

**PERANCANGAN PROGRAM  
SEWASEPEDA.COM MENGGUNAKAN  
APLIKASI NetBeans IDE 8.1**



**Disusun Oleh :**

- Rahma Indriani 11180015
- Chairul Fikri Azis 11180876
- Siti Fatimah 11181110
- Putri Nurwulan Sari 11180278
- Fikri Aulian Haikal 11180869

**SISTEM INFORMASI AKUNTANSI  
UNIVERSITAS BINA SARANA INFORMATIKA  
CENGKARENG  
2020**

## **KATA PENGANTAR**

Segala puji bagi Allah SWT yang telah memberikan kami kemudahan sehingga dapat menyelesaikan makalah Pemrograman Akuntansi II yang berjudul “Perancangan program sewasepeda.com menggunakan aplikasi Netbeans”. Tak lupa kami mengucapkan terima kasih kepada Bapak dosen “Deni Gunawan, M.Kom” yang telah membimbing kami.

Selain itu, kami sebagai manusia biasa menyadari bahwa masih banyak kekurangan dalam penulisan makalah ini, baik dari segi tata bahasa, susunan kalimat maupun isi. Oleh sebab itu kami selaku penyusun menerima segala kritik dan saran yang membangun dari pembaca.

Dengan karya ini kami berharap dapat membantu mahasiswa mengetahui tentang menggunakan aplikasi Netbeans. Dan makalah ini dibuat untuk melengkapi nilai tugas mata kuliah Pemrograman Akuntansi II pada semester 4.

Demikian yang bisa kami sampaikan, semoga makalah ini dapat membantu khazanah ilmu pengetahuan dan memberikan manfaat nyata untuk mahasiswa.

Jakarta, 26 Juni 2020

Penyusun

# DAFTAR ISI

<b>KATA PENGANTAR.....</b>	<b>i</b>
<b>DAFTAR ISI .....</b>	<b>ii</b>
<b>BAB I PENDAHULUAN .....</b>	<b>1</b>
1.1 Latar Belakang Masalah .....	1
1.2 Manfaat dan Tujuan .....	2
1.3 Metode Penelitian .....	3
1.3.1 Metode Pengumpulan Data .....	3
1.3.2 Metode Pengembangan Software .....	3
1.4 Ruang Lingkup .....	4
<b>BAB II LANDASAN TEORI .....</b>	<b>5</b>
2.1 Konsep Dasar .....	5
2.1.1 Pengertian Sistem.....	5
2.1.2 Karakteristik Sistem.....	5
2.1.3 Klasifikasi Sistem .....	7
2.1.4 Pengertian Internet .....	8
2.1.5 Pengertian Website .....	8
2.1.6 Bahasa Program (Script Program) .....	9
2.2 Peralatan Pendukung.....	10
2.2.1 Unified Modelling Language (UML) .....	10
2.2.2 Entity relationship Diagram (ERD) .....	11
2.2.3 Logical Record Structure (LRS) .....	12
2.2.4 Pengujian Web .....	12
2.2.5 Black-Box Testing .....	13
<b>BAB III PEMBAHASAN .....</b>	<b>14</b>
3.1 Analisis Kebutuhan Software .....	14
3.1.1 Analisis Kebutuhan.....	14
3.1.2 Use Case Diagram.....	14
3.1.3 Activity Diagram .....	15
3.2 Desain .....	19
3.2.1 Entity Relationship Diagram (ERD).....	19
3.2.2 Logical Record Structure (LRS) .....	20

3.2.3	Spesifik File .....	21
3.2.4	Sequence Diagram .....	26
3.2.5	Deployment Diagram .....	28
3.2.6	User Interface .....	28
3.3	Implementasi .....	29
3.3.1	Code Generation .....	29
3.3.2	Blackbox Testing .....	47
<b>BAB IV</b>	<b>PENUTUP .....</b>	<b>49</b>
4.1	Kesimpulan .....	49
4.2	Saran .....	49
<b>DAFTAR PUSTAKA</b>	<b>.....</b>	<b>50</b>

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang Masalah**

Sewasepeda.com adalah penyedia layanan penyewaan sepeda yg berada di kawasan jakarta. Sewasepeda.com menyediakan berbagai jenis sepeda yang dapat disewa oleh calon konsumen.

Terdapat beberapa masalah yang perlu diselesaikan di perusahaan ini. Masalah yang pertama yaitu penyajian informasi layanan yang ada di Sewasepeda.com yang kurang lengkap dan detail. Misalnya, sepeda yang disewakan tidak memiliki informasi yang lengkap dan terperinci. Akibatnya, banyak pertanyaan yang disampaikan ke Customer Service dan mereka harus menjelaskan lebih banyak detail layanan yang ada kepada calon pemesan.

Hal ini akan merugikan konsumen maupun Customer Service. Konsumen akan menghabiskan lebih banyak waktu bertanya, sedangkan Customer Service harus bekerja lebih banyak untuk menyampaikan informasi secara detail layanan yang tersedia kepada calon konsumen.

Masalah yang kedua yaitu pemesanan layanan yang kurang efektif dan efisien. Pemesanan sepeda bisa dilakukan dengan datang ke tempat rental atau melalui SMS atau Whatsapp. Jika mengirim SMS ataupun Whatsapp, pemesan harus menunggu jawaban dari Customer Service karena mereka tidak hanya melayani sedikit konsumen. Pemesan juga tidak bisa mengetahui total harga secara cepat dan tepat.

Dengan metode pemesanan yang ada seperti saat ini, Customer Service harus bekerja ganda karena meng-input-kan ulang pemesanan ke aplikasi pengolah data Microsoft Excel.

Masalah yang ketiga yaitu pengelolaan transaksi pemesanan dan pengembalian yang tidak efektif dan efisien. Selama ini, transaksi pemesanan dan pengembalian yang terjadi antar layanan tercampur menjadi satu antara transaksi penyewaan sepeda. Akibatnya, rekapitulasi data transaksi sulit dilakukan, rentan akan kesalahan-kesalahan dan lambat, karena harus dihitung satu per satu.

Masalah yang keempat yaitu pengelolaan bagi hasil dengan mitra yang tidak efektif dan efisien serta kurang transparan. Selain milik sendiri, sepeda yang ada di

Sewasepeda.com berasal dari kerjasama dengan masyarakat umum yang mempercayakan kendaraan mereka untuk dikelola pihak perusahaan. Sewasepeda.com ingin memiliki sistem yang mampu mengelola bagi hasil dengan mitra secara transparan dan terbuka sebagai bentuk tanggung jawab perusahaan kepada mitra. Pengelolaan bagi hasil dengan mitra masih dilakukan secara manual sehingga rentan akan kesalahan.

Maka dari itu, dibuat suatu sistem informasi pengelolaan pada perusahaan tersebut. Sistem informasi yang dibuat berbasis website agar mudah diakses oleh siapapun. Hal ini berguna untuk meningkatkan nilai tambah perusahaan dan meningkatkan daya saing Sewasepeda.com dengan usaha sejenis di kota Jakarta. Sehingga diharapkan usaha Sewasepeda.com lebih maju dibidang pengelolaan data dan penyajian informasi usahanya.

## **1.2 Manfaat dan Tujuan**

### **1.2.1 Manfaat**

Manfaat yang didapat dari penelitian di Sewasepeda.com ini yaitu:

- a. Bagi perusahaan Sewasepeda.com diharapkan dengan adanya sistem informasi penyewaan sepeda berbasis web di Sewasepeda.com ini maka aktivitas promosi, transaksi, dan pengelolaan data dapat dilakukan secara tepat dan akurat sehingga dapat membantu perusahaan berkembang ke arah yang lebih baik.
- b. Bagi penulis mempraktikkan ilmu selama belajar di bangku perkuliahan dengan cara membuat sistem informasi ini.
- c. Bagi mitra usaha meningkatnya kepercayaan mitra lama dan meyakinkan calon mitra untuk bergabung dengan perusahaan Sewasepeda.com. Dengan semakin meningkatnya kualitas layanan yang diberikan Sewasepeda.com untuk mitra, semakin meningkat nilai tambah perusahaan di mata mitra.
- d. Bagi konsumen memberikan kemudahan bagi konsumen untuk melihat informasi layanan yang disediakan oleh Sewasepeda.com Jakarta dan memberikan kemudahan konsumen dalam melakukan reservasi sepeda.

### **1.2.2 Tujuan**

Tujuan dari penelitian ini adalah:

- a. Membuat sistem yang mampu menyelesaikan masalah yang ada di perusahaan Sewasepeda.com dalam hal penyajian informasi layanan yang mereka miliki.
- b. Membuat sistem yang bisa meningkatkan efektivitas dan efisiensi dalam hal pemesanan sepeda di Sewasepeda.com.
- c. Membuat sistem yang mampu meningkatkan efektivitas dan efisiensi dalam pengelolaan transaksi di perusahaan Sewasepeda.com.
- d. Membuat sistem yang mampu meningkatkan efektivitas, efisiensi dan transparansi dalam hal bagi hasil dengan mitra usaha di Sewasepeda.com.

### **1.3 Metode Penelitian**

#### **1.3.1 Metode Pengumpulan Data**

##### **a. Metode Observasi**

Metode observasi adalah metode pengumpulan data dengan melakukan pengamatan langsung terhadap obyek yang diteliti dengan instansi terkait untuk mengumpulkan data dan informasi yang berkaitan dengan permasalahan yang ada. Informasi yang berkaitan adalah proses jalannya sistem di instansi tersebut.

##### **b. Metode Study Literature**

Metode Study Literature adalah metode pengumpulan data dengan menggunakan buku-buku yang berhubungan dengan software engineering dan pemrograman serta dari jurnal-jurnal online internasional.

#### **1.3.2 Metode Pengembangan Software**

Metode pengembangan software yang digunakan dalam perancangan ini adalah NetBeans IDE 8.1 dan software MySQL

##### **a. NetBeans IDE 8.1**

NetBeans adalah suatu serambi pengembangan perangkat lunak yang ditulis dalam bahasa pemrograman Java. Serambi Pada NetBeans, pengembangan suatu aplikasi dapat dilakukan dimulai dari setelan perangkat lunak modular bernama *modules*.

Semula, aplikasi NetBeans IDE ini diperuntukkan bagi pengembangan dalam Java. Namun, aplikasi ini juga mendukung program-program pembuatan bahasa lain secara khusus seperti PHP, C/C++ dan HTML5.

**b. MySQL**

MySQL adalah sebuah database management system (manajemen basis data) menggunakan perintah dasar SQL (*Structured Query Language*) yang cukup terkenal.

Database management system (DBMS) MySQL multi pengguna dan multi alur ini sudah dipakai lebih dari 6 juta pengguna di seluruh dunia.

## **1.4 Ruang Lingkup**

Agar tidak menyimpang pada pokok permasalahan, maka penulis membatasi hanya pada sistem transaksi perusahaan Sewasepeda.com yang masih menggunakan sistem manual beralih menjadi sistem website.



## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Konsep Dasar**

Suatu sistem dapat terdiri dari sistem-sistem bagian (subsistem). Subsistem yang saling berinteraksi dan saling berhubungan membentuk suatu kesatuan sehingga tujuan atau sasaran sistem tersebut dapat tercapai. Interaksi dari subsistem-subsistem terjalin sedemikian rupa, sehingga dicapai suatu kesatuan yang terpadu atau terintegrasi.

##### **2.1.1 Pengertian Sistem**

Menurut Atmosudirdjo dalam (Sutabri,2016) menyatakan bahwa: “suatu sistem terdiri atas objek-objek, atau unsur-unsur, atau komponen-komponen yang berkaitan atau berhubungan satu sama lainya sedemikian rupa sehingga unsurunsur tersebut merupakan suatu kesatuan pemrosesan atau pengolahan yang tertentu”.

Menurut (Sutabri, 2016) menyatakan bahwa : Pendekatan yang menekankan pada prosedur mendefinisikan sistem sebagai suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu. Sedangkan pendekatan sistem yang lebih menekankan pada elemen atau komponen mendefinisikan sistem sebagai kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu.

##### **2.1.2 Karakteristik Sistem**

Menurut (Mustakini, 2017) model umum sebuah sistem terdiri dari input, proses dan output. Hal ini merupakan konsep sebuah sistem yang sangat sederhana mengingat sebuah sistem dapat mempunyai beberapa masukan dan pengeluaran. Adapun karakteristik yang di maksud adalah sebagai berikut:

###### **1. Komponen Sistem (Components)**

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang bekerja sama membentuk suatu kesatuan komponen-komponen sistem tersebut dapat berupa suatu bentuk subsistem. Setiap subsistem memiliki sifat-sifat dari sistem yang menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan. Suatu sistem dapat mempunyai sistem yang lebih besar, yang disebut dengan Supra sistem.

## 2. Batasan Sistem (Boundary)

Ruang lingkup sistem merupakan daerah yang membatasi antara sistem dengan sistem lain atau sistem dengan lingkungan luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang sebagai satu kesatuan yang tidak dapat dipisahkan-pisahkan.

## 3. Lingkungan Luar Sistem (Environment)

Bentuk apapun yang ada di luar ruang lingkup atau batasan sistem yang mempengaruhi operasi sistem tersebut disebut dengan lingkungan luar sistem. Lingkungan luar sistem ini dapat menguntungkan dan dapat juga merugikan sistem tersebut. Lingkungan luar yang menguntungkan merupakan energi bagi sistem tersebut, yang dengan demikian lingkungan luar tersebut harus selalu dijaga dan dipelihara. Sedangkan lingkungan luar yang merugikan harus dikendalikan, karena kalau tidak maka akan mengganggu kelangsungan hidup sistem tersebut.

## 4. Penghubung Sistem (Interface)

Sebagai media yang menghubungkan sistem dengan subsistem yang lain disebut dengan penghubung sistem atau interface. Penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem yang lain. Keluaran suatu subsistem akan menjadi masukan untuk subsistem yang lain dengan melewati penghubung. Dengan demikian terjadi suatu integrasi yang membentuk satu kesatuan.

