

## Aplikasi Penyewaan Ruang Gudang

**Soviani, Ilham Rahman, Ulfah Ulmi, Indrawan Dwi Suyanto<sup>1</sup>, Dahliar Ananda, S.T, M.T.<sup>2</sup>**  
<sup>1</sup>Telkom University, Mahasiswa Manajemen Informatika, <sup>2</sup>Telkom University, Dosen Pembimbing

### Abstrak

Ruang gudang merupakan suatu tempat layanan bagi orang-orang yang ingin menyimpan barang-barang. Saat ini masih banyak tempat penyewaan gudang yang masih bersifat *manual*, sehingga tingkat kesalahan dalam mencari informasi ketersediaan ruang gudang, mengolah data penyewa dan transaksi masih tinggi. Pembangunan aplikasi penyewaan ruang gudang dilakukan untuk melihat ketersediaan ruang gudang yang akan disewakan bagi pengguna yang membutuhkannya. Aplikasi ini dapat mengelola data penyewa, mengelola transaksi, pembuatan laporan transaksi dan melihat grafik pendapatan. Aplikasi ini juga memiliki pembatasan hak akses yang berbeda terhadap *superadmin*, *admin* dan pegawai. Sehingga dengan adanya aplikasi ini diharapkan mampu mempermudah dan mempercepat kinerja perusahaan maupun dalam melayani penyewa gudang. Adapun dalam pembangunan aplikasi ini dengan menggunakan *tools Java Server Pages, JQuery, Javascript, Highcharts, Cascading Style Sheet (CSS)* serta menggunakan database *MySQL*.

**Kata Kunci : Sewa Gudang, Transaksi, Laporan, Tools**

### Abstract

*Warehouse is a place of service for people who want to store stuff. Currently there are many rental places are still be manually, so the error rate in finding warehouse availability information, process data and transaction tenant is still high. Construction of warehouse rental application is made to see the availability of warehouse that will be leased to the users who need it. This application can manage the data of tenants, manage transactions, made transaction reports and view graphs of income. This application also has different restrictions on access rights to superadmin, admin and employee. So with this application is expected to simplify and accelerate the company's performance in serving tenants and warehouses. The development of this application using the tools Java Server Pages, JQuery, Javascript, Highcharts, Cascading Style Sheets (CSS) and using MySQL database.*

**Keywords: Warehouse rental, Transaction, Report, Tools**

## Bab 1 Pengenalan Produk

### 1.1. Latar Belakang

Di era globalisasi sekarang, kebutuhan akan informasi semakin cepat dan terus berkembang. Dengan adanya teknologi kita dapat mencari informasi yang kita butuhkan dengan lebih cepat. Berbagai perusahaan mulai dari perusahaan besar, menengah maupun kecil sudah mulai melakukan perubahan. Dengan memanfaatkan kecanggihan teknologi seperti menerapkan sistem komputerisasi dalam melakukan pendataan pada perusahaan, sehingga dapat memperoleh informasi yang lebih lengkap, tepat dan akurat.

Ruang gudang merupakan suatu tempat layanan bagi orang-orang yang ingin menyimpan barang. Saat ini masih banyak tempat penyewaan gudang yang masih bersifat manual, sehingga tingkat kesalahan dalam mencari informasi dan ketersediaan ruang gudang masih tinggi. Pengelolaan transaksi, mengatur data penyewa dan pembuatan laporan transaksi yang bersifat manual juga dapat menghambat kinerja operasional suatu perusahaan. Begitu juga pembatasan akses pada pengelolaan transaksi yang harus diperhatikan. Maka dari itu, pengelolaan ruang gudang tersebut membutuhkan suatu sistem berbasis *web* yang dapat melihat ketersediaan ruang gudang.

Oleh karena itu, diperlukan suatu aplikasi yang berfungsi dalam melihat ketersediaan ruang gudang yang akan disewakan. Aplikasi yang dapat mengelola data penyewa, mengelola transaksi, mencetak laporan transaksi dan melihat grafik pendapatan serta dapat melakukan pembatasan akses.

### 1.1 Tujuan

Adapun tujuan dari pembuatan karya akhir ini adalah sebagai berikut.

1. Membuat aplikasi yang dapat mengatur data penyewaan gudang
2. Membuat aplikasi yang dapat mengolah transaksi penyewaan gudang
3. Membuat aplikasi yang dapat mencetak laporan transaksi
4. Membuat aplikasi yang dapat menampilkan laporan dan grafik perkembangan perusahaan penyewaan gudang sesuai waktu yang diinginkan
5. Membuat aplikasi yang memiliki hak akses yang berbeda sesuai dengan levelnya.

### 1.2 Batasan Produk

Batasan masalah dari produk ini adalah sebagai berikut.

1. Aplikasi yang dibangun tidak menerima penyewaan ruang untuk waktu kedepan, hanya

- 3. Aplikasi yang dibangun tidak mengatur keuangan perusahaan penyewaan gudang
- 2. Aplikasi yang dibangun tidak mengatur pembayaran terhadap transaksi yang sedang berlangsung

**1.3 Sistematika Penulisan**

**Bab 1** menjelaskan latar belakang, tujuan, batasan produk dan sistematika penulisan.

**Bab 2** menjelaskan arsitektur produk.

**Bab 3** menjelaskan modul-modul produk.

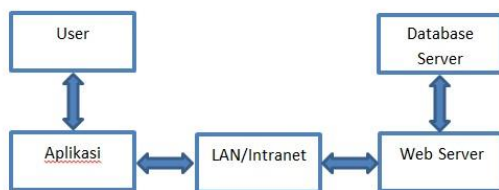
**Bab 4** menjelaskan tata cara instalasi dan penggunaan produk.

**Bab 5** berisi penutup yang terdiri dari hambatan yang dialami dan saran pengembangan produk.

**Bab 2**  
**Arsitektur Produk**

**2.1 Struktur Produk**

Aplikasi Penyewaan Gudang yang akan dibangun memiliki struktur seperti gambar di bawah ini.



**Gambar 2.1.1 Struktur Aplikasi Penyewaan Gudang**

Aplikasi diakses oleh *user* melalui sebuah komputer atau laptop. Aplikasi berinteraksi dengan *web server* melalui jaringan intranet atau LAN. Jika aplikasi membutuhkan data yang ada di basis data, maka *web server* akan berinteraksi dengan basis data *server*.

**2.2 Kebutuhan Sumber Daya Manusia**

Bagian ini berisi tentang kebutuhan sumber daya manusia yang dituliskan dalam bentuk tabel atau daftar beserta penugasan.

**Tabel 2.2.1 Tabel Kebutuhan Sumber Daya Manusia**

No	Sumber Daya Manusia	Posisi dan Penugasan
1	Soviani	<i>Analisis/Programmer/Integrator Modul Transaksi</i>
2	Ulfah Ulmi	<i>Analisis/Programmer/Integrator Modul Laporan Transaksi</i>
3	Indrawan	<i>Analisis/Programmer/Integrator</i>

	Dwi Suyanto	<i>Modul Data Master</i>
4	Ilham Rahman	<i>Analisis/Programmer/Integrator Modul Login and Interface</i>

**2.3 Tools yang Digunakan**

Dalam membangun aplikasi ini, dibutuhkan tools sebagai berikut.

