help in the process of recording parking transactions in and out of the faster and more integrated, also for faster calculations and minimize errors in the calculation and reporting. And the results of this study are parking payment systems that can be integrated with the program so that it is more effective in processing data and minimizing the occurrence of errors that occur before when the system process has not been computerized..

Key Words: Parking Payment, Waterfall Model, Information System

Abstrak—Seiring dengan bertambahnya jumlah penduduk dan meningkatnya kepemilikan kendaraan akan mempengaruhi pertumbuhan dan perkembangan kegiatan manusia didalamnya terutama pada kawasan yang memiliki prosentase yang tinggi atas kegiatan jasa, perdagangan, maupun komersil. Kendala muncul ketika sistem yang berjalan pada proses parkir masih belum terkomputerisasi dalam pengolahan data kendaraan masuk dan keluar serta perhitungan berapa lama waktu parkir masih menggunakan kalkulator untuk menentukan jumlah uang yang harus dibayar. Oleh karena itu diperlukan adanya sistem yang baik untuk mempermudah petugas parkir dalam melaksanakan tugasnya. Metode penelitian yang digunakan oleh penulis adalah metode penelitian deskriptif kuantitatif. Karena metode penelitian deskriptif kuantitatif menganalisa dan memberikan gambaran yang jelas dan terperinci berdasarkan kenyataan yang ditemukan di lapangan melalui hasil wawancara, observasi data, mencari berbagai literatur yang terkait penelitian sehingga bisa ditarik kesimpulan untuk model yang digunakan dengan menggunakan model waterfall yang didukung diagram UML dan bahasa pemrograman web. Tujuan dari penelitian ini dapat membantu dalam proses pencatatan transaksi parkir masuk dan keluar menjadi lebih cepat dan lebih terintegritas, juga untuk perhitungan lebih cepat dan meminimalisir kesalahan dalam perhitungan dan pembuatan laporan. Dan hasil dari penelitian ini adalah sistem pembayaran parkir yang dapat terintegrasi dengan program sehingga lebih efektif dalam pengolahan data dan meminimalisir terjadinya kesalahan yang terjadi sebelumnya saat proses sistem belum terkomputerisasi.

Kata Kunci: Pembayaran Parkir, Waterfall, Sistem Informasi

PENDAHULUAN

Seiring dengan bertambahnya jumlah penduduk dan meningkatnya kepemilikan kendaraan akan mempengaruhi pertumbuhan dan

perkembangan kegiatan manusia didalamnya terutama pada kawasan yang memiliki prosentase yang tinggi atas kegiatan jasa, perdagangan, maupun komersil. Kendala muncul ketika sistem yang berjalan pada proses parkir masih belum

terkomputerisasi dalam pengolahan data kendaraan masuk dan keluar serta perhitungan berapa lama waktu parkir masih menggunakan kalkulator untuk menentukan jumlah uang yang harus dibayar. Parkir adalah keadaan tidak bergerak suatu kendaraan yang bersifat sementara karena ditinggalkan oleh pengemudinya (S. N. S. Nawawi et al., 2015). Dari segi pengelola parkir, biaya yang dikeluarkan untuk melakukan pengontrolan biaya parkir menjadi lebih besar ketika dilakukan secara manual oleh petugas parkir (Ichwana & Syahputra, 2017). masih banyak masalah yang timbul, misalnya tidak tertibnya parkir, lamanya pencatatan data parkir dan biaya parkir kendaraan yang tidak pasti (Eko Wahyudianto, 2013), memerlukan pengawasan yang ekstra ketat karena dengan ribuan kendaraan bermotor bisa sangat rentan dengan tindak kriminal (Andy Prasetyo Utomo, 2013). Dari segi lahan, berdasarkan permasalahan lahan parkir yang luas dan tingginya hingga mencapai beberapa lantai akan menyulitkan pengguna mencari lahan yang masih kosong (Hafif Bustani Wahyudi, 2016).