## 5. Masukan Sistem (Input)

Energi yang di masukan ke dalam sistem disebut masukan sistem. Yang dapat berupa pemeliharaan (maintenance input) dan sinyal (signal input). Sebagai contoh, di dalam suatu unit sistem komputer, "program" adalah maintenance input yang digunakan untuk mengoperasikan komputer sementara "data" adalah signal input yang akan diolah menjadi informasi.

## 6. Keluaran Sistem (Output)

Hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna, keluaran ini merupakan masukan bagi subsistem yang lain. Seperti contoh sistem informasi, keluaran yang dihasilkan adalah informasi, yang mana informasi ini dapat digunakan sebagai masukan untuk pengambilan keputusan atau hal-hal lain yang merupakan input bagi subsistem lainnya.

#### 7. Pengolah Sistem (Process)

Suatu sistem dapat mempunyai suatu proses yang akan mengubah masukan menjadi keluaran. Sebagai contoh, sistem akuntansi. Sistem ini akan mengolah data transaksi menjadi laporan-laporan yang dibutuhkan oleh pihak manajemen.

#### 8. Sasaran Sistem (Objective) Suatu sistem memiliki tujuan dan sasaran yang pasti dan bersifat deterministic. Kalau suatu sistem tidak memiliki sasaran, maka operasi sistem tidak ada gunanya. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuan yang telah direncanakan.

### 2.1.3 Klasifikasi Sistem

Menurut (Sutabri, 2016) Sistem merupakan bentuk integrasi antara satu komponen dengan komponen lain karena sistem memiliki sasaran yang berbeda untuk setiap kasus yang terjadi di dalam sistem tersebut. Oleh karena itu sistem dapat diklasifikasikan dari beberapa sudut pandang sebagai berikut:

#### 1. Sistem abstrak dan sistem fisik

Sistem abstrak adalah sistem yang berupa pemikiran atau ide-ide yang tidak nampak secara fisik, misalnya sistem theologi. Yaitu suatu sistem yang berupa pemikiran tentang hubungan antara manusia dengan Tuhan, sedangkan sistem fisik merupakan sistem yang ada secara fisik, seperti sistem komputer, sistem produksi, sistem penjualan, sistem administrasi personalia, dan lain sebagainya.

#### 2. Sistem alamiah dan sistem buatan manusia

Sistem alamiah adalah sistem yang terjadi melalui proses alam, tidak dibuat oleh manusia, misalnya sistem perputaran bumi, terjadinya siang malam, pergantian musim. Sedangkan sistem buatan manusia merupakan sistem yang melibatkan hubungan manusia dengan mesin, yang disebut dengan human machine system. Sistem informasi berbasis komputer merupakan contohnya, karena menyangkut penggunaan komputer yang berinteraksi dengan manusia.

#### 3. Sistem deterministik dan sistem probabilistik

Sistem yang beroperasi dengan tingkah laku yang dapat diprediksi disebut sistem deterministik. Sistem komputer adalah contoh dari sistem yang tingkah lakunya dapat dipastikan berdasarkan program-program komputer yang di jalankan. Sedangkan sistem yang bersifat probabilistik adalah

sistem yang kondisi masa depannya tidak dapat diprediksi, karena mengandung unsur probabilitas.

4. Sistem terbuka dan sistem tertutup Sistem tertutup merupakan sistem yang tidak berhubungan dan tidak terpengaruh oleh lingkungan luarnya. Sistem ini bekerja secara otomatis tanpa ada campur tangan dari pihak luar. Sedangkan sistem terbuka adalah sistem yang berhubungan dan dipengaruhi oleh lingkungan luarnya, yang menerima masukan dan menghasilkan keluaran untuk subsistem lainnya.

#### **2.1.4 Pengertian Internet**

Menurut Rossa dan Shalahudin dalam (Supriyatna, 2015), Internet atau internetwork adalah sekumpulan jaringan berbeda yang saling berhubungan bersama sebagai salah satu kesatuan dengan menggunakan berbagai macam protocol, salah satunya adalah protocol TCP/IP (Transmission Control Protocol/Internet Protocol).

#### **2.1.5 Pengertian Website**

Menurut raharjo dalam (Supriyanta & Nisa, 2015), "World Wide Web (WWW), sering disingkat dengan web, adalah suatu layanan di dalam jaringan yang berupa ruang informasi". Sedangkan menurut Sibero dalam (Supriyanta & Nisa, 2015), "World Wide Web (W3) atau yang dikenal juga dengan istilah web adalah suatu sistem yang berkaitan dengan dokumen digunakan sebagai media untuk menampilkan teks, gambar, multimedia dan lainnya pada jaringan internet".

Selanjutnya mari kita bahas pengertian dan perbedaan web statis dan web dinamis menurut (Ferdianto, 2013), yaitu sebagai berikut:

##### **1. Web Statis**

Web statis adalah website yang mana pengguna tidak bisa mengubah konten dari web tersebut secara langsung menggunakan browser. Interaksi yang terjadi antara pengguna dan server hanyalah seputar pemrosesan link saja. Halaman-halaman web tersebut tidak memiliki database, data dan informasi yang ada pada web statis tidak berubah-ubah kecuali diubah sintaksnya. Dokumen web yang dikirim kepada client akan sama isinya dengan apa yang ada di webserver. Contoh dari web statis adalah web yang berisi profil perusahaan. Di sana hanya ada beberapa halaman saja dan kontennya hampir tidak pernah berubah karena konten langsung diletakkan dalam file HTML saja.

## 2. Web Dinamis

Dalam web dinamis, interaksi yang terjadi antara pengguna dan server sangat kompleks. Seseorang bisa mengubah konten dari halaman tertentu dengan menggunakan browser. Request (permintaan) dari pengguna dapat diproses oleh server yang kemudian ditampilkan dalam isi yang berbeda-beda menurut alur programnya. Halaman-halaman web tersebut memiliki database. Web dinamis, memiliki data dan informasi yang berbeda-beda tergantung input apa yang disampaikan client. Dokumen yang sampai di client akan berbeda dengan dokumen yang ada di webserver. Contoh dari web dinamis adalah portal berita dan jejaring sosial. Lihat saja web tersebut, isinya sering diperbaharui (diupdate) oleh pemilik atau penggunanya. Bahkan untuk jejaring sosial sangat sering diupdate setiap harinya.

### 2.1.6 Bahasa Program (Script Program)

Bahasa program merupakan bahasa yang digunakan untuk menterjemahkan setiap perintah pada saat website tersebut sedang dijalankan. Bahasa program yang penulis gunakan adalah HTML dan PHP.

#### 1. HTML

Menurut Raharjo dalam (Supriyanta & Nisa, 2015), HTML adalah singkatan dari HyperText Markup Language, yaitu bahasa (aturan) standar yang digunakan untuk menampilkan teks, gambar, video atau audio ke dalam halaman web. Bahasa HTML merupakan bahasa yang digunakan untuk membuat halaman website dengan menggunakan tag-tag yang telah dideklarasikan pada halaman notepad dan dapat saling berhubungan dengan dokumen HTML lainnya atau yang sering disebut dengan istilah link.

#### 2. PHP

Menurut (Winarno & Zaki, 2013) “PHP adalah sebuah bahasa pemrograman scripting untuk membuat halaman web yang dinamis. Maksud dari scripting adalah pembuatannya menggunakan editor teks biasa, seperti Notepad, Notepad++ dan lainnya”.

#### 3. CSS

Menurut (Hidayatullah & Kawistara, 2014) “dengan CSS kita bisa menyimpan format dan menggunakannya kapanpun dan dimanapun kita inginkan.” Dengan CSS kita bisa membuat efek-efek tertentu untuk konten web yang kita buat. Contohnya kita ingin bagian header pada web selalu diberi font Verdana, size 20

dan warna biru. Maka kita definisikan style tersebut untuk tag . CSS (Cascading Style Sheet) sudah didukung oleh hampir semua web browser membuat karena CSS telah distandarkan oleh World Wide Web Consortium (W3C). Jadi pilihan yang tepat untuk kita format halaman web kita agar terlihat lebih indah dan cantik dimanapun user membuka web kita.

## **2.2 Peralatan Pendukung**

### **2.2.1 Unified Modelling Language**

Menurut (Sukamto & Shalahuddin, 2016) “UML (Unified Modeling Language) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan requirement, membuat analisi dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek”

UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak. UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dari teks-teks pendukung.

Pada UML 2,3 terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori yaitu structure diagrams yaitu yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan. Terdiri dari class diagram, object diagram, component diagram, composite diagram, package diagram, deployment diagram. Behavior diagrams yaitu kumpulan diagram yang digunakan untuk 15 menggambarkan suatu struktur statis dari sistem yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem. Terdiri dari, use case diagram, activity diagram, state machine diagram. Interaction diagram yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu sistem. Terdiri dari sequence diagram, communication diagram, timing diagram, interaction overview diagram (Sukamto & Shalahuddin, 2016).

Berikut penjelasan mengenai diagram-diagram UML menurut (Sukamto & Shalahuddin, 2016):

#### **1. Use Case Diagram**

Use case atau diagram use case merupakan pemodelan untuk kelakuan (behavior) sistem informasi yang akan dibuat. Use case mendeskripsikan

sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, use case digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.

## 2. Activity Diagram

Diagram aktivitas atau activity diagram menggambarkan workflow (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktiitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

## 3. Deployment Diagram

Diagram deployment atau deployment diagram menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi. Diagram deployment juga dapat digunakan untuk memodelkan hal-hal berikut, yaitu sistem tambahan (embedded system) yang menggambarkan rancangan device, node, dan hardware, sistem client/server, sistem terdistribusi murni dan rekayasa ulang aplikasi.

## 4. Component Diagram

Diagram komponen atau component diagram dibuat untuk menunjukkan organisasi dan ketergantungan diantara kumpulan komponen dalam sebuah sistem. Diagram komponen fokus pada komponen sistem yang dibutuhkan dan ada di dalam sistem.

### 2.2.2 Entity Relationship Diagram

Entity Relationship Diagram (ERD) digunakan dalam perancangan database. Menurut Rosa dan Shalahuddin dalam (Mulyadi & Sinnun, 2018) Entity Relation Diagram (ERD) adalah bentuk awal dalam melakukan perancangan basis data relasional. ERD merupakan suatu model yang menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi. Pada dasarnya ada tiga simbol yang digunakan sebagai berikut:

1. Entitas Digambarkan dengan kotak persegi panjang dan digunakan untuk menunjukkan sekumpulan orang,tempat, objek atau konsep dan sebagainya yang menunjukkan dimana data dicatat atau disimpan.
2. Hubungan atau Relasi

Digambarkan dengan kotak berbentuk diamond atau belah ketupat dengan garis yang menghubungkan ke entitas yang terkait. Maka relationship diberi nama dengan kata kerja. Hubungan atau relasi menunjukkan abstraksi dari sekumpulan hubungan yang mengaitkan antara entitas yang berbeda.

### 3. Atribut

Digambarkan dengan bentuk elips. Atribut menunjukkan karakteristik dari tiap entitas atau sesuatu yang menjelaskan entitas atau hubungan. Sehingga atribut dikatakan elemen dari entitas dan relasi. Dari setiap atribut entitas terdapat satu atribut yang dijadikan sebagai kunci (key). Beberapa jenis kunci tersebut antara lain :Primary key, Candidate key, Composite key, Secondary key, Alternate key dan Foreign key.

### 4. Tingkat Hubungan (Cardinality)

Entity Relation Diagram (ERD) juga menunjukkan tingkat hubungan yang terjadi, dilihat dari segi kejadian atau banyak tidaknya hubungan antara entitas tersebut.

## 2.2.3 Logical Record Structure

Menurut (Hasugian, 2012 ) menjelaskan bahwa : LRS (Logical Record Structure) merupakan hasil proses transformasi dari ERD dimana primary key yang terdapat pada masing-masing relasi akan masuk pada entitas yang lebih kuat. Proses transformasi digambarkan dengan persegi empat dengan garis putus-putus. Pada LRS akan kelihatan setiap primary key yang akan menjadi foreign key pada entitas lain. LRS ini menjadi patokan untuk pembuatan Conceptual Data Model.

## 2.2.4 Pengujian Web

Pengujian web dapat dilakukan dengan dua macam metode pengujian perangkat lunak yaitu blackbox atau blackbox testing.

Black-box testing yaitu perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Pengujian dimaksudkan untuk mengetahui apakah fungsi-fungsi masukan dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan. Kasus uji yang dibuat untuk melakukan pengujian kotak hitam harus dibuat dengan kasus benar dan kasus salah (Sukanto dan Shalahuddin, 2016 ), 18 misalkan untuk kasus proses login maka kasus uji coba yang dilakukan adalah:



1. Jika user memasukkan nama pemakai (username) dan kata sandi (password) yang benar.
2. Jika user memasukkan nama pemakai (username) dan kata sandi (password) yang salah, misalkan nama pemakai benar tapi kata sandi salah, atau sebaliknya atau keduanya salah.

### **2.2.5 Black-Box Testing**

Menurut (Sukanto dan Shalahuddin, 2016) mengatakan bahwa: “Black-Box testing (pengujian kotak hitam) yaitu menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Pengujian dimaksudkan untuk mengetahui apakah fungsi-fungsi, masukan, dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan”.

Pengujian kotak hitam dilakukan dengan membuat kasus uji yang bersifat mencoba semua fungsi dengan memakai perangkat lunak apakah sesuai dengan spesifikasi yang dibutuhkan. Kasus uji yang dibuat untuk melakukan pengujian kotak hitam harus dibuat dengan kasus benar dan kasus salah, misalkan untuk kasus login maka kasus uji yang dibuat adalah :

1. Jika user memasukkan nama pemakai (username) dan kata sandi (password) yang benar.
2. Jika user memasukkan nama pemakai (username) dan kata sandi (password) yang salah, misalnya nama pemakai benar tapi kata sandi salah, atau sebaliknya, atau keduanya salah

## **BAB III**

### **PEMBAHASAN**

#### **3.1 Analisis Kebutuhan Software**

##### **3.1.1 Analisis Kebutuhan**

Berdasarkan pada sistem yang berjalan saat ini di Sewasepada.com, kita dapat menganalisa kebutuhan-kebutuhan sistem yang akan dikembangkan, diantaranya : *administrator* dapat mengolah semua menu yang ada pada aplikasi, *administrator* dapat mengelola data dan laporan transaksi seluruh penyewaan sedangkan pelanggan atau *user* dapat mengakses menu produk dan menyewa setelah melakukan *login* terlebih dahulu. Berikut ini adalah analisa kebutuhan terhadap *administrator* dan *user* pada Sewasepada.com.