1. XAMPP 1.7.0

XAMPP merupakan paket berbasis *Open Source* yang dikembangkan oleh sebuah komunitas *Open Source*. XAMPP dibutuhkan untuk membangun aplikasi berbasis *web* yang terdiri atas program *Apache HTTP Server*, *MySQL basis data*, dan penterjemah bahasa yang ditulis dengan bahasa pemrograman *PHP* dan *Perl*. [1]

2. Netbeans 7.4

Netbeans adalah sebuah proyek kode terbuka, yg digunakan untuk membuat software aplikasi computer maupun handphone, anda bisa berimajinasi dan berkarya tanpa batas disini. Didalam Netbeans terbaru beberapa plugin yang bisa di tambahkan adalah netbeans untuk membuat program *Java*, *C*, *C++*, *PHP* dan lainnya. Tinggal disesuaikan saja dengan yang kita butuhkan. Di netbeans bisa untuk *javascript*, menulis pemrograman *C* dan pemrograman *C++*. [2]

3. JQuery

*JQuery* merupakan suatu *framework (library) Javascript* yang menekankan bagaimana interaksi antara *Javascript* dan *HTML*. *JQuery* pertama kali dirilis pada tahun 2006 oleh *John Resig*. Pada perkembangannya *JQuery* tidak sekedar sebagai *framework Javascript*, namun memiliki kehandalan dan kelebihan yang cukup banyak. Hal tersebut menyebabkan banyak *developer web* menggunakannya. *JQuery* memiliki slogan “*Write less, do more*” yang kurang lebih maksudnya adalah kesederhanaan dalam penulisan *code*, tapi dengan hasil yang lebih banyak. [3]

4. Datatable

*DataTables* adalah *plug-in jQuery* untuk menampilkan data dalam bentuk tabel yang interaktif. *DataTables* dapat menerima sumber data dari sisi server dalam bentuk *JSON*. Ketika meminta data dari server, *DataTables* akan mengirimkan beberapa parameter dan server harus mengirimkan balasan sesuai format yang telah dispesifikasikan. [3]

5. Bootstrap

*Bootstrap* adalah *platform* baru yang dikembangkan tim twitter. Pertama kali muncul pada ajang *hackweek* dan kini sudah mulai penyempurnaan. *Platform* ini hanya menggunakan sedikit coding *CSS* dan *JavaScript* namun tetap bisa membuat *website* yang *powerfull* mengikuti perkembangan *browser*. *Website* yang menggunakan *bootstrap* akan menjadi *website* yang fleksibel, nyaman dan tentu saja cepat.

*Bootstrap* juga memudahkan pengaturan *website* bagi mereka yang kurang mahir *coding* atau tidak punya waktu banyak. Sekumpulan *javascript* dan CSS yang dibutuhkan untuk banyak sekali *widget* ada dalam satu file yang terus di *update* oleh pengembang. Pemilik *website* tidak akan lagi disibukkan dengan pengaturan *css* atau *javascript* berlebihan. Mereka hanya perlu menentukan *gadget* atau elemen apa yang ingin ditampilkan dalam bentuk *HTML* standar *bootstrap* yang sangat *simple*. [4]

#### 6. Highcharts

*Highcharts* adalah *charting library* berbasis *javascript* yang dibuat oleh perusahaan Highsoft Solution AS. *Highcharts* ini ada yang berbayar juga ada yang free. Dengan *Highcharts* kita dapat membuat beberapa bentuk grafik seperti *line Chart*, *pie*, *Bar Column* atau *Area Chart* dan masih banyak lagi. [5]

### 2.4 Tinjauan Pustaka

#### a. CSS (Cascading Style Sheet)

*Cascading Style Sheet (CSS)* merupakan salah satu bahasa pemrograman web untuk mengendalikan beberapa komponen dalam sebuah web sehingga akan lebih terstruktur dan seragam. CSS dapat mengendalikan ukuran gambar, warna bagian tubuh pada teks, warna tabel, ukuran border, warna border, warna *hyperlink*, *margin* kiri, kanan, atas, bawah, dan parameter lainnya. CSS adalah bahasa *style sheet* yang digunakan untuk mengatur tampilan dokumen. Dengan adanya CSS memungkinkan kita untuk menampilkan halaman yang sama dengan format yang berbeda.

Penggunaan yang paling umum dari CSS adalah untuk memformat halaman web yang ditulis dengan *HTML* dan *XHTML*. Spesifikasi CSS diatur oleh *World Wide Web Consortium (W3C)*. CSS memungkinkan halaman yang sama untuk ditampilkan dengan cara yang berbeda untuk metode presentasi yang berbeda, seperti melalui layar, cetak, suara, (sewaktu dibacakan oleh *browser*) dan juga alat pembaca *Braille*. Halaman *HTML* atau *XML* yang sama juga dapat ditampilkan secara berbeda, baik dari segi gaya tampilan atau skema warna dengan menggunakan CSS. [6]

#### b. MySQL

MySQL adalah suatu perangkat lunak *database* relasi (*Relational Database Management System* atau RDBMS), seperti halnya ORACLE, PostgreSQL, MS SQL, dan sebagainya. MySQL AB menyebut produknya sebagai *database open source* terpopuler di dunia. Berdasarkan riset dinyatakan bahwa di *platform Website*, dan baik untuk kategori *open source* maupun umum, MySQL adalah *database* yang paling banyak dipakai. Menurut perusahaan pengembangnya, MySQL telah terpasang di sekitar 3 juta komputer. Puluhan hingga ratusan ribu situs mengandalkan MySQL bekerja siang malam memompa data bagi para pengunjungnya. [6]

#### c. Apache

Apache [<http://www.apache.org>] merupakan web server yang terpopuler saat ini. Menurut survey netcraft lebih dari 50% situs di Internet menggunakan Apache sebagai Web server. Server Apache sangat luwes, memenuhi standar HTTP/1.1, mengimplementasikan protokol terbaru dan sangat luwes

dikonfigurasi serta dapat ditambahkan modul lainnya melalui module Apache. Tersedia untuk berbagai sistem operasi. Apache menyediakan fasilitas yang kaya, yang sangat dibutuhkan suatu server serius, seperti otentikasi, pengaturan akses direktori, virtual host, kemampuan URL rewriting, dan juga alias. Kemampuan melakukan *content negotiation* membuat Apache mampu melayani beragam client secara otomatis, baik untuk berbagai browser yang memiliki kemampuan berbeda ataupun untuk *device* akses yang berbeda. Fungsi log yang dimiliki oleh Apache dapat dikirim melalui proses, sehingga dapat dilakukan rotasi log, filter log, serta melakukan pemisahan log secara langsung.