Berdasarkan permasalahan tersebut, maka pada penelitian ini mengusulkan menggunakan program berbasis *web* dengan menggunakan model *waterfall* (I. Nawawi et al., 2019). *Web* atau *World Wide Web (WWW)* merupakan salah satu sumber daya *internet* yang berkembang pesat. Informasi *Web* didistribusikan dengan pendekatan *hyperlink* yang memungkinkan seseorang memperoleh informasi dengan melompat dari satu halaman ke halaman lain (Juliani et al., 2018). Dengan menggunakan program berbasis *web* diharapkan dapat membantu pengolahan data dalam pembayaran parkir menjadi lebih terintegrasi dan lebih cepat. Pemrograman berbasis *web* dalam penelitian menggunakan model *waterfall* dalam perancangan, dan pembuatan. Model *waterfall* adalah bagian dari Metode *SDLC* yang sering juga disebut sekuensial linier (*sequential linear*) atau alur hidup klasik. Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengodean, pengujian, dan tahap pendukung (*support*) (S. & Shalahudin, 2018).

Tujuan dari penelitian ini dapat membantu dalam proses pencatatan transaksi parkir masuk dan keluar menjadi lebih cepat dan lebih terintegrasi, juga untuk perhitungan lebih cepat dan meminimalisir kesalahan dalam perhitungan dan pembuatan laporan

Metode penelitian yang digunakan oleh penulis adalah metode penelitian deskriptif kuantitatif. Karena metode penelitian deskriptif kuantitatif menganalisa dan memberikan gambaran yang jelas dan terperinci berdasarkan kenyataan yang ditemukan di lapangan melalui hasil wawancara, observasi data, mencari berbagai literatur yang terkait penelitian sehingga bisa ditarik kesimpulan (Sugiyono, 2015).

Dalam proses penerapannya, model *waterfall* memiliki beberapa langkah atau tahapan utama sebagai berikut :

1. Analisa Kebutuhan *Software*

Analisa kebutuhan *software* merupakan suatu kegiatan yang dimulai dari proses awal didalam mempelajari sesuatu mengevaluasi suatu bentuk permasalahan yang ada. Dalam tahapan ini penulis melakukan analisis kebutuhan yang diperlukan oleh perangkat lunak yang terdiri analisis kebutuhan fungsional dan analisis kebutuhan non-fungsional.

2. Desain

Desain merupakan kegiatan dalam penggambaran perencanaan, dan perancangan atau pengaturan dari beberapa elemen yang terpisah didalam sistem menjadi kesatuan dan berfungsi dengan baik. Dalam tahapan ini, penulis melakukan desain yang terdiri dari struktur data, arsitektur perangkat lunak, representasi antar muka dan prosedur pengkodean.

3. *Code Generation*

Code generation merupakan upaya dalam mengimplementasikan desain menjadi perangkat lunak. Dalam tahapan ini, penulis melakukan pembuatan kode program dengan menggunakan bahasa pemrograman *Java Server Pages (JSP)*. *JSP* merupakan *script* atau program yang diletakan di *server*, yang dapat dijalankan memakai *browser* tertentu (Kuryanti, 2014) dan aplikasi pendukung yang diperlukan seperti *Netbeans*. *Netbeans* merupakan sebuah aplikasi *Integrated Development Enviroment (IDE)* yang berbasiskan *java* dari *sun Microsystem* yang berjalan diatas *swing* dan banyak digunakan sekarang sebagai editor untuk berbagai bahasa pemrograman (Nofriadi, 2015).

4. *Testing*

Testing merupakan upaya dalam menelusuri lebih lanjut dalam perangkat lunak yang telah dibuat untuk mendapatkan informasi mengenai kualitas perangkat lunak yang sedang diuji. Dalam tahapan ini, penulis dihasilkan sesuai yang diinginkan.

HASIL DAN PEMBAHASAN

A. Analisa Kebutuhan *Software*

BAHAN DAN METODE

Analisa kebutuhan *software* merupakan suatu kegiatan yang dimulai dari proses awal didalam mempelajari sesuatu mengevaluasi suatu bentuk permasalahan yang ada. Dalam tahapan ini penulis melakukan analisis kebutuhan yang diperlukan oleh perangkat lunak yang terdiri analisis kebutuhan fungsional dan analisis kebutuhan non-fungsional.