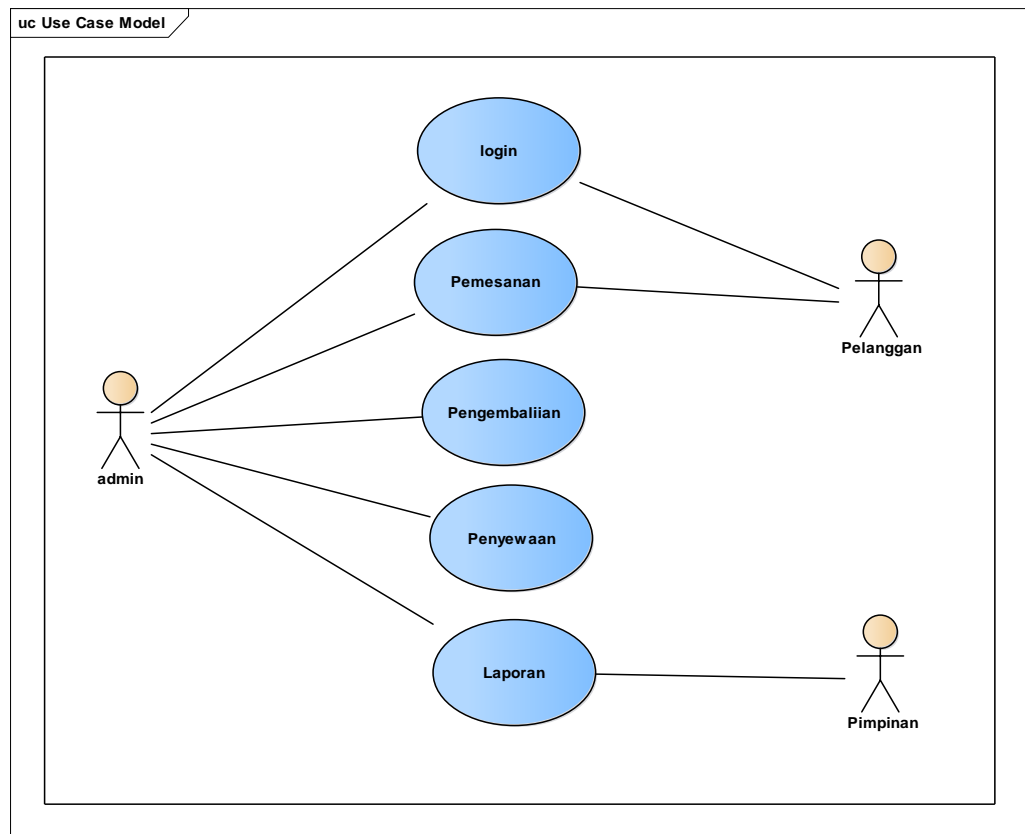
##### **Halaman Customer :**

- A1. Customer dapat mengelola login
- A2. Customer dapat mengelola pemesanan

##### **Halaman Administrator :**

- C1. Admin dapat mengelola data pada menu home.
- C2. Admin dapat mengelola data menu sewa.
- C3. Admin dapat mengelola data login.
- C3. Admin dapat mengelola data barang.
- C4. Admin dapat mengelola data pelanggan.
- C5. Admin dapat mengelola pesanan barang.
- C6. Admin dapat mengelola laporan.