Awalnya Apache dikembangkan berdasarkan keinginan untuk memperbaiki Web server yang saat itu populer (NCSA *web server*). Tetapi akhirnya mengalami perombakan dan penulisan ulang dan menjadi Web server yang berdiri sendiri dan berbeda dengan NCSA. Kini malah mengalahkan kepopuleran NCSA *Web server*. Pada tahun 1999 dibentuk Apache *Software Foundation* untuk mengurus perkembangan Apache ini. Apache telah membuktikan sebagai *web server* yang cepat, stabil dengan fitur yang paling kaya di antara *web server* lainnya. Saat ini proyek Apache telah berkembang dan tidak hanya sekedar *Web server*. [7]

#### d. Javascript

JavaScript adalah bahasa pemrograman ditafsirkan bahwa sebagian besar digunakan untuk mengubah halaman web statis menjadi halaman dinamis dan interaktif setelah *web browser* telah selesai men-*download* sebuah halaman web. Misalnya, jam ditampilkan pada halaman yang update sendiri untuk menunjukkan waktu saat ini pada komputer pengguna. Desain JavaScript dipengaruhi oleh banyak bahasa pemrograman, termasuk C, tetapi dimaksudkan untuk lebih digunakan oleh *non-programmer*. JavaScript tidak didasarkan pada atau terkait ke Java, ini adalah kesalahpahaman umum. JavaScript seringkali disertakan dalam file HTML atau *link* dari file HTML dan dijalankan secara lokal oleh *web browser*. Ini berarti bahwa server bebas untuk mengerjakan sesuatu yang lain daripada pemrosesan instruksi untuk setiap klien. Hal ini telah membuat JavaScript pilihan yang lebih populer daripada bahasa yang memerlukan server untuk melakukan pengolahan. [8]

#### e. Java Server PagesTM (JSP)

Java Server PagesTM (JSP) adalah teknologi Web berbasis bahasa pemrograman Java dan berjalan pada Platform Java yang diproduksi oleh *Sun microsystem*. JSP memungkinkan kita menggabungkan static HTML dengan dynamic content yang dihasilkan dari *Servlet*. JSP juga merupakan bagian dari teknologi J2EE (Java 2 Enterprise Edition). J2EE merupakan platform Java untuk pengembangan aplikasi enterprise dengan dukungan API (Application Program Interface) yang lengkap dan portabilitas serta memberikan sarana untuk membuat aplikasi *multi tier* yang memisahkan antara Presentation layer, Application layer dan Data Layer.

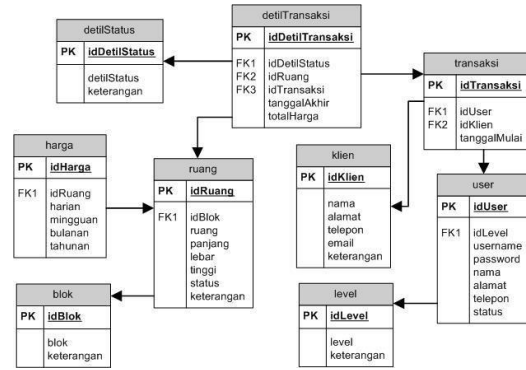
#### Keunggulan JSP

JSP mudah dipelajari dan memungkinkan developer membuat Web Site dan aplikasi dengan cepat. JSP berjalan diatas Java yang merupakan object oriented Programming sehingga JSP

berada pada platform yang kuat untuk pengembangan aplikasi. [9]

f. HTML

*HyperText Markup Language* (HTML) adalah sebuah *bahasa* yang digunakan untuk membuat sebuah halaman web, menampilkan berbagai informasi di dalam sebuah *penjelajah web* Internet dan pemformatan hiperteks sederhana yang ditulis dalam berkas format ASCII agar dapat menghasilkan tampilan wujud yang terintegrasi. Dengan kata lain, berkas yang dibuat dalam perangkat lunak pengolah kata dan disimpan dalam format *ASCII* normal sehingga menjadi *halaman web* dengan perintah-perintah HTML. Bermula dari sebuah bahasa yang sebelumnya banyak digunakan di dunia penerbitan dan percetakan yang disebut dengan *SGML* (*Standard Generalized Markup Language*), HTML adalah sebuah standar yang digunakan secara luas untuk menampilkan halaman web. HTML saat ini merupakan standar *Internet* yang didefinisikan dan dikendalikan penggunaannya oleh *World Wide Web Consortium* (W3C). HTML dibuat oleh kolaborasi Caillau TIM dengan Berners-lee Robert ketika mereka bekerja di CERN pada tahun 1989 (CERN adalah lembaga penelitian fisika energi tinggi di Jenewa). [10]

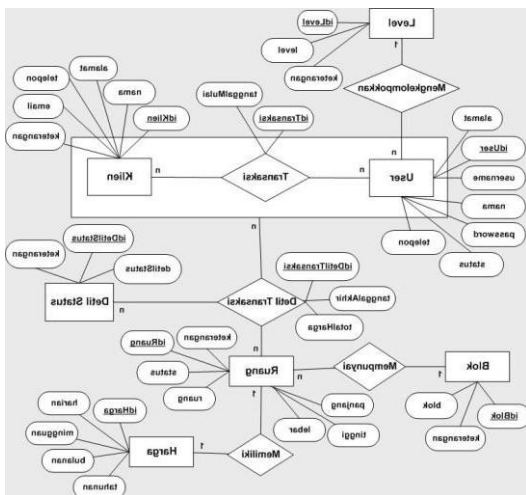


Gambar 3.2.1 Skema Relasi

Bab 3

Pembuatan Produk

3.1 Entity Relationship Diagram



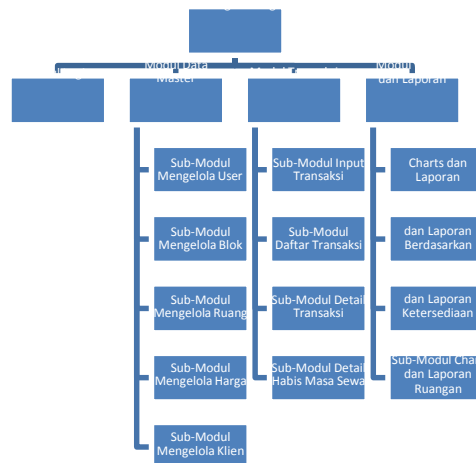
Gambar 3.1.1 Entity Diagram Relationship

3.2 Skema Relasi

Basis data dibangun guna memfasilitasi aplikasi dalam penyimpanan data. Basis data yang digunakan adalah MySQL. Pada aplikasi penyewaan gudang ini terdapat sembilan tabel basis data. Berikut ini relasi tabel dari masing-masing tabel beserta atributnya.

3.3 Struktur Modul

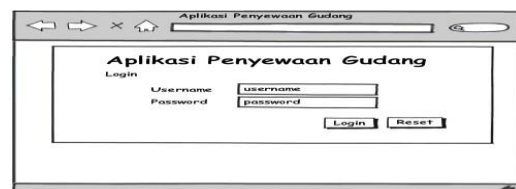
Aplikasi penyewaan gudang ini, terdapat 17 modul yang akan dibangun. Gambaran ke-17 modul tersebut dapat terlihat pada gambar di bawah ini.



Gambar 3.3.1 Struktur Modul dalam Aplikasi Penyewaan Ruang

3.3.1 Modul Login

Modul login digunakan sebagai autentikasi pengguna dalam menggunakan aplikasi. *Login* ini hanya bisa di akses oleh *Super Admin*, *Admin* dan pegawai. Setiap *user* mempunyai tingkatan *level* untuk mengakses.