1. Analisa Kebutuhan Bagian Admin

Dapat Melakukan Login; Mengelola Data User; Mengelola Data Akun; Mengelola Kategori; Mengelola Data Member; Mengelola Jurnal Umum Mengelola Laporan Data Member; Mengelola Laporan Rekap Parkir ; Mengelola Laporan Parkir Mengelola Laporan Jurnal Umum

2. Analisa Kebutuhan Petugas Parkir

a. Dapat Melakukan Login

Mengelola Kategori; Mengelola Data Member; Mengelola Menu Tiket Parking; Mengelola Laporan Data Member; Mengelola Laporan Rekap Parkir

b. Mengelola Laporan Parkir

3. Analisa Kebutuhan Bagian Keuangan

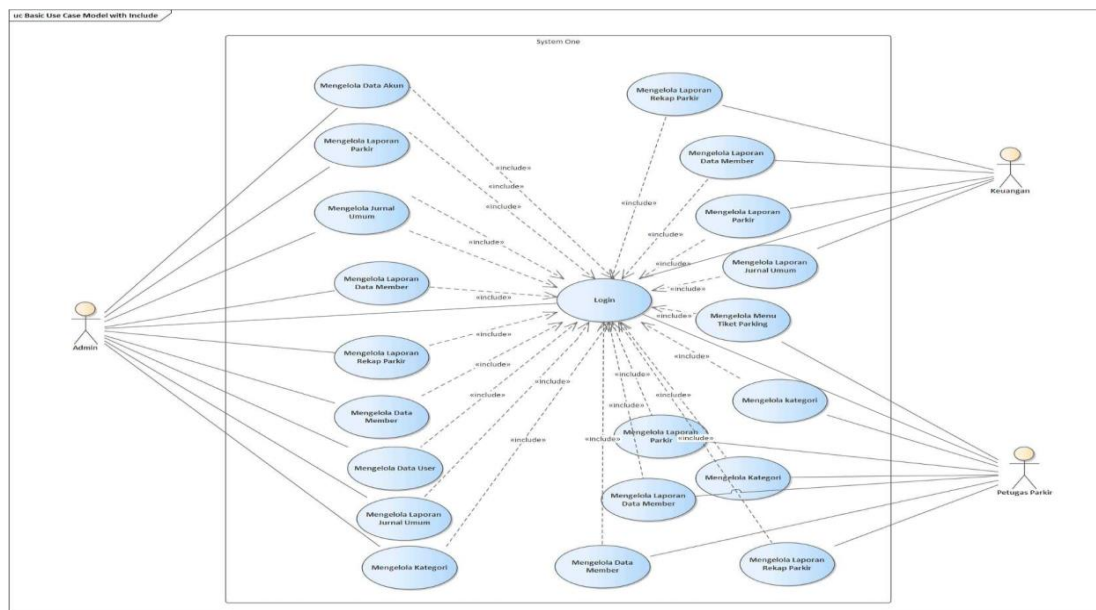
Dapat Melakukan Login; Mengelola Laporan Data Member; Mengelola Laporan Rekap Parkir; Mengelola Laporan Parkir; Mengelola Laporan Jurnal Umum

B. Desain

Desain merupakan kegiatan dalam penggambaran perencanaan, dan perancangan atau pengaturan dari beberapa elemen yang terpisah didalam sistem menjadi kesatuan dan berfungsi dengan baik.

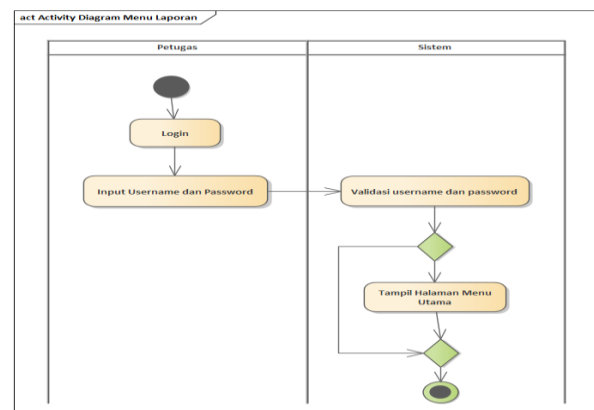
1. Desain Perangkat Lunak

Dalam tahapan ini desain perangkat lunak dibuat untuk pengembangan perangkat lunak dalam pembuatan program