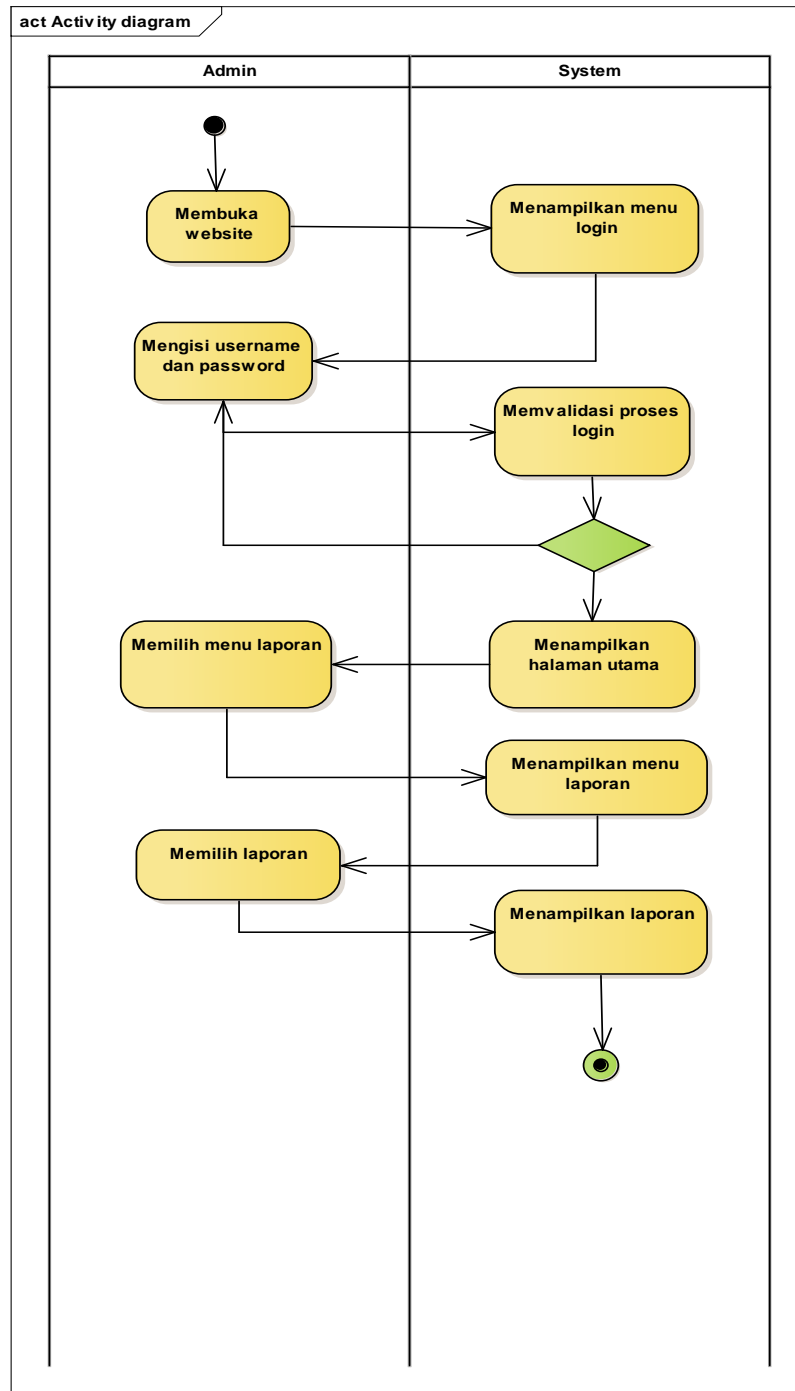
##### **3.1.2 Use Case Diagram**



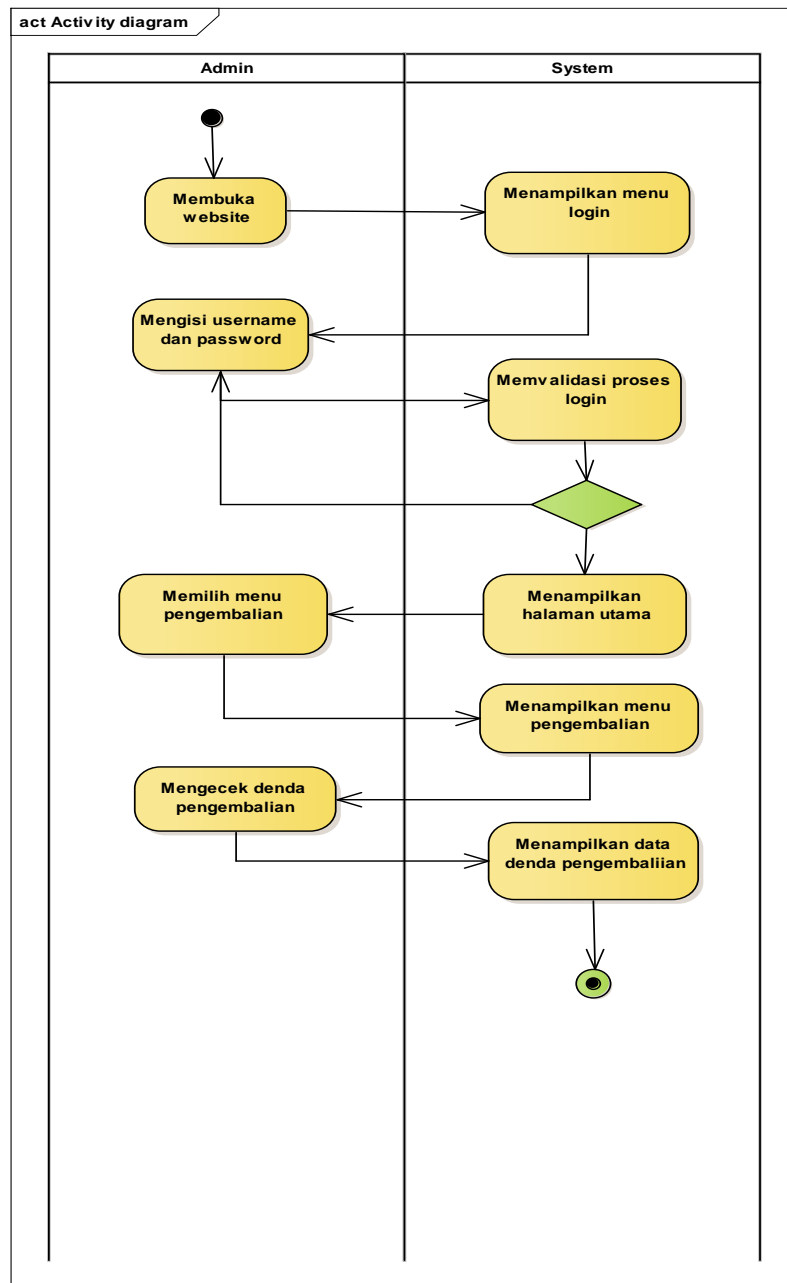
### 3.1.3 Activity Diagram

- Activity Diagram Penyewa

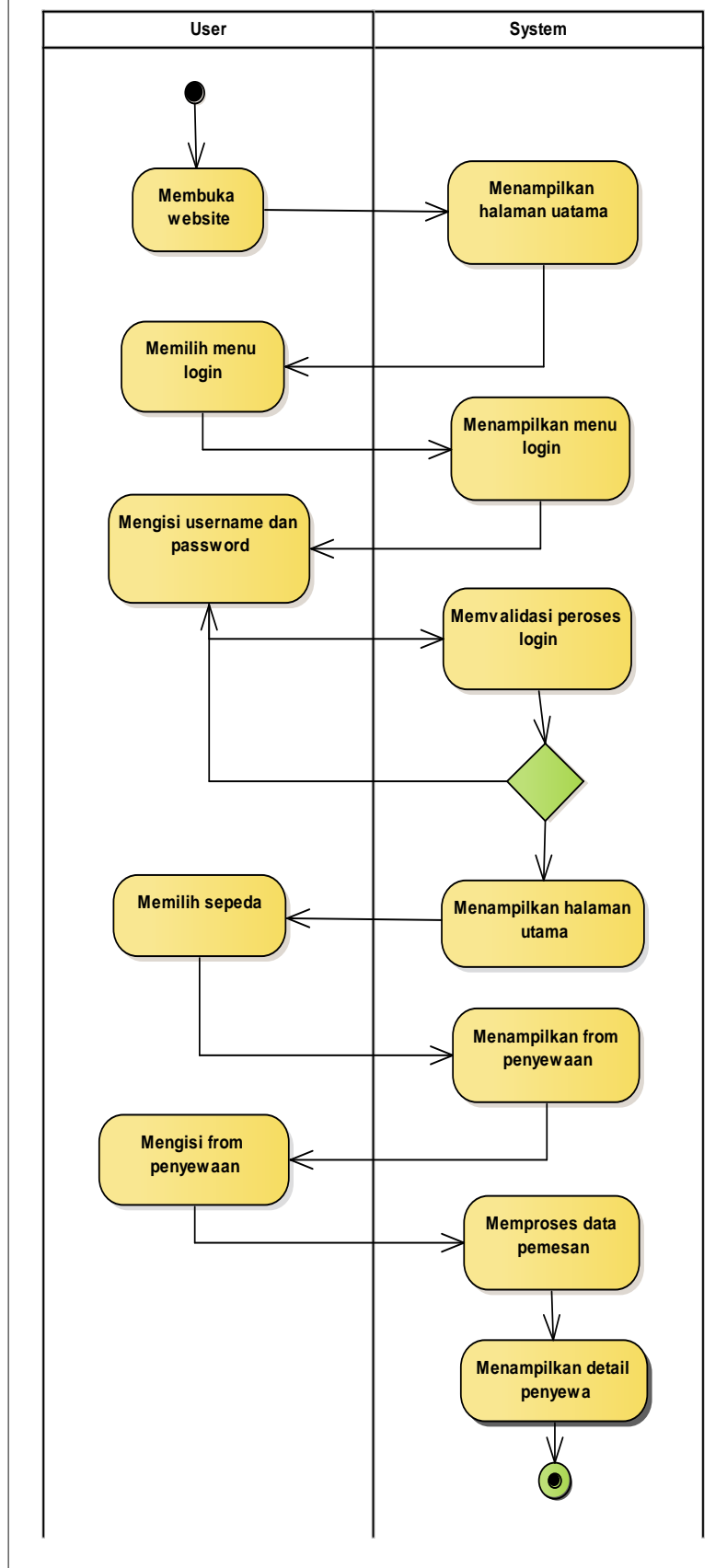




- **Activity Diagram Pengembalian**

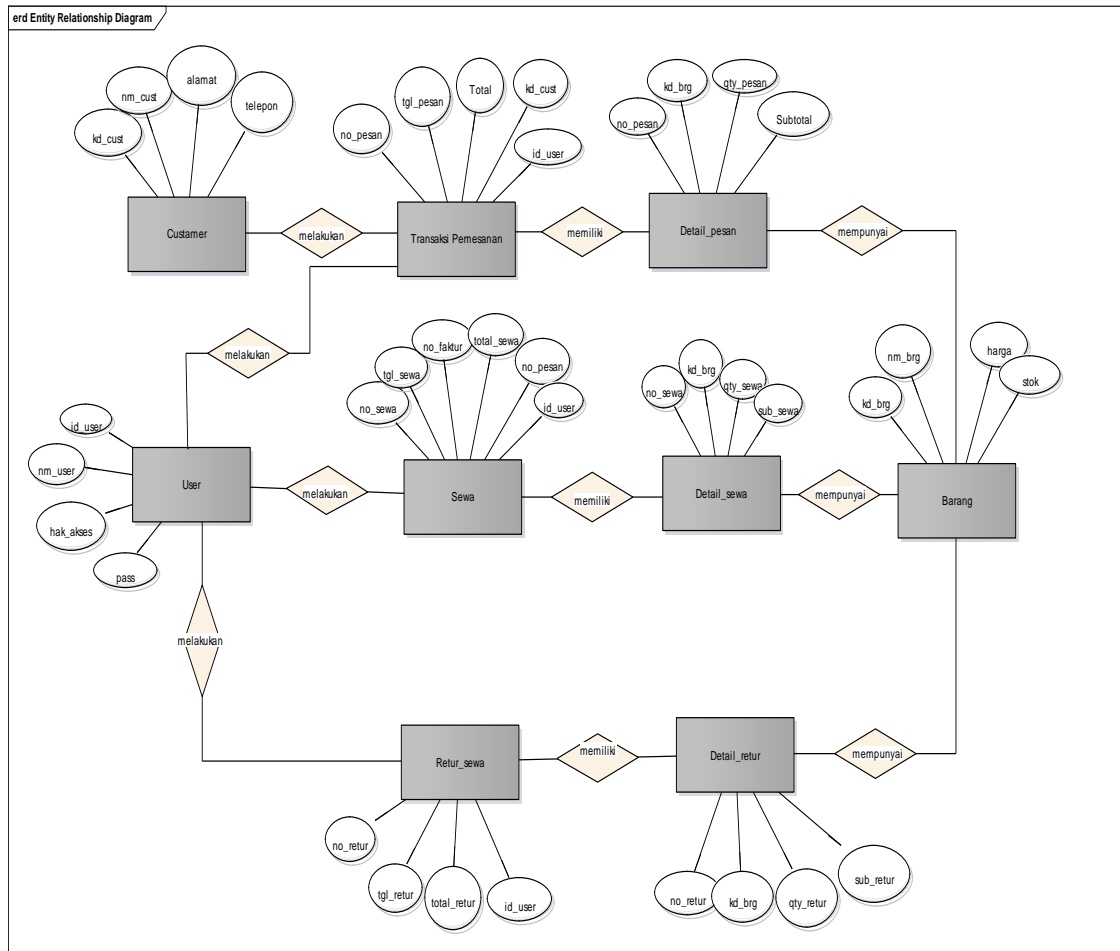


- **Activity Diagram Pemesanan**

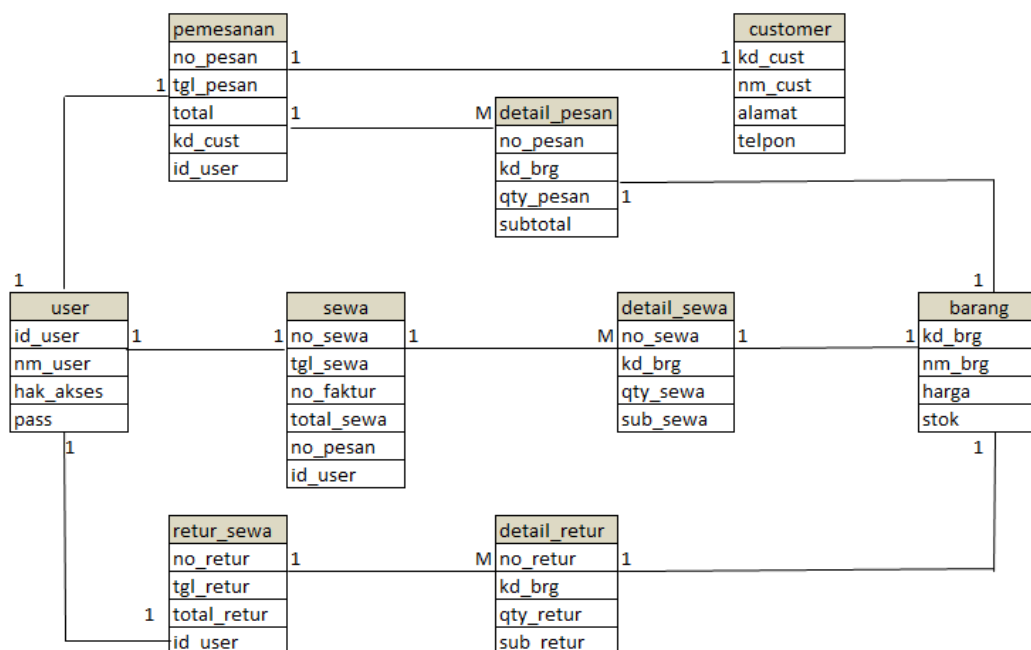


## 3.2 Desain

### 3.2.1 Entity Relationship Diagramma (ERD)



### 3.2.2 Logical Record Structure (LRS)





### 3.2.3 Spesifikasi File

Nama Database : sewasepeda

Jumlah Tabel : 14 Tabel

#### 1. Spesifikasi File user

Nama file : user  
Akronim : user  
Fungsi : menyimpan data pengguna  
Tipe File : file master  
Organisasi File : Index Sequential  
Akses File : Random  
Kunci Field : id\_user  
Software : MySQL

No	Elemen Data	Akronim	Tipe	Panjang	keterangan
1.	Id user	id_user	char	5	Primary key
2.	Nama user	nm_user	char	20	
3.	Hak akses	hak_akses	char	10	
4.	Pasword	pass	char	10	

#### 2. Spesifikasi File akun

Nama file : akun  
Akronim : akun  
Fungsi : menyimpan data akun  
Tipe File : file master  
Organisasi File : Index Sequential  
Akses File : Random  
Kunci Field : no\_akun  
Software : MySQL

No	Elemen Data	Akronim	Tipe	Panjang	keterangan
1.	Nomor akun	no_akun	char	5	Primary key
2.	Nama akun	nm_akun	char	20	

#### 3. Spesifikasi File sewa

Nama file : sewa  
Akronim : sewa  
Fungsi : menyimpan data sewa

Tipe File : file master  
 Organisasi File : Index Sequential  
 Akses File : Random  
 Kunci Field : no\_sewa  
 Software : MySQL

No	Elemen Data	Akronim	Tipe	Panjang	keterangan
1.	Nomor sewa	no_sewa	char	8	Primary key
2.	Tanggal sewa	tgl_sewa	date		
3.	Nomor faktur	no_faktur	char	10	
4.	Total sewa	total_sewa	int	11	
5.	Nomor pesan	no_pesan	char	8	

#### 4. Spesifikasi File detail sewa

Nama file : detail sewa  
 Akronim : detail\_sewa  
 Fungsi : menyimpan data detail sewa  
 Tipe File : -  
 Organisasi File : Index Sequential  
 Akses File : Random  
 Kunci Field : -  
 Software : MySQL

No	Elemen Data	Akronim	Tipe	Panjang	keterangan
1.	Nomor sewa	no_sewa	char	5	
2.	Kode barang	kd_sewa	char	25	
3.	Quantity sewa	qty_sewa	int	20	
4.	Subtotal sewa	sub_sewa	int	15	

#### 5. Spesifikasi File pemesanan

Nama file : pemesanan  
 Akronim : pemesanan  
 Fungsi : menyimpan data pesanan  
 Tipe File : file master  
 Organisasi File : Index Sequential  
 Akses File : Random  
 Kunci Field : no\_pesan  
 Software : MySQL

No	Elemen Data	Akronim	Tipe	Panjang	keterangan
1.	Nomor pemesanan	no_pesanan	char	8	Primary key
2.	Tanggal pesan	tgl_pesanan	date		
3.	Total	total	int	11	
4.	Kode customer	kd_cust	varchar	5	

## 6. Spesifikasi File detail pesan

Nama file : detail pesan  
 Akronim : detail\_pesanan  
 Fungsi : menyimpan data detail pesan  
 Tipe File : -  
 Organisasi File : Index Sequential  
 Akses File : Random  
 Kunci Field : -  
 Software : MySQL

No	Elemen Data	Akronim	Tipe	Panjang	keterangan
1.	Nomor pesan	no_pesanan	Char	8	
2.	Kode barang	kd_brg	Char	5	
3.	Quantity pesan	qty_pesanan	Int	11	
4.	Subtotal	subtotal	Int	11	

## 7. Spesifikasi File Barang

Nama file : barang  
 Akronim : barang  
 Fungsi : menyimpan data barang  
 Tipe File : file master  
 Organisasi File : Index Sequential  
 Akses File : Random  
 Kunci Field : kd\_brg  
 Software : MySQL

No	Elemen Data	Akronim	Tipe	Panjang	keterangan
1.	Kode barang	kd_brg	char	5	Primary key
2.	Nama barang	nm_brg	char	25	

3.	Harga	harga	int	11	
4.	Stok	stok	int	11	

### 8. Spesifikasi File customer

Nama file : customer  
 Akronim : customer  
 Fungsi : menyimpan data pelanggan  
 Tipe File : file master  
 Organisasi File : Index Sequential  
 Akses File : Random  
 Kunci Field : kd\_cust  
 Software : MySQL

No	Elemen Data	Akronim	Tipe	Panjang	keterangan
1.	Kode customer	kd_cust	char	5	Primary key
2.	Nama customer	nm_cust	char	25	
3.	Alamat	alamat	char	20	
4.	Telepon	telpon	char	15	

### 9. Spesifikasi File jurnal

Nama file : jurnal  
 Akronim : jurnal  
 Fungsi : menyimpan data jurnal  
 Tipe File : file master  
 Organisasi File : Index Sequential  
 Akses File : Random  
 Kunci Field : no\_jurnal  
 Software : MySQL

No	Elemen Data	Akronim	Tipe	Panjang	keterangan
1.	Nomor jurnal	no_jurnal	Char	5	Primary key
2.	Tanggal jurnal	tgl_jurnal	Date		

### 10. Spesifikasi File detail jurnal

Nama file : detail jurnal  
 Akronim : detail\_jurnal  
 Fungsi : menyimpan data detail jurnal  
 Tipe File : -

Organisasi File : Index Sequential  
 Akses File : Random  
 Kunci Field : -  
 Software : MySQL

No	Elemen Data	Akronim	Tipe	Panjang	keterangan
1.	Nomor jurnal	no_jurnal	char	8	
2.	Nomor akun	no_akun	char	5	
3.	Debet	debet	int	11	
4.	Kredit	akun	int	11	

#### 11. Spesifikasi File retur sewa

Nama file : retur sewa  
 Akronim : retur\_sewa  
 Fungsi : menyimpan data retur sewa  
 Tipe File : file transaksi  
 Organisasi File : Index Sequential  
 Akses File : Random  
 Kunci Field : no\_retur  
 Software : MySQL

No	Elemen Data	Akronim	Tipe	Panjang	keterangan
1.	Nomor retur	no_retur	char	8	Primary key
2.	Tanggal retur	tgl_retur	date		
3.	Total retur	total_retur	int	11	

#### 12. Spesifikasi File detail retur

Nama file : detail retur  
 Akronim : detail\_retur  
 Fungsi : menyimpan data detail retur  
 Tipe File : -  
 Organisasi File : Index Sequential  
 Akses File : Random  
 Kunci Field : -  
 Software : MySQL

No	Elemen Data	Akronim	Tipe	Panjang	keterangan
1.	Nomor retur	no_retur	char	8	
2.	Kode barang	kd_brg	char	5	

3.	Quantity retur	qty_retur	int	11	
4.	Subtotal retur	sub_retur	int	11	

### 13. Spesifikasi File sementara

Nama file : sementara  
 Akronim : sementara  
 Fungsi : menyimpan data sementara  
 Tipe File : file sementara  
 Organisasi File : Index Sequential  
 Akses File : Random  
 Kunci Field : -  
 Software : MySQL

No	Elemen Data	Akronim	Tipe	Panjang	keterangan
1.	Nomor	nomor	char	8	
2.	Kode	kode	char	5	
3.	Quantity	quantity	int	11	
4.	Subtotal	subtotal	int	11	