Gambar 3.3.1.1 Desain Login

Berikut adalah potongan *source code* jsp untuk *login*.

```

<body>
  <div id="header">
    <div id="libram">
      <div class="account-container">
        <div class="alert alert-danger alert-dismissible" data-dismiss="alert">
          <div class="alert">
            <div class="close">
              <span></span>
            </div>
          </div>
          <div class="form-horizontal">
            <table>
              <tr>
                <td>username </td></td>
                <input type="text" name="username" required id="username" />
              </tr>
              <tr>
                <td>password </td></td>
                <input type="password" name="password" required id="password">
              </tr>
              <tr>
                <td colspan="2"><input type="submit" value="login"></td></tr>
            </table>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>

```

Gambar 3.3.1.3 Source code jsp login

```

private void processRequest(HttpServletRequest request, HttpServletResponse response) throws IOException {
  PrintWriter printwriter;
  try {
    printwriter = response.getWriter();
    if ("login".equals(request.getParameter("act"))) {
      request.getSession().setAttribute("login");
      response.sendRedirect("login.jsp");
    } else {
      String urlForward;
      String username = request.getParameter("username");
      String password = request.getParameter("password");
      int[] data = userLogin(username, password);
      System.out.println("Data[0]" + data[0]);
      System.out.println("Data[1]" + data[1]);
      System.out.println("Data[2]" + data[2]);
      System.out.println("Data[3]" + data[3]);
      if (data[0] == 1) {
        System.out.println("idlevel untuk " + username + " adalah " + data[0]);
        request.getSession().setAttribute("username", username);
        request.getSession().setAttribute("idlevel", data[0]);
        request.getSession().setAttribute("iduser", data[1]);
        request.getSession().setAttribute("login", 1);
        response.sendRedirect("homepage.jsp");
      } else if (data[0] == 0) {
        urlForward = ERROR_MSG_ACCESS;
        response.sendRedirect(request.getContextPath() + urlForward);
      } else {
        urlForward = ERROR_SALAH_PASWORD;
        response.sendRedirect(request.getContextPath() + urlForward);
      }
    }
  }
}

```

Gambar 3.3.1.2 Source code servlet login

Modul *login* memiliki level pengguna, untuk itu digunakan sintaks *Servlet* untuk pengecekan status user pengguna. Berikut adalah potongan *source code*-nya.

```

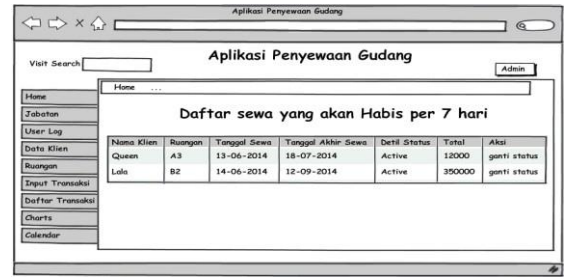
public int[] login(String username, String password) throws SQLException {
  int idlevel = -1;
  int status = -1;
  int iduser = -1;
  PreparedStatement p = conn.prepareStatement("select idlevel, status, iduser from user where username = ? and password = ?");
  p.setString(1, username);
  p.setString(2, password);
  ResultSet rs = p.executeQuery();
  if (rs.next()) {
    idlevel = rs.getInt("idlevel");
    status = rs.getInt("status");
    iduser = rs.getInt("iduser");
    if (idlevel == 3) {
      System.out.println("anda adalah mahasiswa");
    } else if (idlevel == 2) {
      System.out.println("anda adalah dosen");
    } else if (idlevel == 1) {
      System.out.println("anda adalah admin");
    } else {
      System.out.println("username atau password yang anda masukan salah");
    }
    System.out.println("ID LEVEL : " + idlevel);
    return new int[]{idlevel, status, iduser};
  }
}

```

Gambar 3.3.1.4 Source code servlet hak akses

**3.3.2 Modul Home**

Modul ini diperuntukan untuk semua petugas. Modul ini akan muncul setelah petugas *login*. Pada modul ini petugas dapat melihat daftar masa sewa yang akan habis kurang dari 7 hari. Petugas juga dapat mengubah status apabila masa sewa telah lewat masa habis. Berikut adalah sketsa tampilan bagian *home* setelah berhasil *login*.



Gambar 3.3.2.1 Desain dashboard

Berikut adalah *source code* jsp untuk halaman *dashboard*.

```

<div class="row">
  <div class="col">
    <table border="1">
      <caption>Daftar sewa yang akan Habis</caption>
      <thead>
        <tr>
          <th>Nama Klien</th>
          <th>Ruangan</th>
          <th>Tanggal Awal Sewa</th>
          <th>Tanggal Akhir Sewa</th>
          <th>Detil Status</th>
          <th>Total</th>
          <th>Aksi</th>
        </tr>
      </thead>
      <tbody>
        <tr>
          <td>Queen</td>
          <td>A3</td>
          <td>13-06-2014</td>
          <td>18-07-2014</td>
          <td>Active</td>
          <td>12000</td>
          <td>ganti status</td>
        </tr>
        <tr>
          <td>Lola</td>
          <td>B2</td>
          <td>14-06-2014</td>
          <td>12-09-2014</td>
          <td>Active</td>
          <td>30000</td>
          <td>ganti status</td>
        </tr>
      </tbody>
    </table>
  </div>

```

Gambar 3.3.2.2 Source code jsp dashboard

Untuk fungsionalitas dari daftar sewa yang akan habis, digunakan *servlet*. Berikut adalah *source code servlet* untuk daftar sewa yang akan habis.

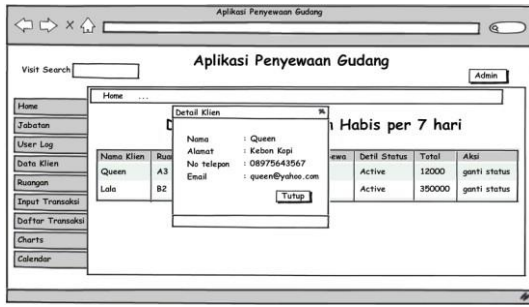
```

private String transaksi_daftar_transaksi(String id_transaksi) {
  String sql = "SELECT detil_transaksi.iddetil_transaksi, detil_transaksi.tanggalakhir, transaksi.id_transaksi, transaksi.tanggalawal, transaksi.tanggalakhir, ruang.idruang, ruang.ruang, detil_transaksi.id_transaksi, detil_transaksi.tanggalakhir, detil_transaksi.klien, klien.nama, detil_transaksi.harga FROM detil_transaksi, transaksi, ruang, detil_transaksi, klien WHERE transaksi.id_transaksi = detil_transaksi.id_transaksi AND ruang.idruang = detil_transaksi.idruang AND detil_transaksi.iddetil_transaksi = detil_transaksi.iddetil_transaksi AND klien.idklien = transaksi.idklien AND transaksi.id_transaksi = id_transaksi";
  ResultSet rs;
  try {
    rs = new ConnectionBean().peek(query);
    while (rs.next()) {
      val = "<tr><tr>\n"
        + "<td><input type='button' value='"+ rs.getString("nama") + "'\n"
        + "onclick='detailKlien()' + rs.getString("idklien") + "' class='btn btn-info'></td>\n"
        + "<td>"+ rs.getString("ruang") + "</td>\n"
        + "<td>"+ rs.getString("tanggalawal") + "</td>\n"
        + "<td>"+ rs.getString("tanggalakhir") + "</td>\n"
        + "<td>"+ rs.getString("detil_status") + "</td>\n"
        + "<td>"+ rs.getString("keterangan") + "</td>\n"
        + "<td>"+ rs.getString("harga") + "</td>\n"
        + "<td><input type='button' value='Ganti Status' onclick='gantiStatusTransaksi(' +\n"
        + "rs.getString("idDetil_transaksi") + ")' class='btn btn-info'></td>\n"
        + "</tr></tr>";
    }
  } catch (SQLException e) {
    System.out.println("Error in detil_transaksi_daftar_transaksi : " + e.getMessage());
  }
  return val;
}

```

Gambar 3.3.2.3 Source code servlet Masa Sewa yang akan habis

Pada tabel daftar masa sewa, user dapat melihat data klien yang berfungsi jika masa sewa akan habis, user bisa menghubungi klien dengan melihat detail klien tersebut. Berikut adalah tampilan sketsa *popup* detail klien.