Gambar 1. Use Case Diagram Admin

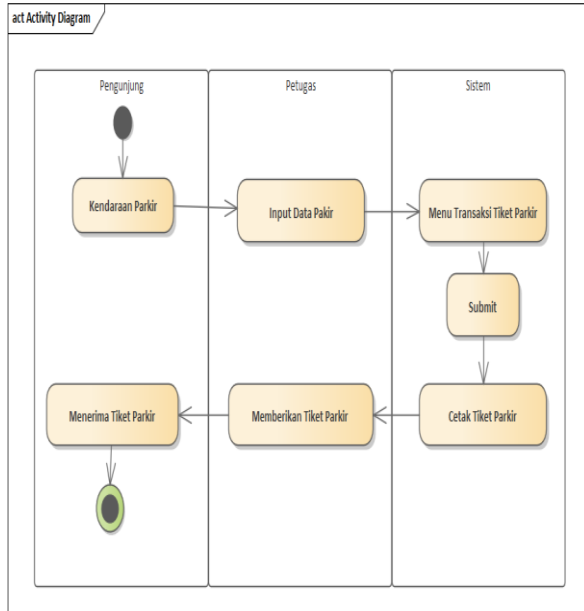
Dalam gambar 1 menggambarkan Use Case dari kegiatan admin, petugas parkir, dan bagian keuangan.



Sumber: (Achmad,Biktra.,2020)

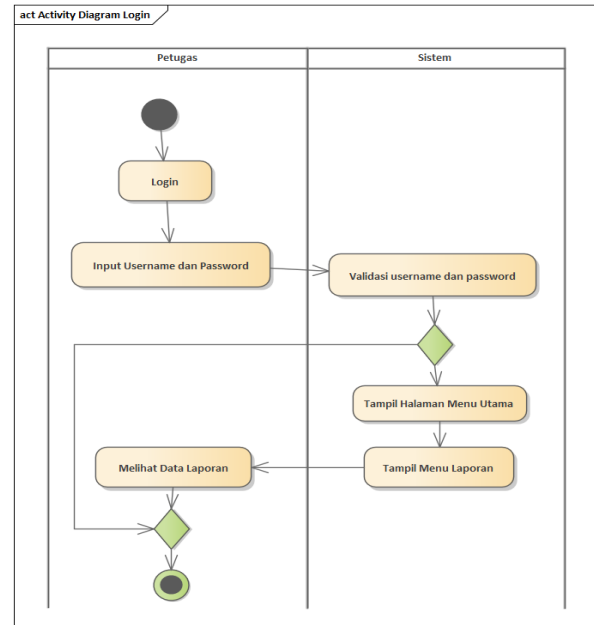
Gambar 2. Activity Diagram Login

Pada gambar 2, petugas parkir sebelum masuk ke dalam sistem harus melakukan *login* terlebih dahulu dengan memasukkan *username* dan *password*.



Gambar 3. Activity Diagram Tiket Parkir Masuk

Pada Gambar 3, pengunjung masuk parkir dan petugas mulai memasukkan data parkir dan pengunjung menerima tiket parkir yang diberikan petugas.