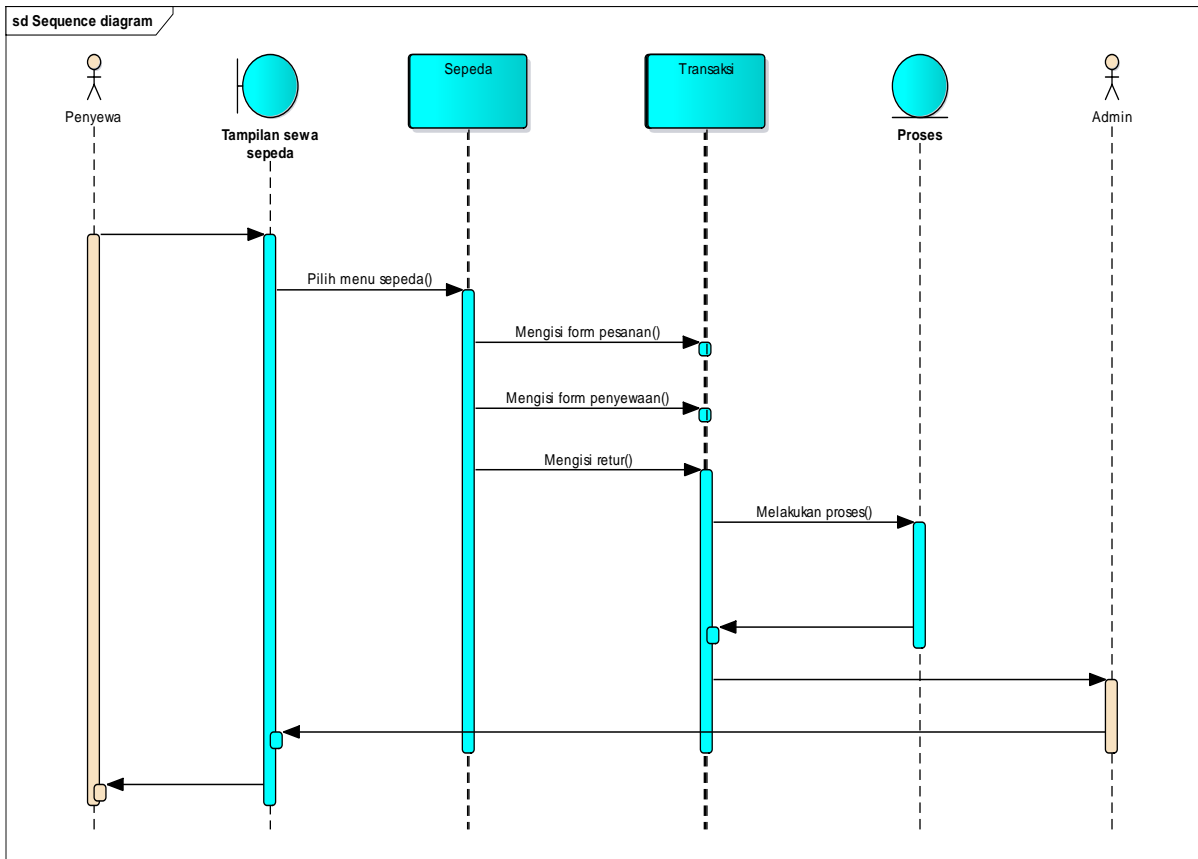
### 14. Spesifikasi File sementara retur

Nama file : sementara retur  
 Akronim : sementara\_retur  
 Fungsi : menyimpan data sementara retur  
 Tipe File : file sementara retur  
 Organisasi File : Index Sequential  
 Akses File : Random  
 Kunci Field : -  
 Software : MySQL

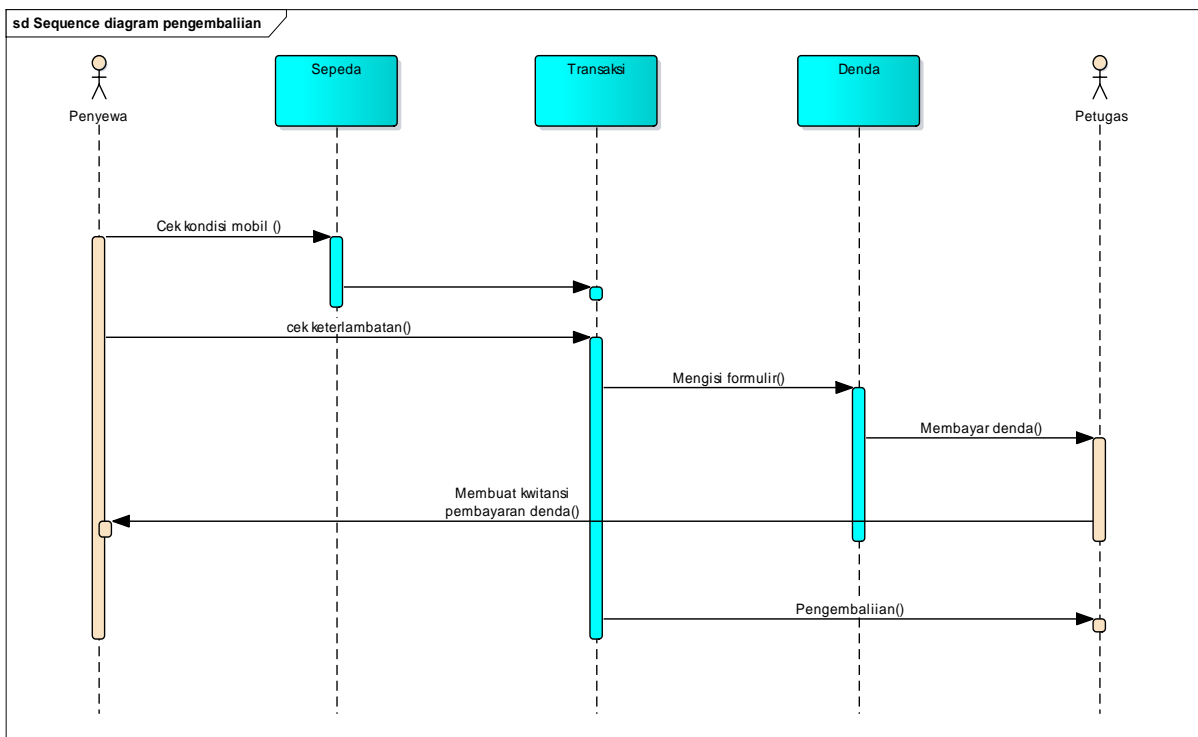
No	Elemen Data	Akronim	Tipe	Panjang	keterangan
1.	Nomor retur	no_retur	char	8	
2.	Kode barang	kd_brg	char	5	
3.	Quantity	quantity	int	11	
4.	Subtotal	subtotal	int	11	