Gambar 3.3.2.2 Desain popup detail klien

Kemudian berikut adalah *source code servlet* untuk popup detail klien.

```
private String klien_detailKlien(String idKlien) {
    String val = "";
    String query = "SELECT * FROM 'klien' WHERE 'idklien' = " + idKlien + "";
    ResultSet rs;
    try {
        rs = connection.peek(query);
        if (rs.next()) {
            val = "<form>\n"
                + "<table border='0' cellpadding='0' cellspacing='0'>\n"
                + "<tr>\n"
                + "<td colspan='2'>Nama Klien</td>\n"
                + "<td colspan='2'>" + rs.getString(2) + "</td>\n"
                + "</tr>\n"
                + "<tr>\n"
                + "<td colspan='2'>Alamat</td>\n"
                + "<td colspan='2'>" + rs.getString(3) + "</td>\n"
                + "</tr>\n"
                + "<tr>\n"
                + "<td colspan='2'>No telepon</td>\n"
                + "<td colspan='2'>" + rs.getString(4) + "</td>\n"
                + "</tr>\n"
                + "<tr>\n"
                + "<td colspan='2'>Email</td>\n"
                + "<td colspan='2'>" + rs.getString(5) + "</td>\n"
                + "</tr>\n"
                + "</table>\n"
                + "</form>";
        }
    } catch (SQLException e) {
        System.out.println("Error in klien_detailKlien: " + e.getMessage());
    }
    return val;
}
```

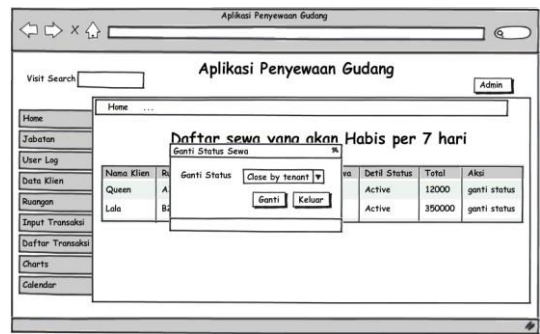
Gambar 3.3.2.5 Source code servlet popup detail klien

Untuk memanggil popup agar tampil di halaman, dibutuhkan *code javascript*. Berikut adalah *source code javascript* untuk popup detail klien.

```
$('#detail-klien').dialog({
    autoOpen: false,
    height: 360,
    width: 360,
    modal: true,
    resizable: false,
    buttons: {
        Tutup: function() {
            $(this).dialog("close");
        }
    }
});
function detailKlien(e) {
    jQuery(function($) {
        var data = {
            action: "detailKlien",
            idKlien: e
        };
        $.ajax({
            url: "transaksi",
            type: "POST",
            data: data,
            success: function(data) {
                $('#detail-klien').html(data);
                $('#detail-klien').dialog("open");
            }
        });
    });
}
```

Gambar 3.3.2.6 Source code javascript popup detail klien

Pada halaman ini juga, user dapat mengganti status masa sewa dari aktif menjadi *close* (tutup). Jika klien tidak konfirmasi tentang penutupan ruang gudang maka pemilik gudang yang akan mengubah status gudang tersebut (*close by landlord*). Dan jika klien konfirmasi maka klien yang akan mengubah status gudang tersebut (*close by tenant*). Berikut tampilan sketsa popup ganti status sewa ruang gudang.



Gambar 3.3.2.3 Desain popup ganti status sewa

Berikut *source code jsp* untuk popup ganti status sewa.

```
<div id="detail-form" title="Ganti Status">
<form
<table border="0" cellpadding="0" cellspacing="0">
<tr>
<td colspan="2">Ganti Status</td>
</tr>
<tr>
<td colspan="2">
<select id="detail-comboStatusSewa">
<option value="">
<script>
<!--
<!--
</script>
</td>
</tr>
<tr>
<td colspan="2">
<input type="button" value="Keluar" />
<input type="button" value="Ganti" />
</td>
</tr>
</table>
</div>
```

Gambar 3.3.2.8 Source code jsp popup ganti status sewa

Untuk query ganti status sewa digunakan *servlet*. Berikut adalah *source code servlet* untuk ganti status sewa.

```
private void detailTransaksi_updateDetailTransaksi(String idDetailStatus, String idDetailTransaksi) {
    String idRuang = "";
    String ambilIdRuang = "SELECT 'idRuang' FROM 'detailTransaksi' WHERE 'idDetailTransaksi' = " + idDetailTransaksi + "";
    ResultSet rs;
    try {
        rs = connection.peek(ambilIdRuang);
        if (rs.next()) {
            idRuang = rs.getString(1);
        }
        System.out.println("idRuang: " + idRuang);
        String query = "UPDATE 'detailTransaksi' SET 'idDetailStatus' = " + idDetailStatus + " WHERE 'idDetailTransaksi' = " + idDetailTransaksi + "";
        connection.poke(query);
        String updateRuang = "UPDATE 'ruang' SET 'status' = '0' WHERE 'idRuang' = " + idRuang + "";
        connection.poke(updateRuang);
        rs.close();
    } catch (SQLException e) {
        System.out.println("Error in detailTransaksi_updateDetailTransaksi: " + e.getMessage());
    }
}
```

Gambar 3.3.2.9 Source code servlet ganti status masa sewa

Fungsionalitas untuk ganti status masa sewa menggunakan *javascript*. Berikut *source code javascript* untuk ganti status masa sewa.

```
$('#detail-form').dialog({
    autoOpen: false,
    height: 200,
    width: 420,
    modal: true,
    resizable: false,
    buttons: {
        Ganti: function() {
            var data = {
                action: "updateDetailTransaksi",
                idDetailStatus: $('#detail-comboStatusSewa').val(),
                idDetailTransaksi: idDetailTransaksi
            };
            $.ajax({
                url: "transaksi",
                type: "POST",
                data: data,
                success: function() {
                    listDefran();
                    $('#detail-form').dialog("close");
                }
            });
        },
        Batal: function() {
            $(this).dialog("close");
        }
    }
});
function gantiStatusTransaksi(e) {
    idDetailTransaksi = e;
    $('#detail-form').dialog("open");
}
```

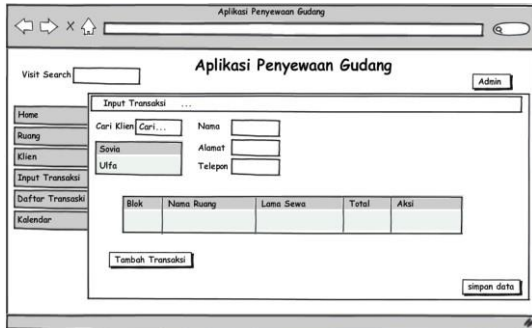
Gambar 3.3.2.10 Source code javascript ganti status masa sewa

### 3.3.3 Modul Transaksi

Modul ini diperuntukkan khusus untuk bagian transaksi. Modul transaksi memiliki beberapa modul yaitu, modul *input* transaksi, modul *daftar* transaksi dan modul *detail* transaksi. Petugas yang dapat mengakses modul ini hanya pegawai, sedangkan *admin* dan *superadmin* hanya bisa mengakses modul *daftar* transaksi dan *detail* transaksi.