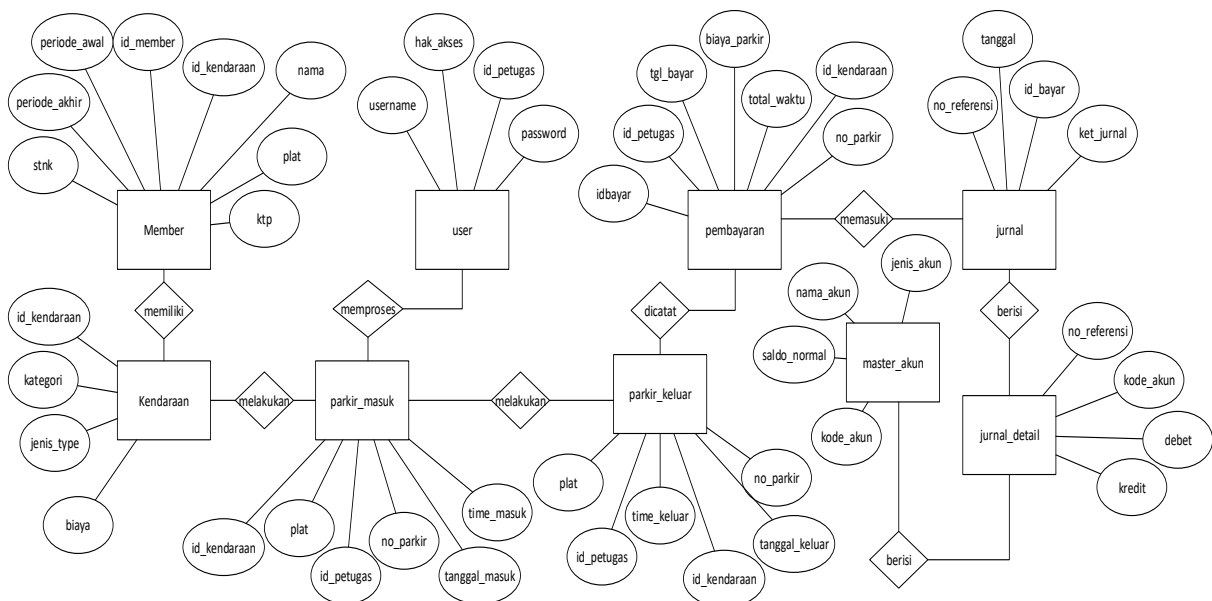


Gambar 4. Activity Diagram Menu Laporan

Pada gambar 4, Bagian keuangan harus login terlebih dahulu untuk mengakses menu laporan dengan memasukkan ID User dan Password dan kemudian akan tampil menu utama dan melihat menu laporan

2. Desain Database

Desain *Database* dibuat untuk merancang dan menentukan tabel-tabel yang ada saling berelasi dalam pembuatan program.



Sumber: (Achmad,Biktra,2020)

Gambar 5. Entity Relationship Diagram

Dalam gambar 5, *ERD* tersebut menggambarkan tabel-tabel yang berelasi yang dipergunakan untuk menyimpan dan mengolah data-data dalam *database*.

3. Desain User Interface

a. Halaman Menu Login



Gambar 6. Halaman Login Admin

Berdasarkan Gambar 6, Admin dalam mengelola dan mengolah data di dalam program terlebih dahulu harus login untuk masuk ke menu utama. Masukkan ID *User* dan *Password* yang terdaftar di dalam *database*.

b. Halaman Menu Utama



Gambar 7. Halaman Menu Utama

Pada Gambar 7 halaman menu utama ini, admin dapat mengelola dan mengolah data. Admin dapat membuka menu master, transaksi, dan menu laporan.

c. Halaman Tambah Member Parkir

Gambar 8. Halaman Tambah Member

Berdasarkan Gambar 8, Halaman ini berfungsi untuk menambah dan mencari data member ketika ingin menjadi member parkir. Untuk menjadi member parkir terdapat masa berlakunya yang nanti bisa diperpanjang secara berkala. Syarat untuk menjadi member parkir harus memasukan nomor KTP, STNK, dan Plat Nomor Kendaraan.

d. Halaman Tambah Kendaraan

ID Kendaraan	Kategori	Jenis Type	Plat	Aksi
0000001	Reguler	Mobil	4000	Hapus Edit
0000002	Reguler	Motor	2000	Hapus Edit
0000003	Tenant	Mobil	440000	Hapus Edit
0000004	Reguler	Motor	220000	Hapus Edit
0000005	Karyawan	Mobil	100000	Hapus Edit

Gambar 9. Halaman Tambah Kendaraan

Pada gambar 9, Setelah Menjadi member, maka langkah selanjutnya adalah memasukan data kendaraan. Dengan memilih jenis kendaraan dan memasukan biaya tersebut.

e. Halaman Tiket Parkir Masuk

No. Parkir	Tanggal	ID Petugas	ID Kendaraan	Plat	Aksi
89004	2019-08-14	11161080	3300451	B 1257	Cetak Tiket

Gambar 10. Halaman Parkir Masuk

Berdasarkan gambar 10, ketika ada kendaraan yang hendak parkir admin memasukkan data kendaraan tersebut dengan memilih jenis kendaraan dan memasukkan nomor plat tersebut dan klik tombol simpan dan data akan tersimpan ke dalam database. Dan apabila ada kendaraan yang sudah selesai parkir, maka admin menekan tombol data kendaraan parkir keluar dan akan diarahkan ke halaman parkir keluar seperti di gambar 10.