## 3.2.4 Sequence Diagram

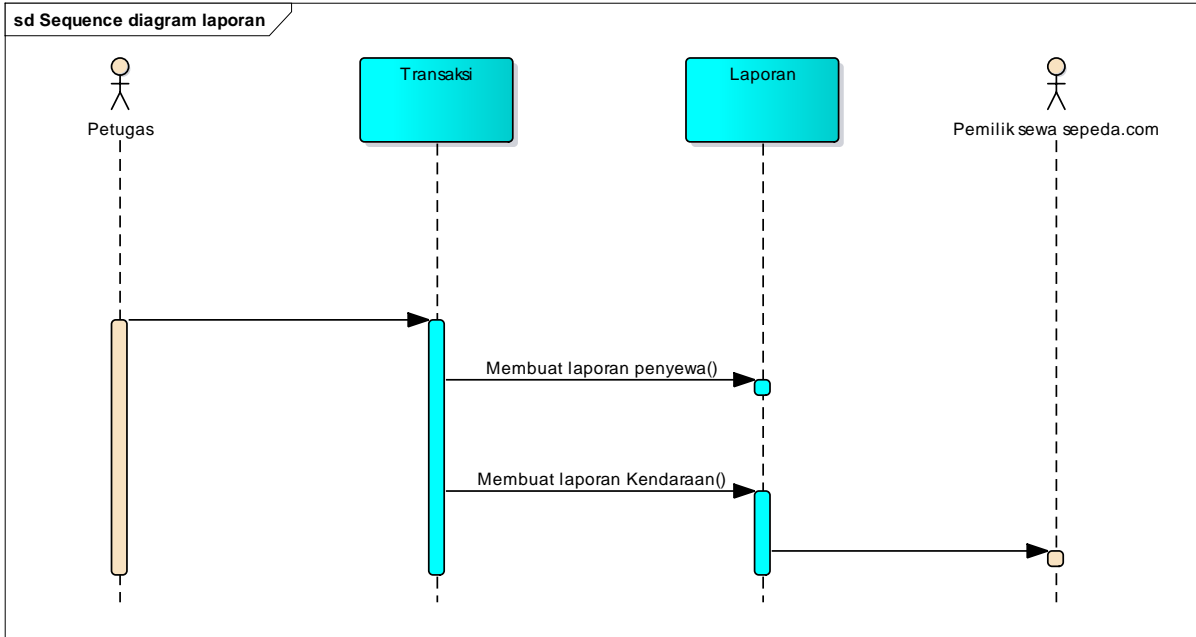
- Sequence Diagram Penyewaan



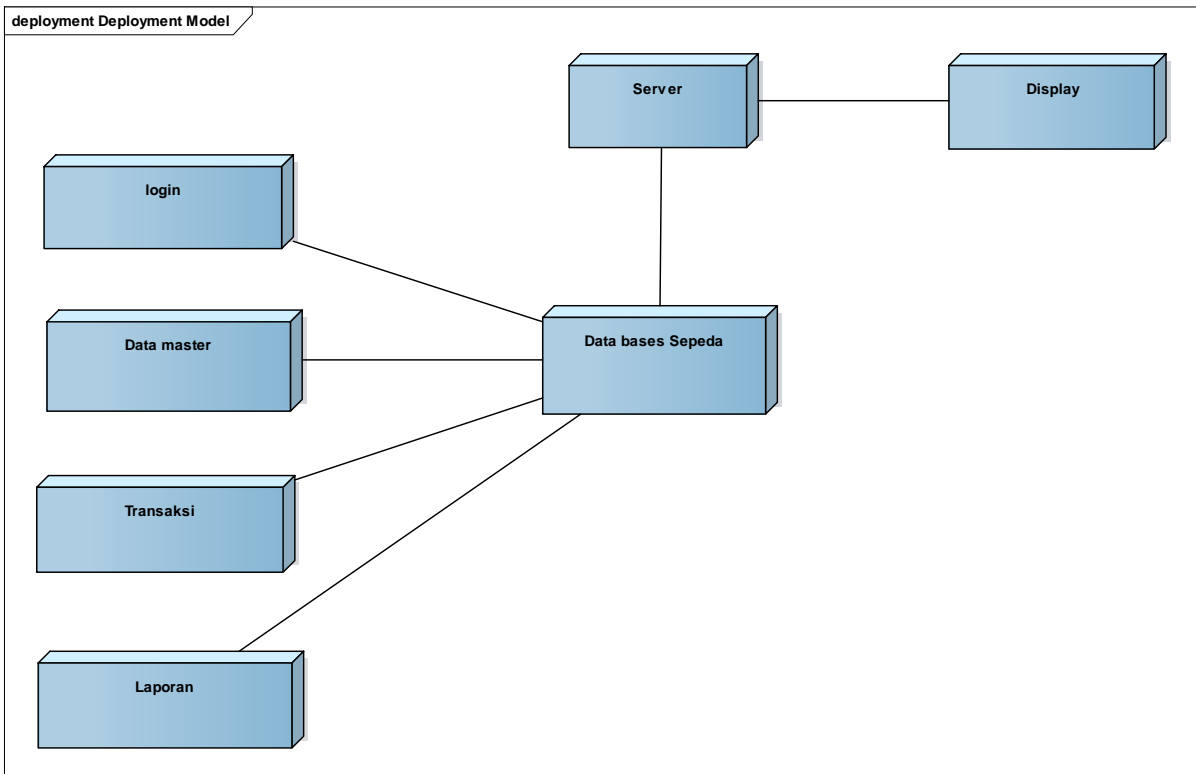
- Sequence Diagram Pengembalian



- Sequence Diagram Laporan

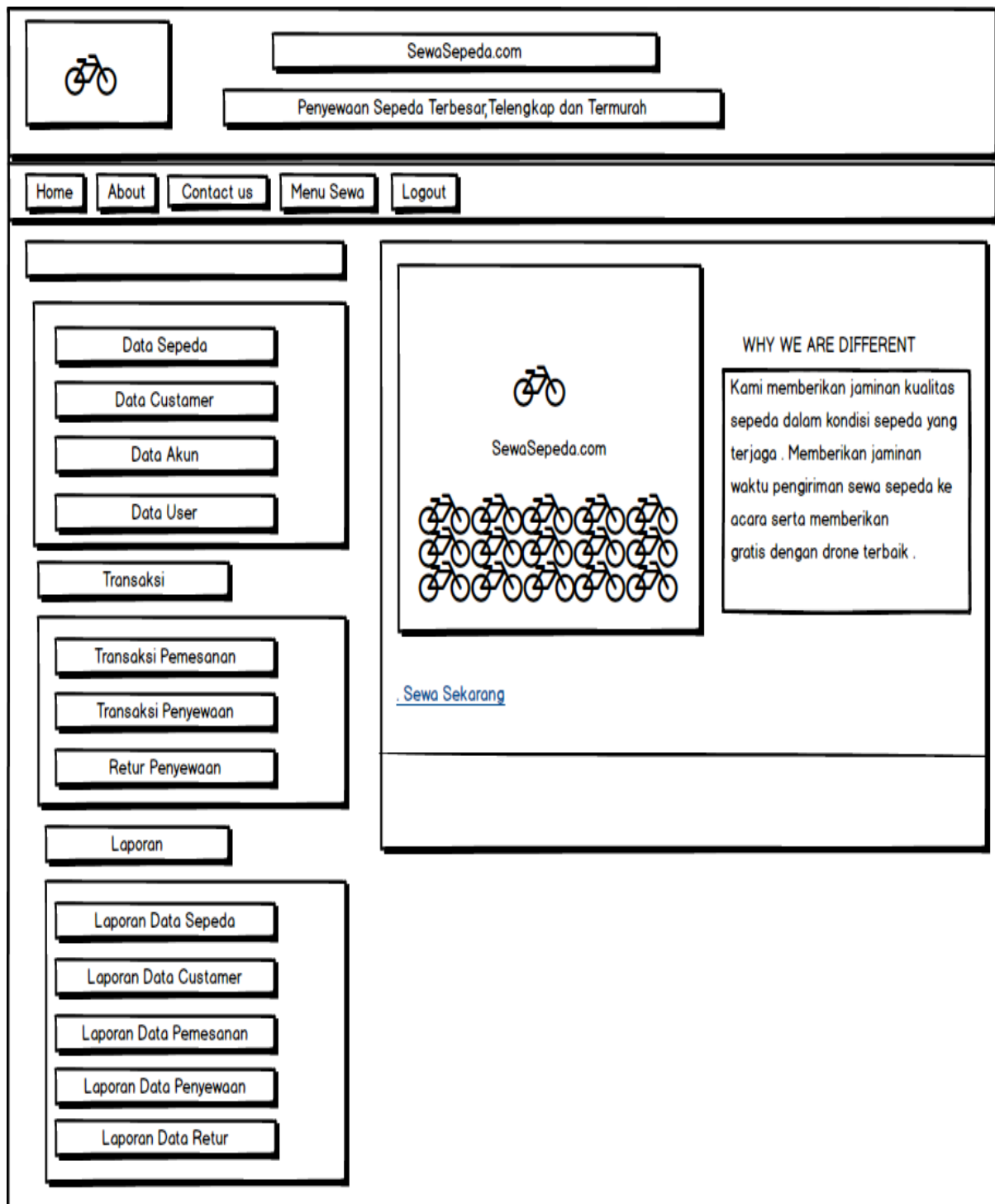


### 3.2.5 Deployment Diagram





### 3.2.6 User Interface



## 3.3 Implementasi

### 3.3.1 Code Generation

#### A. Source Code Tampilan Beranda

```
<%@page contentType="text/html" pageEncoding="UTF-8" %>
<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<%@taglib prefix="sql" uri="http://java.sun.com/jsp/jstl/sql"%>
<sql:setDataSource var="dataSource"
    driver="com.mysql.jdbc.Driver"
    url="jdbc:mysql://localhost:3306/sewasepeda"
    user="root"password=""/>

<!DOCTYPE html>
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>SewaSepeda.com</title>
    <link href="config/stylesheet.css" rel="stylesheet" type="text/css" />
    <script type="text/javascript" src="config/setting.js"></script>
    <link href="config/cssTable.css" rel="stylesheet" type="text/css" />
</head>
<body>
    <div id="wrap">
        <div id="header"><%-- awal header --%>
            
            <h1><a href="http://sewasepeda.com">SewaSepeda.com</a></h1>
            <h2>Penyewaan Sepeda Terbesar, Terlengkap dan Termurah</h2>
        </div><%-- akhir dari header --%>
        <div id="menu"><%-- awal dari menu atas --%>
            <ul>
                <li><a href="beranda.jsp?halaman=home">Home</a></li>
                <li><a href="beranda.jsp?halaman=about">About</a></li>
                <li><a href="beranda.jsp?halaman=kontak">Contact Us</a></li>
                <li><a href="beranda.jsp?halaman=menu">Menu Sewa</a></li>
                <li><a href="indexspd.jsp">Logout</a></li>
            </ul>
        </div><%-- akhir dari menu atas --%>
        <div id="content"><%-- awal dari isi content --%>
            <div class="left"><%-- awal dari content sebelah kiri web --%>
                <h2>Master Data</h2>
                <ul>
                    <li><a href="beranda.jsp?halaman=tampilbrg&aksi=SIMPAN">Data Sepeda</a></li>
                    <li><a href="beranda.jsp?halaman=tampilcus&aksi=SIMPAN">Data Customer</a></li>
                    <li><a href="beranda.jsp?halaman=tampilakun&aksi=SIMPAN">Data Akun</a></li>
                    <li><a href="beranda.jsp?halaman=tampiluser&aksi=SIMPAN">Data User</a></li>
                </ul>
                <h2>Transaksi</h2>
                <ul>
                    <li><a href="beranda.jsp?halaman=pesan">Transaksi Pemesanan</a></li>
                    <li><a href="beranda.jsp?halaman=sewa">Transaksi Penyewaan</a></li>
                    <li><a href="beranda.jsp?halaman=retursewa">Retur Penyewaan</a></li>
                </ul>
                <h2>Laporan</h2>
                <ul>
                    <li><a href="beranda.jsp?halaman=cetaklapbarang">Laporan Data Sepeda</a></li>
                    <li><a href="beranda.jsp?halaman=cetaklapcust">laporan Data Customer</a></li>
                    <li><a href="beranda.jsp?halaman=cetaklappemesanan">laporan Data Pemesanan</a></li>
                    <li><a href="beranda.jsp?halaman=cetaklapsewa">laporan Data Penyewaan</a></li>
                    <li><a href="beranda.jsp?halaman=cetaklapretur">laporan Data Retur Penyewaan</a></li>
                </ul>
                <div class="box" align="justify">
                    <div class="subtitle">Date and Time</div>
                    <marquee direction="left"><%=new java.util.Date()%>
                </div>
            </div>
        </div>
    </div>
</body>
</html>
```

```

    </marquee>
    <br />
</div>
<div class="box" align="justify">
    <div class="subtitle">About this website</div>
    <marquee behavior="scroll" direction="up" scrollamount="5">
        SISTEM INFORMASI AKUNTANSI UNIVERSITAS BSI APLIKASI SEWA SEPEDA
    </marquee>
</div>
<div class="box">
    <div class="subtitle">About this website</div>
    <marquee behavior="scroll" direction="up" scrollamount="5">
        MEMBER OF SEWASEPEDA.COM :
        <br>
        * RAHMA INDRIANI
        <br>
        * SITI FATIMAH
        <br>
        * CHAIRUL FIKRI AZIS
        <br>
        * PUTRI NURWULAN SARI
        <br>
        * FIKRI AULIAN HAIKAL
        <br>
    </marquee>
</div>
</div><!-- akhir dari content sebelah kiri web -->

```

```

<div class="right"><!-- awal dari content sebelah kanan web -->
    <table width="100%">
        <tr>
            <td valign="top" width="110%">
                <div class="text_area" align="justify">
                    <!-- isi dari content sebelah kanan -->
                    <c:choose>
                        <c:when test="{param.halaman=='home'}">
                            <@@include file="homespd.jsp" @>
                        </c:when>
                        <c:when test="{param.halaman=='about'}">
                            <@@include file="aboutsdpd.jsp" @>
                        </c:when>
                        <c:when test="{param.halaman=='kontak'}">
                            <@@include file="kontakus.jsp" @>
                        </c:when>
                        <c:when test="{param.halaman=='menu'}">
                            <@@include file="menuspdpd.jsp" @>
                        </c:when>
                        <c:when test="{param.halaman=='ps1'}">
                            <@@include file="paketsewa1.jsp" @>
                        </c:when>
                        <c:when test="{param.halaman=='ps2'}">
                            <@@include file="paketsewa2.jsp" @>
                        </c:when>
                        <c:when test="{param.halaman=='ps3'}">
                            <@@include file="paketsewa3.jsp" @>
                        </c:when>
                        <c:when test="{param.halaman=='tampilbrg'}">
                            <@@include file="barang.jsp" @>
                        </c:when>
                    </c:choose>
                </div>
            </td>
        </tr>
    </table>

```

```

<c:when test="${param.halaman=='tampilcus'}">
    <%@include file="customer.jsp" %>
</c:when>
<c:when test="${param.halaman=='tampilakun'}">
    <%@include file="akun.jsp" %>
</c:when>
<c:when test="${param.halaman=='tampiluser'}">
    <%@include file="user.jsp" %>
</c:when>
<c:when test="${param.halaman=='pesan'}">
    <%@include file="pemesanan.jsp" %>
</c:when>
<c:when test="${param.halaman=='retur sewa'}">
    <%@include file="retur.jsp" %>
</c:when>
<c:when test="${param.halaman=='sewa'}">
    <%@include file="penyewaan.jsp" %>
</c:when>
<c:when test="${param.halaman=='cetaklapbarang'}">
    <%@include file="CetakLaporanBarang.jsp" %>
</c:when>
<c:when test="${param.halaman=='cetaklapcust'}">
    <%@include file="CetakLaporanCustomer.jsp" %>
</c:when>
<c:when test="${param.halaman=='cetaklappemesanan'}">
    <%@include file="CetakLaporanPemesanan.jsp" %>
</c:when>

```

```

<c:when test="${param.halaman=='cetaklapsewa'}">
    <%@include file="CetakLaporanSewa.jsp" %>
</c:when>
<c:when test="${param.halaman=='cetaklapretur'}">
    <%@include file="CetakLaporanRetur.jsp" %>
</c:when>
<c:when test="${param.halaman=='logout'}">
    <%@include file="indexspd.jsp" %>
</c:when>
<c:otherwise>
    <%@include file="homespd.jsp" %>
</c:otherwise>
</c:choose>
</div>
</td>
</tr>
</table>
</div>
</div>
<div class="footer"><!-- awal dari content footer web -->
    Support by <a href="http://bsi.ac.id/">Junior Programmer
    Mahasiswa Program Studi Sistem Informasi Akuntansi</a>,<br>
    Thanks to Layout menu utama menggunakan CSS
</div><!-- akhir dari content footer web -->
<div style="clear: both;"> </div>
</div><!--akhir dari halaman web -->

</html>

```

## B. Source Code Tampilan Data Master Barang

```

<%@page import="control.koneksi"%>
<%@page import="java.sql.*"%>
<h1>Data Master Barang</h1>
<table border="0">
  <form action="BarangServlet" method="post" onsubmit="return validasi_inputBarang(this)">
    <input type="hidden" name="aksi" value="{param.aksi}"/>
    <tr>
      <td>Kode Barang</td> <td>: <input type="text" name="kode" value="{param.kd_brg}"/> </td>
    </tr>
    <tr>
      <td>Nama Barang</td> <td>: <input type="text" name="nama" value="{param.nm_brg}"/> </td>
    </tr>
    <tr>
      <td>Harga Barang</td> <td>: <input type="text" name="harga" onkeypress="return hanyaAngka(event)" value="{param.harga}"/> </td>
    </tr>
    <tr>
      <td>Stok Barang</td> <td>: <input type="text" name="stok" onkeypress="return hanyaAngka(event)" value="{param.stok}"/> </td>
    </tr>
    <tr>
      <td><input type="submit" value="Simpan" class="button"/></td> <td></td>
    </tr>
  </form>
  <td></td></tr></table><br>
  <form action="" method="post">
    Cari Data Barang : <input type="text" name="cari" value="{param.cari}"/>
    <input type="submit" value="Cari" class="button"/> <br>
  </form>
  <sql:query var="querycustomer" dataSource="{dataSource}">
    SELECT * from barang where kd_brg like '{param.cari}%' or
    nm_brg like '{param.cari}%'
  </sql:query>

  </sql:query>
  <br><table border="1" width="100%" cellpadding="0" cellspacing="0">
    <tr class="head">
      <td>Kode Barang</td><td>Nama Barang</td><td>Harga Barang</td><td>Stok Barang</td><td>Action</td>
    </tr>
    <c:forEach var="rowbrg" items="{querycustomer.rowsByIndex}">
      <tr class="isi">
        <td valign="top">{rowbrg[0]}</td>
        <td valign="top">{rowbrg[1]}</td>
        <td valign="top">{rowbrg[2]}</td>
        <td valign="top">{rowbrg[3]}</td>
        <td valign="top"><a href="BarangServlet?aksi=HAPUS&kode={rowbrg[0]}"> Hapus</a> | <a href="beranda.jsp?halaman=tampilbrg&aksi=GANTI&kd_brg={rowbrg[0]}&nm_brg={rowbrg[1]}&harga={rowbrg[2]}&stok={rowbrg[3]}"> Edit </a>
      </td>
    </tr>
    </c:forEach>
  </table>

```

## C. Source Code Servlet Barang

```
package control;

import java.io.IOException;
import java.io.PrintWriter;
import static java.lang.System.out;
import java.sql.SQLException;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import model.barang;

public class BarangServlet extends HttpServlet{

    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {
            String aksi = request.getParameter("aksi");
            String kodebarang=request.getParameter("kode");
            String namabarang=request.getParameter("nama");
            String harga=request.getParameter("harga");
            String stok=request.getParameter("stok");
            barang var = new barang(kodebarang,namabarang,harga,stok);
            String URL = "beranda.jsp?halaman=tampilbrg&aksi=SIMPAN";
            if(aksi.equals("SIMPAN")) {
                out.print(super.proses(var.toInsert(), URL, "SIMPAN"));
            } else if (aksi.equals("GANTI")) {
                out.print(super.proses(var.toUpdate(), URL, "GANTI"));
            } else if (aksi.equals("HAPUS")) {
                out.print(super.proses(var.toDelete(), URL, "HAPUS"));
            } else if (aksi.equals("HAPUS")) {
                out.print(super.proses(var.toDelete(), URL, "HAPUS"));
            }
        } catch (ClassNotFoundException ex) {
            Logger.getLogger(BarangServlet.class.getName()).log(Level.SEVERE, null, ex);
        } catch (SQLException ex) {
            Logger.getLogger(BarangServlet.class.getName()).log(Level.SEVERE, null, ex);
        } finally {
            out.close();
        }
    }
}

HttpServlet methods. Click on the + sign on the left to edit the code.
```

## D. Source Code Model Class Barang

```
package model;

public class barang {
    private String kodebarang,namabarang,harga,stok;

    public barang() {}

    public barang(String kodebarang, String namabarang, String harga, String stok) {
        this.kodebarang = kodebarang;
        this.namabarang = namabarang;
        this.harga = harga;
        this.stok = stok;
    }

    public String getKodebarang() {
        return kodebarang;
    }

    public void setKodebarang(String kodebarang) {
        this.kodebarang = kodebarang;
    }

    public String getNamabarang() {
        return namabarang;
    }

    public void setNamabarang(String namabarang) {
        this.namabarang = namabarang;
    }

    public String getHarga() {
        return harga;
    }

    public void setHarga(String harga){
        this.harga = harga;
    }

    public String getStok() {
        return stok;
    }

    public void setStok(String stok){
        this.stok = stok;
    }

    public String selectAll() {
        String sql = "select * from barang ";
        return sql;
    }

    public String select() {
        String sql = "select * from barang where "
            + "nm_brg='" + namabarang + "'";
        return sql;
    }

    public String toUpdate() {
        return " UPDATE barang SET " +
            "nm_brg='" + namabarang +
            "', harga='" + harga +
            "', stok='" + stok +
            "'WHERE kd_brg='" + kodebarang + "'";
    }

    public String toDelete() {
        return " DELETE FROM barang "
            + "WHERE kd_brg='" + kodebarang + "'";
    }

    public String toInsert() {
        return " INSERT INTO barang VALUES ('"
            + kodebarang + "','"
            + namabarang + "','"
            + harga + "','"
            + stok + "')";
    }
}
```