3.3.3.1 Sub-Modul *Input* Transaksi

Modul ini hanya bisa diakses oleh pegawai. Modul ini digunakan untuk menambah transaksi klien yang sudah terdaftar. Berikut sketsa tampilan halaman *input* transaksi setelah berhasil *login*.



Gambar 3.3.3.1.1 Desain Tambah Transaksi

Berikut adalah beberapa potongan *source code* JSP untuk menampilkan data klien dengan menggunakan kotak pencarian data klien.

```
<div>
<ul class="breadcrumb">
<li>
<a href="input_transaksi.jsp" >Input Transaksi</a>
</li>
</ul>
</div>
<div style="float: left; width: 400px; margin-bottom: 30px">
<input type="hidden" value="" >
<input type="text" value="" id="nama" style="height: 30px; width: 140px;"/>
</div>
<div style="float: left; width: 400px; margin-bottom: 30px">
<input type="text" value="" id="alamat" style="height: 30px; width: 140px;"/>
</div>
<div style="float: left; width: 400px; margin-bottom: 30px">
<input type="text" value="" id="telepon" style="height: 30px; width: 140px;"/>
</div>
</div>
```

Gambar 3.3.3.1.2 Source code jsp cari data klien

*Query* dari cari data klien berada di servlet. Berikut *source code* servlet dari cari data klien.

```
private String klien_cariKlien(String nama) {
String sql = "SELECT * FROM 'klien' WHERE 'nama' LIKE '%" + nama + "%'";
ResultSet rs;
try {
rs = connection.prepareStatement(sql).executeQuery();
while (rs.next()) {
val = "<option value='" + rs.getString(1) + "'>" + rs.getString(2) + "</option>";
}
rs.close();
} catch (SQLException e) {
System.out.println("Error in klien_cariKlien: " + e.getMessage());
}
return val;
}
else if (action.equals("ambilDataKlien")) {
String idKlien = request.getParameter("idKlien");
output = klien_ambilDataKlien(idKlien);
}
```

Gambar 3.3.3.1.2 Source code servlet cari data klien

Fungsi dari cari data klien diterapkan pada fungsi *jquery*. Berikut *source code* *jquery* untuk cari data klien.

```
$( "#cari" ).keyup(function() {
if ( $(this).val() != "" ) {
var data = { action: "cariKlien", nama: $(this).val() };
$.ajax({
url: "transaksi",
data: data,
type: "POST",
success: function(data) {
$( "#hasilPencarian" ).html(data);
}
}
);
} else {
$( "#hasilPencarian" ).html("");
}
});
$( "#hasilPencarian" ).click(function() {
var id = $( "#hasilPencarian" ).val();
var data = { action: "ambilDataKlien", idKlien: id };
$.ajax({
url: "transaksi",
data: data,
type: "POST",
success: function(data) {
var values = data.split("");
$( "#hasilPencarian" ).val( $( "#hasilPencarian" ).val() );
$( "#nama" ).val( values[0] );
$( "#alamat" ).val( values[1] );
$( "#telepon" ).val( values[2] );
checkButtonSiapan();
}
}
});
});
```

Gambar 3.3.3.1.4 Source code *jquery* cari data klien

Setelah user mencari data klien. User dapat menambahkan transaksi dengan *code* *jsp* seperti berikut.

```
<td style="width: 150px; text-align: center;">
<input type="text" value="" id="nama" style="height: 30px; width: 140px;"/>
</td>
<td style="width: 150px; text-align: center;">
<input type="text" value="" id="alamat" style="height: 30px; width: 140px;"/>
</td>
<td style="width: 150px; text-align: center;">
<input type="text" value="" id="telepon" style="height: 30px; width: 140px;"/>
</td>
</tr>
</table>
</div>
```

Gambar 3.3.3.1.5 Source code *jsp* tambah transaksi

Fungsi dari *query* tambah transaksi berada di servlet. Berikut *source code* servlet dari tambah transaksi.

```
private String transaksi_insertTransaksi(HttpServletRequest request, HttpServletResponse response) {
String idTransaksi = "";
try {
java.util.Date tanggalMulai = new java.util.Date();
//date transaksi
java.sql.Date tanggalSelesai = new java.sql.Date(dateSelesai.getTime());
String idKlien = request.getParameter("idKlien");
String idRuang = request.getParameter("idRuang");
String totalHarga = request.getParameter("totalHarga");
//detail transaksi
String idRuang = request.getParameter("idRuang");
java.util.Date dateSelesai = new java.util.Date();
String tipeSewa = request.getParameter("tipeSewa");
int lamaSewa = Integer.parseInt(request.getParameter("lamaSewa"));
if (tipeSewa.equals("hari")) {
dateSelesai.setDate(dateSelesai.getDate() + lamaSewa);
} else if (tipeSewa.equals("minggu")) {
dateSelesai.setDate(dateSelesai.getDate() + (lamaSewa * 7));
} else if (tipeSewa.equals("bulan")) {
dateSelesai.setMonth(dateSelesai.getMonth() + lamaSewa);
} else if (tipeSewa.equals("tahun")) {
dateSelesai.setYear(dateSelesai.getYear() + lamaSewa);
}
java.sql.Date tanggalSelesai = new java.sql.Date(dateSelesai.getTime());
idTransaksi = transaksi_getPKTransaksi(tanggalMulai, idKlien, idRuang);
if (idTransaksi.isEmpty()) {
String transaksi = "INSERT INTO 'transaksi' VALUES ('0','" + tanggalMulai + "','" + idKlien + "','" + idRuang + "','" + tanggalSelesai + "','" + totalHarga + "')";
connection.prepareStatement(transaksi);
} else {
idTransaksi = transaksi_getPKTransaksi(tanggalMulai, idKlien, idRuang);
if (idTransaksi.isEmpty()) {
String detailTransaksi = "INSERT INTO 'detailtransaksi' VALUES ('0','" + idTransaksi + "','" + idRuang + "','" + tanggalSelesai + "','" + totalHarga + "')";
connection.prepareStatement(detailTransaksi);
} else {
String detailTransaksi = "INSERT INTO 'detailtransaksi' VALUES ('0','" + idTransaksi + "','" + idRuang + "','" + tanggalSelesai + "','" + totalHarga + "')";
connection.prepareStatement(detailTransaksi);
}
}
String updateRuang = "UPDATE 'ruang' SET 'status'='1' WHERE 'idRuang'='" + idRuang + "'";
connection.prepareStatement(updateRuang);
String updateHarga = "UPDATE 'transaksi' SET 'totalHarga'='" + harga + "' WHERE 'idTransaksi'='" + idTransaksi + "'";
connection.prepareStatement(updateHarga);
} catch (SQLException e) {
System.out.println("Error in transaksi_insertTransaksi: " + e.getMessage());
}
```