f. Halaman Data Kendaraan

Gambar 11. Halaman Data Kendaraan

Berdasarkan gambar 11, halaman ini berfungsi untuk menampilkan data kendaraan yang terdaftar di sistem parkir. Data kendaraan berisi data kendaraan yang tersimpan sebagai member kendaraan dalam sistem pembayaran parkir.

g. Halaman Data Pembayaran Parkir

No. Parkir	No. Plat	Waktu Masuk	Waktu Keluar	Lama Waktu (jam)	Biaya Per Jam	Biaya Parkir	Petugas	Kategori	Aksi
892017	B 5456 DOF	2019-08-09 11:37:22.0	2019-08-12 09:34:00.0	26	2000	52000	11161035	Reguler	Cetak
892016	B 4524 TWN	2019-08-09 11:37:07.0	2019-08-09 11:39:11.0	1	4000	4000	11161035	Reguler	Cetak
892015	B 3243 WRE	2019-08-09 07:40:51.0	2019-08-09 07:41:33.0	1	4000	4000	11161035	Reguler	Cetak
892014	B 4544 SWR	2019-08-09 07:40:27.0	2019-08-09 07:40:52.0	0	220000	0	11161035	Tenant	Cetak
892013	B 3243 WRE	2019-08-09 07:30:11.0	2019-08-09 07:39:28.0	1	2000	2000	11161035	Reguler	Cetak
892012	B 2503 SEG	2019-08-09 07:26:21.0	2019-08-09 07:27:47.0	1	4000	4000	11161035	Reguler	Cetak
892011	B 2503 TKN	2019-08-09 07:22:47.0	2019-08-09 07:23:20.0	1	2000	2000	11161035	Reguler	Cetak
892005	B 2503 TKN	2019-08-07 10:32:38.0	2019-08-07 14:17:57.0	0	220000	0	11161035	Tenant	Cetak

Gambar 12. Halaman Data Pembayaran Parkir

Berdasarkan gambar 12, untuk melihat data pembayaran parkir yang keluar bisa dilakukan di halaman data pembayaran parkir yang berada pada menu transaksi di halaman menu utama.

h. Halaman Laporan

Gambar 13. Halaman Rekap Laporan

Berdasarkan ntuk melihat laporan penerimaan data pembayaran, di halaman ini dapat dilakukan dengan memilih bulan dan tahun, setelah itu klik tombol tampil dan laporan akan ditampilkan sesuai bulan dan tahun yang dipilih