## E. Source Code Koneksi

```
package control;

import java.sql.Connection;
import java.sql.*;
import java.util.logging.Level;
import java.util.logging.Logger;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;

public class koneksi {

    Connection conn = null;
    public Statement stmt = null;
    ResultSet rs;

    public static void ambilkoneksi() {
        try {
            String db = "jdbc:mysql://localhost:3306/sewasepeda";
            String user = "root";
            String pass = "";
            Class.forName("com.mysql.jdbc.Driver");
            Connection con = DriverManager.getConnection(db, user, pass);
        } catch (Exception x) {
            System.out.println("Terjadi kesalahan ambil koneksi : " + x );
        }
    }

    public koneksi() throws ClassNotFoundException {
        String dbName = "sewasepeda";
        String uName = "root";
        String pass = "";
        String URL = "jdbc:mysql://localhost/" + dbName;

        String dbName = "sewasepeda";
        String uName = "root";
        String pass = "";
        String URL = "jdbc:mysql://localhost/" + dbName;
        Class.forName("com.mysql.jdbc.Driver");
        try {
            conn = DriverManager.getConnection(URL, uName, pass);
            stmt = conn.createStatement();
        } catch (SQLException ex) {
            Logger.getLogger(koneksi.class.getName()).log(Level.SEVERE, null, ex);
        }
    }

    public koneksi(String dbName,String uName, String pass) throws ClassNotFoundException {
        String URL = "jdbc:mysql://localhost/" + dbName;
        Class.forName("com.mysql.jdbc.Driver");
        try {
            conn = DriverManager.getConnection(URL, uName, pass);
            stmt = conn.createStatement();
        } catch (SQLException ex) {
            Logger.getLogger(koneksi.class.getName()).log(Level.SEVERE, null, ex);
        }
    }

    public void bukakoneksi() {
        try {
            String db = "jdbc:mysql://localhost:3306/sewasepeda";
            String user = "root";
            String pass = "";
            Class.forName("com.mysql.jdbc.Driver");
            conn = DriverManager.getConnection(db, user, pass);
            stmt = conn.createStatement();

            String db = "jdbc:mysql://localhost:3306/sewasepeda";
            String user = "root";
            String pass = "";
            Class.forName("com.mysql.jdbc.Driver");
            conn = DriverManager.getConnection(db, user, pass);
            stmt = conn.createStatement();
        } catch (Exception x) {
            System.out.println("Terjadi kesalahan koneksi : " + x );
        }
    }

    public void diskonek(ResultSet rs) {
        try {
            if (rs != null) {
                rs.close();
            }
            stmt.close();
            conn.close();
        } catch (Exception x) {
            System.out.println("Terjadi kesalahan diskoneksi : " + x);
        }
    }

    public ResultSet ambilData(String sql) {
        ResultSet rs = null;
        try {
            bukakoneksi();
            rs = stmt.executeQuery(sql);
        } catch (Exception x) {
            System.out.println("Terjadi kesalahan ambil data : " + x);
        }
        return rs;
    }
}
```



```

    }
    return rs;
}

public void close() throws SQLException {
    conn.close();
    stmt.close();
}
}

```

## F. Source Code Pesan

```

package control;

import java.sql.SQLException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;

/**
 *
 * @author ASUS
 */
@WebServlet(name = "pesan", urlPatterns = {"/pesan"})
public class pesan extends HttpServlet {

    public String proses(String SQL, String URL, String status)
        throws ClassNotFoundException, SQLException {
        koneksi kon = new koneksi();
        kon.stmt.executeUpdate(SQL);
        kon.close();
        String result = "<script>"
            + "alert('Data berhasil di " + status + "');"
            + "window.location='" + URL + "';"
            + "</script>";
        return result;
    }

    public String proses2 (String SQL, String URL, String status)
        throws ClassNotFoundException, SQLException {
        koneksi kon = new koneksi();
        kon.stmt.executeUpdate (SQL);
        kon.close();
        String result = "<script>"
            + "alert('Data Barang Berhasil Di " + status + "');"
            + "window.location='" + URL + "';"
            + "</script>";
        return result;
    }

    public void proses(String SQL) throws ClassNotFoundException,
        SQLException {
        koneksi kon = new koneksi();
        kon.stmt.executeUpdate(SQL);
        kon.close();
    }
}

```

## G. Source Code Login

```
package control;
```

```
import java.io.IOException;
import java.io.PrintWriter;
import static java.lang.System.out;
import java.util.ArrayList;
import java.util.List;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import model.user;
```

```
@WebServlet(name = "Login", urlPatterns = {"/Login"})
public class Login extends HttpServlet {
```

```
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException{
        String proses=request.getParameter("proses");
        if (proses.equals("logout")){
            HttpSession sesi=request.getSession();
            sesi.invalidate();
            response.sendRedirect("indexspd.jsp");
        }
    }
```

```
    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException{
        String proses=request.getParameter("proses");
```

```
        throws ServletException, IOException{
        String proses=request.getParameter("proses");
        if (proses.equals("login")){
            String user = request.getParameter("username");
            String pass = request.getParameter("password");
            String hak_akses = request.getParameter("hak_akses");
            String id = request.getParameter("id");
            if (pass == null || pass.equals("")){
                response.sendRedirect("indexspd.jsp");
            } else {
                user pm = new user();
                List<user> datalogin = new ArrayList<user>();

                datalogin = pm.LoginUser(user, pass);
                if (datalogin.isEmpty()){
                    response.sendRedirect("indexspd.jsp");
                } else {
                    HttpSession session = request.getSession(true);
                    session.setAttribute("username", datalogin.get(0).getIduser());
                    session.setAttribute("id", datalogin.get(0).getId());
                    session.setAttribute("hak_akses", datalogin.get(0).getHakakses());
                    if(datalogin.get(0).getHakakses().equals("Admin")){
                        response.sendRedirect("beranda.jsp");
                    }
                    else if (datalogin.get(0).getHakakses().equals("User")){
                        response.sendRedirect("beranda2.jsp");
                    }
                }
            }
        }
    }
}
```

## H. Source Code AutoKode

```
<%@page import="java.sql.*"%>
<%@page import="control.koneksi"%>
<!-- Auto Number No Pemesanan -->
<%
    try {
        koneksi kon = new koneksi();
        Class.forName("com.mysql.jdbc.Driver");
        ResultSet nopesan = null;
        nopesan = kon.stmt.executeQuery("select max(right(no_pesanan,6)) as no FROM pemesanan");
        while (nopesan.next()) {
            if (nopesan.first() == false) {
                out.println("<input type='hidden' name='no_pesanan' value='PS000001' id='nopesan' />");
            } else {
                nopesan.last();
                int autonops = nopesan.getInt(1) + 1;
                String nomorps = String.valueOf(autonops);
                int noLong = nomorps.length();

                for (int a = 1; a < 7 - noLong; a++) {
                    nomorps = "0" + nomorps;
                }
                String nomerps = "PS" + nomorps;
                out.println("<input type='hidden' name='no_pesanan' id='nopesan' value='" + nomerps + "' />");
            }
        }
    } catch (Exception e) {
        out.println(e);
    }
%>
```

## I. Source Code Setting

```
function hanyaAngka(evt) {
    var charCode = (evt.which) ? evt.which : event.keyCode
    if (charCode > 31 && (charCode < 48 || charCode > 57))
        return false;
    return true;
}

function validasi_inputBarang(form) {
    if (form.kode.value == "") {
        alert("Kode Barang masih kosong!");
        form.kode.focus();
        return (false);
    } else if (form.nama.value == "") {
        alert("Nama Barang masih kosong!");
        form.nama.focus();
        return (false);
    } else if (form.harga.value == "") {
        alert("Harga masih kosong!");
        form.harga.focus();
        return (false);
    } else if (form.stok.value == "") {
        alert("Stok masih kosong!");
        form.stok.focus();
        return (false);
    }
    return (true);
}

function validasi_inputCustomer(form) {
    if (form.kodecust.value == "") {
        alert("Kode Customer masih kosong!");
        form.kodecust.focus();
        return (false);
    }
}
```

```

function validasi_inputCustomer(form) {
    if (form.kodecust.value == "") {
        alert("Kode Customer masih kosong!");
        form.kodecust.focus();
        return (false);
    } else if (form.namacust.value == "") {
        alert("Nama Customer masih kosong!");
        form.namacust.focus();
        return (false);
    } else if (form.alamat.value == "") {
        alert("Alamat masih kosong!");
        form.alamat.focus();
        return (false);
    } else if (form.tlp.value == "") {
        alert("Telepon masih kosong!");
        form.tlp.focus();
        return (false);
    }
    return (true);
}

function validasi_inputAkun(form) {
    if (form.noakun.value == "") {
        alert("Nomor Akun masih kosong!");
        form.noakun.focus();
        return (false);
    } else if (form.nmakun.value == "") {
        alert("Nama Akun masih kosong!");
        form.nmakun.focus();
        return (false);
    }
    return (true);
}

function validasi_inputUser(form) {
    if (form.iduser.value == "") {
        alert("Id User masih kosong!");
        form.iduser.focus();
        return (false);
    } else if (form.nmuser.value == "") {
        alert("Nama User masih kosong!");
        form.nmuser.focus();
        return (false);
    } else if (form.hakakses.value == "") {
        alert("Hak Akses masih kosong!");
        form.hakakses.focus();
        return (false);
    } else if (form.password.value == "") {
        alert("Password masih kosong!");
        form.password.focus();
        return (false);
    }
    return (true);
}

function pesan() {
    document.getElementById("nopesan").value = document.getElementById("nopes").value;
}

function retur() {
    document.getElementById("noretur").value = document.getElementById("noret").value;
}

function showEmp(emp_value)
{
    function showEmp(emp_value)
    {
        if (document.getElementById("emp_id").value != "-1")
        {
            xmlhttp = GetXmlHttpRequestObject();
            if (xmlhttp == null)
            {
                alert("Browser does not support HTTP Request");
                return
            }
            var url = "getbarang.jsp"
            url = url + "?emp_id=" + emp_value

            xmlhttp.onreadystatechange = stateChanged
            xmlhttp.open("GET", url, true)
            xmlhttp.send(null)
        } else
        {
            alert("Pilih Kode Barang");
        }
    }

    function stateChanged()
    {
        document.getElementById("ename").value = "";
        document.getElementById("emp_id").value = "";
        if (xmlhttp.readyState == 4 || xmlhttp.readyState == "complete")
        {
            var showdata = xmlhttp.responseText;
            var strar = showdata.split(":");
            if (strar.length == 1)
            {

```

```

var showdata = xmlHttp.responseText;
var strar = showdata.split(":");
if (strar.length == 1)
{
    document.getElementById("emp_id").focus();
    alert("Pilih Kode Barang");
    document.getElementById("ename").value = " ";
    document.getElementById("emp_id").value = " ";
} else if (strar.length > 1)
{
    document.getElementById("ename").value = strar[1];
}
}
}

function GetXmlHttpRequestObject()
{
    var xmlHttp = null;
    try
    {
        xmlHttp = new XMLHttpRequest();
    } catch (e)
    {
        try
        {
            xmlHttp = new ActiveXObject("Msxml2.XMLHTTP");
        } catch (e)
        {
            xmlHttp = new ActiveXObject("Microsoft.XMLHTTP");
        }
    }
    return xmlHttp;
}

function sumPesanan() {
    var txtFirstNumberValue = document.getElementById('ename').value;
    var txtSecondNumberValue = document.getElementById('jml').value;
    var result = parseFloat(txtFirstNumberValue) * parseFloat(txtSecondNumberValue);
    if(!isNaN(result)) {
        document.getElementById('subtotal').value = result;
    }
}

function sumRetur() {
    var txtFirstNumberValue = document.getElementById('ename').value;
    var txtSecondNumberValue = document.getElementById('jml').value;
    var result = parseFloat(txtFirstNumberValue) * parseFloat(txtSecondNumberValue);
    if(!isNaN(result)) {
        document.getElementById('subtotal').value = result;
    }
}
}

```

## J. Source Code Index.jsp

```

<%%@page contentType="text/html" pageEncoding="UTF-8" %>
<!DOCTYPE html>
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Halaman Login</title>
    <style type="text/css">
        html, body{
            height: 160px;
            width: 100%;
            color: black;
        }
        table{
            margin: 0 auto;
            background : #DDA0DD;
        }
        h1{
            text-align: center;
        }
    </style>
</head>
<body>
    <h1>Login</h1>
    <form method="post" action="Login?proses=login">
        <table border="1" width="30%" cellpadding="3">
            <thead>
                <tr>
                    <th colspan="2">Login Aplikasi</th>
                </tr>
            </thead>
            <tbody>
                <tr>
                    <td>Username</td>

```

```

<tr>
  <td>Username</td>
  <td><input type="text" name="username"/></td>
</tr>
<tr>
  <td>Password</td>
  <td><input type="password" name="password"/></td>
</tr>
<tr>
  <td><input type="submit" value="Login"/></td>
  <td><input type="reset" value="Reset"/></td>
</tr>
</tbody>
</table>
</form>
</body>
</html>

```

## K. Source Code Cetak Laporan

```

<%%page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Cetak Laporan Sepeda</title>
  </head>
  <body>
    <h2>Cetak Laporan Sepeda Berdasarkan Stok Tersedia</h2>
    <form action="TampilLapBarang.jsp" method="get">
      stok <= <input type="number" name="jumlah" value="0">
      <input type="submit" name="submit" value="Cetak Laporan"/>
    </form>
  </body>
</html>