Gambar 3.3.3.1.6 Source code servlet tambah transaksi

Seluruh fungsionalitas tambah transaksi menggunakan *jquery*. Berikut *source code* *jquery* untuk tambah transaksi.

```

$( "#dialog-form" ).dialog({
  autoOpen: false,
  height: 480,
  width: 480,
  modal: true,
  resizable: false,
  buttons: {
    Simpan: function() {
      var flag1 = $('#comboBlok')[0].selectedIndex > 0;
      var flag2 = $('#comboTipeSewa')[0].selectedIndex > 0;
      if (flag1 && flag2) {
        var blok = $('#comboBlok').val().toString().split("-");
        var ruang = $('#comboRuang').val().toString().split("-");
        var durasiSewa = $('#lamaSewa').val();
        var tipeSewa = $('#comboTipeSewa').val().toString();
        var lamaSewa = durasiSewa * " " + tipeSewa;
        var totalHarga = $('#totalHarga').val();
      }

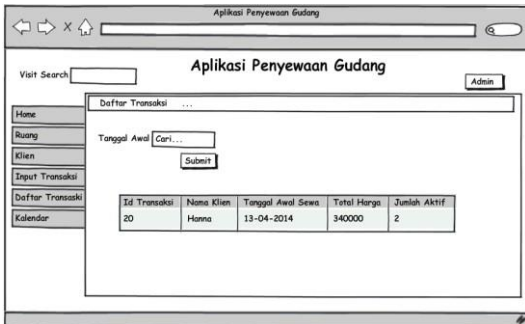
      if (operasi == "insert") {
        selectedRuang = $('#comboRuang').val() + " ";
        $.row.add({
          "input type='hidden' id='idBlok'" + counter + "' value='" + blok[0] + "'" +
            "span id='blok'" + counter + "' *'" + blok[1] + "' /span",
          "input type='hidden' id='idRuang'" + counter + "' value='" + ruang[0] + "'" +
            "span id='ruang'" + counter + "' *'" + ruang[1] + "' /span",
          "span id='lamaSewa'" + counter + "' *'" + lamaSewa + "' /span",
          "span id='totalHarga'" + counter + "' *'" + totalHarga + "' /span",
          "input type='button' value='Ganti' onclick='gantiDetailTransaksi(" + counter + ")'"
        });
        counter++;
        checkButtonSimpan();
      } else {
        $('#idBlok' + selected).val(blok[0]);
        $('#blok' + selected).text(blok[1]);
        $('#idRuang' + selected).val(ruang[0]);
        $('#ruang' + selected).text(ruang[1]);
        $('#lamaSewa' + selected).text(lamaSewa);
        $('#totalHarga' + selected).text(totalHarga);
      }

      $(this).dialog("close");
    } else {
      $('#warning').text('Mohon isi semua data');
    }
  },
  cancel: function() {
    $(this).dialog("close");
  }
});
    
```

Gambar 3.3.3.1.7 Source code jquery tambah transaksi

### 3.3.3.2 Sub-Modul Daftar Transaksi

Modul ini dapat diakses oleh *admin*, *superadmin* dan pegawai. Modul ini berfungsi untuk melihat data daftar transaksi. Berikut sketsa tampilan modul daftar transaksi setelah berhasil *login*.



Gambar 3.3.3.2.1 Desain daftar transaksi

Pada halaman daftar transaksi user dapat memilih daftar transaksi berdasarkan tanggal. Berikut *source code jsp* untuk pencarian tanggal daftar transaksi.

```

<div style="float: left; width: 480px; margin-bottom: 80px;">
  <form method="post" action="daftar_transaksi.jsp">
    <table>
      <tr>
        <td width="125px">Tanggal Awal /td>
        <td>
          <input type="text" name="tanggal" class="datepicker" style="height:30px;"/>
        </td>
      </tr>
      <tr>
        <td colspan="2" align="right">
          <input type="submit" value="Submit" class="btn btn-info"/>
        </td>
      </tr>
    </table>
  </div>
    
```

Gambar 3.3.3.2.2 Source code jsp pencarian tanggal daftar transaksi

Setelah user mencari tanggal, daftar transaksi akan muncul pada tabel. Berikut *source code jsp* untuk daftar transaksi.

```

<form method="post" action="inform_detail_transaksi.jsp">
  <table id="example" class="table table-striped table-bordered bootstrap-datable datatables">
    <thead>
      <tr>
        <th>ID Transaksi</th>
        <th>Nama Klien</th>
        <th>Tanggal Awal Sewa</th>
        <th>Total Harga</th>
        <th>Jumlah Aktif</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <td>
          <input type="text" value="" />
          <input type="submit" value="Print" class="btn btn-info" />
        </td>
      </tr>
    </tbody>
  </table>
</form>
    
```

Gambar 3.3.3.2.3 Source code jsp daftar transaksi

Fungsi dari tabel daftar transaksi diterapkan di javascript dan jquery. Berikut *source code javascript* daftar transaksi.

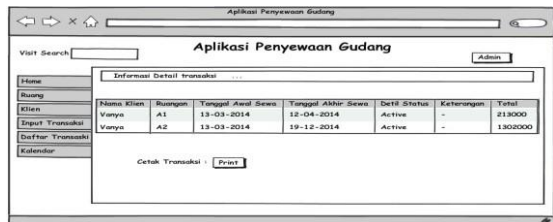
```

private String transaksi_daftarTransaksi(String idTransaksi) {
  String sql = "SELECT detail_transaksi.id_transaksi, detail_transaksi.tanggal_lahir, transaksi.id_transaksi, transaksi.total_harga, transaksi.tanggal_mulai, ruang.id_ruang, ruang.ruang, detail_transaksi.id_transaksi, detail_transaksi.detail_transaksi, detail_transaksi.detail_transaksi, klien.id_klien, klien.nama, detail_transaksi.harga FROM detail_transaksi, transaksi, ruang, detail_transaksi, klien WHERE transaksi.id_transaksi = detail_transaksi.id_transaksi AND ruang.id_ruang = detail_transaksi.id_ruang AND detail_transaksi.id_transaksi = detail_transaksi.id_transaksi AND klien.id_klien = transaksi.id_klien AND transaksi.id_transaksi = id_transaksi ORDER BY detail_transaksi.id_transaksi";
  ResultSet rs;
  try {
    rs = new ConnectionBean().peek(query);
    while (rs.next()) {
      val = "<tr>\n"
        + "<td><input type='button' value=''" + rs.getString("nama") + "'\n"
        + "onclick='detailKlien(" + rs.getString("id_klien") + ")'\n"
        + "</td>\n"
        + "<td>" + rs.getString("tanggal_mulai") + "</td>\n"
        + "<td>" + rs.getString("tanggal_lahir") + "</td>\n"
        + "<td>" + rs.getString("detail_transaksi") + "</td>\n"
        + "<td>" + rs.getString("keterangan") + "</td>\n"
        + "<td>" + rs.getString("tanggal") + "</td>\n"
        + "<td><input type='button' value='Ganti Status' onclick='gantiStatusTransaksi(" + rs.getString("id_transaksi") + ")'\n"
        + "class='btn btn-info'></td>\n"
        + "</tr>";
    }
  } catch (SQLException e) {
    System.out.println("Error in detail_transaksi_daftar_transaksi: " + e.getMessage());
  }
  return val;
}
    
```

Gambar 3.3.3.2.4 Source code javascript daftar transaksi

### 3.3.3.3 Sub-Modul Detail Transaksi

Modul ini dapat diakses oleh *admin*, *superadmin* dan pegawai. Modul ini digunakan untuk melihat data detail transaksi berdasarkan id transaksi yang sebelumnya dipilih pada halaman daftar transaksi. Berikut sketsa tampilan halaman detail transaksi setelah berhasil *login*.