C. Code

Merupakan upaya dalam mengimplementasikan desain menjadi perangkat lunak. Dalam tahapan ini, penulis melakukan pembuatan kode program dengan menggunakan bahasa pemrograman Java Web dan aplikasi pendukung yang diperlukan.

```
<%@page import="java.util.Date"%>
<%@page import="java.text.SimpleDateFormat"%>
<%@page import="java.sql.*"%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<sql:query var="id_kendaraan" dataSource="${dataSource}">
SELECT id_kendaraan, kategori, jenis_type, biaya
FROM kendaraan
</sql:query>
<sql:query var="id_user" dataSource="${dataSource}">
SELECT id_user, nama_user, password, hak_akses
FROM user
</sql:query>
<!DOCTYPE html>
<%
//--koneksi database--
Connection koneksi = null;
Statement stmt = null;
ResultSet rs = null;
Class.forName("com.mysql.jdbc.Driver");
koneksi
=
DriverManager.getConnection("jdbc:mysql://localhost/db_park
ing", "root", "");
stmt = koneksi.createStatement();
%>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;
charset=UTF-8">
<title>JSP Page</title>
<%
SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd
HH:mm:ss");
String date = sdf.format(new Date());
%>
<meta http-equiv="Content-Type" content="text/html;
charset=UTF-8">
```

```

<table class="table table-striped table-bordered table-hover"
align="center" border="3" >
<td font="14"> TIKET PARKIR MASUK <td><a align="right"
href='halaman_utama.jsp?halaman=pembayaran' class="btn
btn-danger">DATA KENDARAAN PARKIR KELUAR</a></td>
</table>
</head>
<body>
<sql:query var="qryparkir_masuk" dataSource="${dataSource}"
>
Select * from parkir_masuk where no_parkir like
'%${param.cari}%' OR tanggal_masuk like '%${param.cari}%'
OR id_user like '%${param.cari}%' OR id_kendaraan like
'%${param.cari}%' OR plat like '%${param.cari}%'
</sql:query>
<style>
table {
border-collapse: collapse;
width: 680px;
}
th,td { text-align: left; padding: 8px; }
tr:nth-child(even){background-color: #f2f2f2}
th { background-color: #4CAF50; color: white; }
</style>
<form action="aksi_parkir_masuk" method="post">
<table border="0">
<input type="hidden" name="aksi" value="${param.aksi}"/>
<tr> <td>No. Parkir </td><td></td> <td>
<%
try {
ResultSet idparkir_masuk = null;
Statement perintah = koneksi.createStatement();
idparkir_masuk = perintah.executeQuery("Select
max(right(no_parkir,8)) as no FROM parkir_masuk_bc");
while (idparkir_masuk.next()) {
if (idparkir_masuk.first() == false) {
out.println("<input class="
readonly="" type='hidden'
name='no_parkir' value='ID0001' id='t1'/>");
} else {
idparkir_masuk.last();
int autoidparkir_masuk = idparkir_masuk.getInt(1) + 1;
String nomorid = String.valueOf(autoidparkir_masuk);
int noLong = nomorid.length();
</td>
</c:forEach>
</table>
</body>
<%@page import="java.sql.*, model.Akun, model.Jurnal" %>
<%
Jurnal Jurnal = new Jurnal();
Akun akun = new Akun();
/--Koneksi database--
Connection koneksi = null;
Statement perintah = null;
ResultSet rs = null;
ResultSet no = null;
Class.forName("com.mysql.jdbc.Driver");
koneksi
DriverManager.getConnection("jdbc:mysql://localhost/db_park
ing", "root", "");
perintah = koneksi.createStatement();
%>
<table class="table table-striped table-bordered table-hover"
align="center" border="3" >
<td font="14"> FORM JURNAL UMUM </td>
</table>
<form action="jurnalServ" method="post">
<table border="0">
<tr>
<td> No Referensi</td>
<td></td>
<td><% try {

```

```

rs = perintah.executeQuery("Select max(right(no_referensi,8))
as no FROM jurnal");
while (rs.next()) {if (rs.first() == false) {out.println("<input
type='hidden' name='no_referensi' value='JU000000001'/>");
out.println("<input type='text' value='JU000000001'
readonly='readonly'/>");
} else { rs.last(); int autoNo = rs.getInt(1) + 1; String nomor =
String.valueOf(autoNo); int noLong = nomor.length(); for (int a =
1; a < 9 - noLong; a++) { nomor = "0" + nomor; }
String nomer = "JU" + nomor;
out.println("<input type='hidden' name='no_referensi' value="
+ nomer + "'>"); out.println("<input type='text' value="
+ nomer + "' readonly='readonly'/>");
}} } catch (Exception e) {
out.println(e);
}
%>
</td></tr>
<tr><td> Tanggal Transaksi</td> <td></td>
<td><input type="date" name="tgl_trans" /></td></tr>
<tr> <td> No Bukti Transaksi</td> <td></td>
<td><input type="text" name="no_bukti_transaksi" /></td>
</tr>
<tr> <td> Keterangan</td> <td></td> <td><textarea
name="ket"></textarea></td></tr><tr>
<td colspan="3">
<table width="100%" border="0" cellpadding="0"
cellspacing="0"> <tr class="head"><td></td><td>Kode dan
Nama Akun</td> <td>Debet</td><td>Kredit</td></tr>
<tr class="isi"><td align="center"><input type="checkbox"
name="pilih[]" value="${param.pilih}"/></td>
<td><select name="no_akunkredit" id="akunkredit">
<option>Kode Akun | Nama Akun</option>
</table>
</form>

```

D. Testing

Merupakan upaya dalam menelusuri lebih lanjut dalam perangkat lunak yang telah dibuat untuk mendapatkan informasi mengenai kualitas perangkat lunak yang sedang diuji

Scan details

Scan information	
Start time	08/06/2020, 10:32:53
Start url	http://localhost/web-parkir/index.php
Host	http://localhost/web-parkir/index.php
Scan time	18 minutes, 47 seconds
Profile	Full Scan

Gambar 14. Hasil Pengujian Berdasarkan Waktu Pengujian

Berdasarkan gambar 14, pengujian web memerlukan waktu sekitar 18 menit dan 47 detik dengan skema pengujian *Full Scan*.