```

## L. Source Code Tampil Laporan

```

<%%page import="control.koneksi"%>
<%%page import="java.io.*, java.util.*, java.sql.*"%>
<%%page import="net.sf.jasperreports.engine.*"%>
<%%page import="net.sf.jasperreports.view.JasperViewer.*"%>
<%%page import="javax.servlet.ServletResponse" %>
<%%page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Laporan Barang</title>
  </head>
  <body>
    <%
      koneksi kon = new koneksi();
      int jumlah = Integer.parseInt(request.getParameter("jumlah"));
      File reportFile = new File(application.getRealPath("laporan/Laporan Sepeda.jasper"));
      Map param = new HashMap();
      param.put("pstok", jumlah);
      byte[] bytes = JasperRunManager.runReportToPdf(reportFile.getPath(), param, kon.stmt.getConnection());
      response.setContentType("application/pdf");
      response.setContentLength(bytes.length);

      ServletOutputStream outStream = response.getOutputStream();
      outStream.write(bytes, 0, bytes.length);
      outStream.flush();
      outStream.close();
    %>
  </body>
</html>

```

## J. Source Code Transaksi.jsp

```

<@page import="java.sql.ResultSet"%>
<@page import="control.koneksi"%>
<@page import="model.barang,model.customer,model.pemesanan"%>
<@page import="java.util.Date"%>
<@include file="config/autokode.jsp"%>
<@page contentType="text/html" pageEncoding="UTF-8"%>
<%
    java.util.Date waktu = new java.util.Date();
    int tanggal = waktu.getDate();
    int tahun = waktu.getYear() + 1900;
    int bulan = waktu.getMonth() + 1;
    String tgl = tahun + "-" + bulan + "-" + tanggal;
    pemesanan pemesanan = new pemesanan();
    barang barang = new barang();
    customer customer = new customer();
    koneksi kon = new koneksi();
    ResultSet rs = null;
    ResultSet qrybarang = null;
    ResultSet qrypemesanan = null;
%>

<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title> Transaksi Pemesanan</title>
</head>
<body>
<sql:query var="ttl" dataSource="${dataSource}">
    SELECT SUM(quantity) AS ttl_D, SUM(subtotal) AS ttl_K FROM sementara
</sql:query>
<h1>Transaksi Pemesanan Barang</h1></br>
<table border="0">
    <form action="ServletPemesanan" method="post">
        <h2>Data Barang</h2>
        <tr>
            <td>No Pesan</td>
            <td>: <input type="text" name="no_pesan" id="nopesan" value="" readonly="readonly"/>
            <input type="button" onclick="pesan()" value="Add No Pesan" class="button"></td>
        </tr>
        <tr>
            <td>Tanggal Pesan</td>
            <td>: <input type="text" name="tanggal" value="<%= tgl %>" readonly="readonly"></td>
        </tr>
        <tr>
            <td>Barang</td>
            <td>: <select name="kode" onchange="showEmp(this.value);">
                <option value="-1">Pilih Barang</option>
                <%
                    rs = kon.stmt.executeQuery("SELECT * from barang ");
                    while (rs.next()) {
                        barang.setKodebarang(rs.getString("kd_brng"));
                        barang.setNamabarang(rs.getString("nm_brng"));
                    }
                %>
                <option value="<%= barang.getKodebarang() %>"><%= barang.getKodebarang() %> || <%= barang.getNamabarang() %></option>
                <% } %>
            </select></td>
        </tr>
        <tr>
            <td>Harga Barang</td>
            <td>: <input type="text" name="harga" id="ename" value="" readonly="readonly" onkeyup="sumPesan();"></td>
        </tr>
        <tr>
            <td>Jumlah Pesan</td>
            <td>: <input type="text" name="jml" id="jml" onkeyup="sumPesan();"></td>
        </tr>
        <tr>
            <td>Subtotal</td>
            <td>: <input type="text" name="sub" id="subtotal" readonly="readonly"></td>
        </tr>
    </form>
    </table>

```

```

        </tr>
        <tr>
            <td>Subtotal</td> <td>: <input type="text" name="sub" id="subtotal" readonly="readonly"></td>
        </tr>
        <tr>
            <td><input type="submit" value="TAMBAH" name="aksi1"></td>
        </tr>
    </table>
    <h2>Data Customer</h2>
    Supplier <select name="customer">
        <option value="-1">Pilih Customer</option>
        <%
            rs = kon.stmt.executeQuery("SELECT * from customer ");
            while (rs.next()) {
                customer.setKodecust(rs.getString("kd_cust"));
                customer.setNamacust(rs.getString("nm_cust"));
            %>
        <option value="<%=customer.getKodecust() %>"><%=customer.getNamacust() %></option>
        <% } %>
    </select><br><br>

    <table width="535" border="1" align="center">
        <tr align="center">
            <td colspan="5"><h3>Data Barang Untuk Dipesan</h3></td>
        </tr>
        <tr>
            <th>Kode Barang</th>
            <th>Quantity</th>
            <th>Subtotal</th>
            <th>AKSI</th>
        </tr>
    </table>
    <%

```

```

        <th>Kode Barang</th>
        <th>Quantity</th>
        <th>Subtotal</th>
        <th>AKSI</th>
    </tr>
    <%
        rs = kon.stmt.executeQuery("SELECT * from sementara");
        while (rs.next()) {
            out.println("<tr class=isi>"
                + "<td>" + rs.getString(2) + "</td>"
                + "<td>" + rs.getString(3) + "</td>"
                + "<td>" + rs.getString(4) + "</td>"
                + "<td><a href=ServletPemesanan?aksi2=HAPUS&kode=" + rs.getString(2) + ">Hapus</a></td>"
                + "</tr>");
        }
    %>
</table>
<c:forEach var='ttlDK' items='${ttl.rowsByIndex}'><br>
    <table border="0">
        <tr><td>Total Jumlah Barang Dipesan</td> <td>: <input type="text" readonly="readonly" name="totjum" size="1" value="${ttlDK[0]}"></td></tr>
        <tr><td>Total Bayar</td> <td>: <input type="text" readonly="readonly" name="tobay" value="${ttlDK[1]}"></td></tr>
    </c:forEach>
    <%kon.close();%>
    <tr><td><input type="submit" value="PROSES" name="aksi3"> <input type="reset" value="CANCEL"></td></tr>
    </table>
</form>
</body>
</html>

```



## J. Source Code Transaksi Servlet

```

2 package control;
3
4 import java.io.IOException;
5 import java.io.PrintWriter;
6 import static java.lang.System.out;
7 import java.sql.SQLException;
8 import java.util.logging.Level;
9 import java.util.logging.Logger;
10 import javax.servlet.ServletException;
11 import javax.servlet.http.HttpServlet;
12 import javax.servlet.http.HttpServletRequest;
13 import javax.servlet.http.HttpServletResponse;
14 import model.pemesanan;
15 import java.sql.*;
16
17 public class ServletPemesanan extends pesan {
18
19     protected void processRequest(HttpServletRequest request, HttpServletResponse response)
20         throws ServletException, IOException {
21         response.setContentType("text/html;charset=UTF-8");
22         try (PrintWriter out = response.getWriter()) {
23             pemesanan pemesanan = new pemesanan();
24             PreparedStatement pstmt = null;
25             PreparedStatement pstmt1 = null;
26             PreparedStatement pstmt2 = null;
27             koneksi kon = new koneksi();
28             int result = 0;
29             int result1 = 0;
30             int result2 = 0;
31             String res;
32             String aksi = request.getParameter("aksi");
33
34             int result1 = 0;
35             int result2 = 0;
36             String res;
37             String aksi = request.getParameter("aksi");
38             String nopesan = request.getParameter("no_pesan");
39             String tglpesan = request.getParameter("tanggal");
40             String kdcust=request.getParameter("customer");
41             String kdbarang=request.getParameter("kode");
42             String total=request.getParameter("tobay");
43             String qty=request.getParameter("jml");
44             String subtotal=request.getParameter("sub");
45             String aksi1=request.getParameter("aksi1");
46             String aksi2=request.getParameter("aksi2");
47             String aksi3=request.getParameter("aksi3");
48             pemesanan var = new pemesanan(nopesan, tglpesan, kdcust, kdbarang, total, qty, subtotal);
49             String URL = "beranda.jsp?halaman=pesan&aksi=SIMPAN";
50             if (aksi1 != null) {
51                 out.print(super.proses2(var.toInsert(), URL, "TAMBAH"));
52             } else if (aksi2 != null) {
53                 out.print(super.proses2(var.toDelete(), URL, "HAPUS"));
54             } else if (aksi3 != null) {
55                 pemesanan.setNopesan(request.getParameter("no_pesan"));
56                 pemesanan.setTglpesan(request.getParameter("tanggal"));
57                 pemesanan.setKdcust(request.getParameter("customer"));
58                 pemesanan.setTotal(request.getParameter("tobay"));
59                 pstmt=kon.conn.prepareStatement("INSERT detail_pesan SELECT '"+pemesanan.getNopesan()+"', "
60                     + "kode,quantity,subtotal FROM sementara");
61                 result=pstmt.executeUpdate();
62                 pstmt1 = kon.conn.prepareStatement("INSERT INTO pemesanan values('"+ pemesanan.getNopesan() + "','"+
63                     + pemesanan.getTglpesan() + "','"+
64                     + pemesanan.getTotal() + "','"+
65                     + pemesanan.getKdcust() + "')");
66                 result1 = pstmt1.executeUpdate();

```

```

result1 = pstmt1.executeUpdate();
pstmt2 = kon.conn.prepareStatement("TRUNCATE TABLE sementara");
result2=pstmt2.executeUpdate();
    if (result +result1 + result2 > 0) {
        out.println("<script> "
            + "alert('Data telah disimpan');"
            + "document.location='beranda.jsp?hal=pesan';"
            + " </script>");
    }

}

} catch(ClassNotFoundException ex){
    Logger.getLogger(BarangServlet.class.getName()).log(Level.SEVERE, null, ex);
} catch(SQLException ex){
    Logger.getLogger(BarangServlet.class.getName()).log(Level.SEVERE, null, ex);
} finally {
    out.close();
}
}

```



HttpServlet methods. Click on the + sign on the left to edit the code.

```

}

```

### 3.3.2 Blackbox Testing

Pengujian dilakukan terhadap program yang dibuat menggunakan *blackboxtesting* yang fokus terhadap proses masukan dan keluaran program.

#### 1. Hasil pengujian Tahap *Form Login Admin*

No	Skenario pengujian	Test Case	Hasil Yang diharapkan	Hasil pengujian	kesimpulan
1	Username dan password tidak diisi kemudian klik tombol login	Username: (kosong) Password: (kosong)	Sistem akan menolak akses dan menampilkan "masukan username!"	Sesuai harapan	Valid
2	Username dan password tidak diisi kemudian klik tombol login	Username: admin Password: (kosong)	Sistem akan menolak akses user dan menampilkan "masukan password!"	Sesuai harapan	Valid
3	Username dan password tidak diisi kemudian klik tombol login	Username: (kosong) Password: admin	Sistem akan menolak akses user dan menampilkan "masukan username!"	Sesuai harapan	Valid
4	Mengetikan salah satu kondisi salah pada usernameatau password kemudian klik tombol login	Username: tono (salah) Password: admin (benar)	Sistem akan menolak akses user dan menampilkan "Maaf, username / password salah!" Sesuai	Sesuai harapan	Valid
5	Mengetikan usernameatau password yang benar kemudian klik tombol login	Username: admin (benar) Password: admin	Sistem akan menerima akses login dan menampilkan "Selamat datang, Administator!", kemudian masuk ke beranda admin	Sesuai harapan	Valid

#### 2. Hasil pengujian blackbox testing form Transaksi

No	Skenario Pengujian	Tast Case	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Belum melakukan login	Klik Tombol Beli Pada Item Yang Dipilih	Sistem akan mengarahkan pengunjung untuk melakukan registrasi atau member melakukan login terlebih dahulu untuk melanjutkan transaksi	Sesuai Harapan	Valid

2	Salah satu field kosong kemudian isi klik pilihan kurir	Alamat : (kosong) Provinsi : DKI Jakarta Kota : Jakarta Barat No.Telepon / HP :085716088528	Sistem akan menampilkan "jenis layanan masih kosong	Sesuai Harapan	Valid
3	Semua field diisi kemudian klik tombol jenis layanan	Alamat : kp utan no 54 A rt 03 rw 12 cengkareng timur Provinsi : DKI Jakarta Kota : Jakarta Barat No.Telepon / HP :085716088528	Sistem menerima akses dan lanjutkan pembayaran ,akan menampilkan "Transaksi berhasil .Silahkan lanjutan ke pembayaran !"	Sesuai Harapan	Valid

## **BAB IV**

### **PENUTUP**

#### **Kesimpulan**

1. Memudahkan sistem pengolahan data-data barang, customer, sewa, user, pemesanan, dan retur pemesanan
2. Dalam pengolahan data dapat dilakukan secara teliti, aman, dan terhindar dari hal-hal yang tidak kita inginkan sehingga data tersebut dapat kita baca kembali untuk melakukan perbaikan data bila kita perlukan.
3. Data-data yang tersimpan didalam database dapat dicetak apabila sewaktu-waktu diperlukan.

#### **Saran**

Beberapa hal yang penulis sarankan , yaitu:

1. Pengembangan lebih lanjut untuk terus menyempurnakan program ini
2. Penggunaan antivirus untuk menjaga keamanan data dan software dari kerusakan sistem
3. Pengguna komputer secara wajar untuk menjaga komputer dan seluruh sistem didalamnya lebih awet dan tahan lama.

## DAFTAR PUSTAKA

<https://www.codepolitan.com/10-bahasa-pemrograman-populer-di-indonesia>

[https://repository.bsi.ac.id/index.php/unduh/item/61928/File\\_14-BAB-II-Landasan-Teori.pdf](https://repository.bsi.ac.id/index.php/unduh/item/61928/File_14-BAB-II-Landasan-Teori.pdf)

<http://hawarimuhtarom.blogspot.com/2017/12/sequence-diagram-penyewaan-pengembalian-laporan-pt-bendi-car.html>

[https://elib.unikom.ac.id/files/disk1/767/jbptunikompp-gdl-rizkinim10-38315-1-unikom\\_r-l.pdf](https://elib.unikom.ac.id/files/disk1/767/jbptunikompp-gdl-rizkinim10-38315-1-unikom_r-l.pdf)