Threat level

Acunetix Threat Level 2

One or more medium-severity type vulnerabilities have been discovered by the scanner. You should investigate each of these vulnerabilities to ensure they will not escalate to more severe problems.

Alerts distribution

Total alerts found	18
High	0
Medium	9
Low	8
Informational	1

Gambar 15. Hasil Pengujian Berdasarkan Vulnerabilities

Berdasarkan gambar 15, untuk hasil pengujian dalam level kerentanan *web* yang diuji berada pada level 2 dan peneliti harus menyelidiki masing-masing kerentanan ini untuk memastikan mereka tidak akan meningkat ke masalah yang lebih parah.

KESIMPULAN

Simpulan dari penelitian ini adalah sistem yang diusulkan dapat digunakan untuk mempermudah sistem parkir yang telah ada sebelumnya dan mengurangi adanya kesalahan. Sistem informasi parkir yang diusulkan dapat membantu dalam pengolahan data transaksi perhitungan pemasukan parkir menjadi efisien. Dengan sistem informasi parkir yang diusulkan dapat mempermudah dalam penyimpanan data parkir. Di dalam pembuatan laporan parkir dapat dilakukan dengan mudah dan cepat, karena adanya sistem informasi parkir yang memiliki sistem penyimpanan dengan *database* dapat meminimalkan terjadinya kesalahan saat proses pembuatan laporan.

REFERENSI

- Andy Prasetyo Utomo. (2013). ANALISA DAN PERANCANGAN SISTEM INFORMASI PARKIR DI UNIVERSITAS MURIA KUDUS. *Jurnal SIMETRIS*, 1(1), 69–73. <https://doi.org/10.11113/jt.v56.60>
- Eko Wahyudianto. (2013). Perancangan Sistem Informasi Parkir Pada Dinas Perhubungan Komunikasi Dan Informatika Kabupaten Pacitan. *Seminar Riset Unggulan Nasional Informatika Dan Komputer FTI UNSA*, 19–26.
- Hafif Bustani Wahyudi. (2016). Sistem Pendeteksi Lahan Parkir Menggunakan Raspberry Pi, Sensor Ultrasonik dan Mikrokontroler. *J-INTECH*, 4(1), 58–65. <https://doi.org/10.1017/CBO9781107415324.004>
- Ichwana, D., & Syahputra, W. (2017). Sistem Pembayaran Parkir Menggunakan NEAR FIELD COMMUNICATION Berbasis ANDROID dan Teknologi INTERNET OF THINGS. *Jurnal Nasional Teknologi Dan Sistem Informasi*, 3(1), 153–164. <https://doi.org/10.25077/teknosi.v3i1.2017.153-164>
- Juliani, I. K., Salamuddin, M., & Dewi, Y. K. (2018). Perancangan Sistem Informasi E-Marketplace Bank Sampah Berbasis Web. *Seminar Nasional Teknologi Informasi Dan Multimedia 2018*, 19–24. <https://ojs.amikom.ac.id/index.php/semnasteknomedia/article/view/2017>
- Kuryanti, S. J. (2014). Perancangan Sistem Informasi Akademik Sekolah Berbasis Web Dengan Java. *Jurnal Khatulistiwa Informatika*, 2(1), 77–86.
- Nawawi, I., Nurajijah, N., & Abdilah, A. (2019). SISTEM MONITORING BARANG CETAK BERBASIS WEB MENGGUNAKAN MODEL WATERFALL. *INTI Nusa Mandiri*, 14(1), 77–84. <https://ejournal.nusamandiri.ac.id/index.php/inti/article/view/639>
- Nawawi, S. N. S., Sulistyorini, R., & Martono, Y. (2015). *Studi Optimalisasi Perparkiran dan Pedestrian di Fakultas Teknik Jurusan Sipil Universitas Lampung*. 3(1), 71–80, ISSN 2303-0011.
- Nofriadi. (2015). *Java Fundamental Netbeans 8.0.2 (Kesatu)*. Deepublish.
- S., R. A., & Shalahudin, M. (2018). *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek* (revisi). Informatika Bandung.
- Sugiyono. (2015). *Metode Penelitian Kuantitatif dan Kualitatif dan R&D* (Cetakan ke). Alfabeta.