Gambar 3.3.3.3.1 Desain Informasi detail transaksi

Berikut adalah *source code jsp* untuk tabel detail transaksi.

```

<div class="box-content">
  <input type="hidden" value="" />
  <table class="table table-striped table-bordered bootstrap-datable datatables" id="example">
    <thead>
      <tr>
        <th>Nama Klien</th>
        <th>Ruang</th>
        <th>Tanggal Awal Sewa</th>
        <th>Tanggal Akhir Sewa</th>
        <th>Detail Status</th>
        <th>Keterangan</th>
        <th>Total</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <td colspan="7">
          <input type="submit" value="Print" class="btn btn-info" />
        </td>
      </tr>
    </tbody>
  </table>
</div>
    
```

Gambar 3.3.3.3.2 Source code jsp detail transaksi



Fungsi query detail transaksi ada di servlet. Berikut *source code* servlet untuk detail transaksi.

```
private String transaksi_daftar_transaksi(String id_transaksi) {
    String val = "";
    String query = "SELECT detail_transaksi.id_detail_transaksi, detail_transaksi.tanggal_akhir,
    transaksi.id_transaksi, transaksi.total_harga, transaksi.tanggal_mulai, ruang.id_ruang,
    detail_transaksi.id_detail_transaksi, detail_transaksi.detail_transaksi, detail_transaksi.keterangan, klien.id_klien, klien.nama,
    detail_transaksi.harga FROM detail_transaksi, transaksi, ruang, detail_transaksi, klien WHERE transaksi.id_transaksi =
    detail_transaksi.id_transaksi AND ruang.id_ruang = detail_transaksi.id_ruang AND detail_transaksi.id_detail_transaksi =
    detail_transaksi.id_detail_transaksi AND klien.id_klien = transaksi.id_klien AND transaksi.id_transaksi = " + id_transaksi
    + " ORDER BY detail_transaksi.id_detail_transaksi";
    ResultSet rs;
    try {
        rs = new ConnectionBean().peek(query);
        while (rs.next()) {
            val += "\n";
            + "<td><input type='button' value='Ganti Status' onclick='gantistatus_transaksi(\" +
            + rs.getString("id_detail_transaksi") + "\")' class='btn btn-info'></td>\n";
            + "<td><input type='button' value='Ganti Status' onclick='gantistatus_transaksi(\" +
            + rs.getString("id_detail_transaksi") + "\")' class='btn btn-info'></td>\n";
        }
    } catch (SQLException e) {
        System.out.println("Error in detail_transaksi_daftar_transaksi: " + e.getMessage());
    }
    return val;
}
```

Gambar 3.3.3.3.3 Source code servlet detail transaksi

Fungsionalitas dari detail transaksi diterapkan pada *javascript* dan *jquery*. Berikut adalah *source code javascript* untuk detail transaksi.

```
$( "#detail-form" ).dialog({
    autoOpen: false,
    height: 200,
    width: 420,
    modal: true,
    resizable: false,
    buttons: {
        ganti: function() {
            var data = {
                action: "updateDetail_transaksi",
                id_detail_transaksi: $( "#detail-combostatussewa" ).val(),
                id_transaksi: id_transaksi
            };
            $.ajax({
                url: "transaksi",
                type: "POST",
                data: data,
                success: function() {
                    listDaftar();
                    $( "#detail-form" ).dialog("close");
                }
            });
        },
        batal: function() {
            $( this ).dialog("close");
        }
    }
});
function listDaftar() {
    $.ajax({
        url: "transaksi",
        type: "POST",
        success: function(data) {
            $( "#tabel-detail" ).html(data);
        }
    });
}
```

Gambar 3.3.3.3.4 Source code javascript detail transaksi

Pada detail transaksi ini terdapat fungsi untuk melihat detail klien dan ganti status sewa seperti pada halaman *dashboard*. Untuk penjelasannya sama seperti halaman *dashboard*.

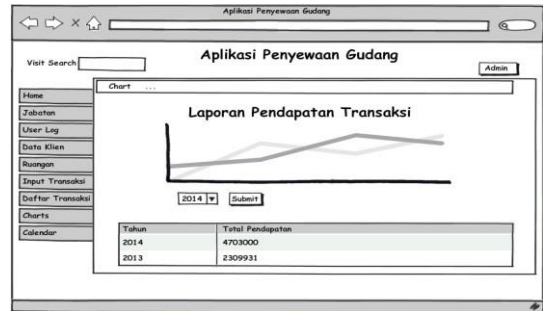
### 3.3.4 Modul Chart dan Laporan Transaksi

Pada modul ini digunakan untuk pengecekan laporan data keuangan maupun laporan data ruangan. Modul laporan transaksi memiliki beberapa sub-modul yaitu, *chart* dan tabel daftar transaksi, *chart* dan tabel daftar transaksi berdasarkan rentang tanggal, *chart* dan tabel daftar ruangan dan informasi ketersediaan ruangan. Admin dan petugas yang dapat mengakses sub-modul *chart* dan tabel daftar ruangan dan informasi ketersediaan ruangan, sedangkan superadmin hanya bisa mengakses sub-modul *chart* dan tabel daftar transaksi dan *chart* dan tabel daftar transaksi berdasarkan rentang tanggal.

#### 3.3.4.1 Sub-modul Chart dan Tabel Daftar Transaksi

Sub-modul ini hanya dapat diakses oleh *superadmin*. Sub-modul ini digunakan untuk melihat laporan keuangan untuk setiap transaksi yang dilakukan, baik secara perbulan, pertahun maupun laporan transaksi untuk semua tahun. Berikut sketsa tampilan halaman *chart* dan tabel daftar

ransaksi.



Gambar 3.3.4.1.1 Desain chart dan laporan transaksi

Berikut *source code javascript* untuk *chart* transaksi.

```
<script type="text/javascript">
$(document).ready(function() {
    $( "#container" ).highcharts({
        chart: { type: 'spline' },
        title: { text: 'Laporan Pendapatan Transaksi' },
        subtitle: { text: 'Penyewaan Gudang' },
        xAxis: { categories: ['<out.printIn(arr.toISOString()); %>'],
            title: { text: 'Total Pendapatan (Rp)'} },
        tooltip: { crosshairs: true, shared: true },
        plotOptions: {
            spline: {
                marker: { radius: 4, lineColor: '#666666', lineWidth: 1
            }
        },
        series: [{ name: 'Transaksi', data: ['<out.printIn(arr.toISOString()); %>']
        }
    }
});
</script>
```

Gambar 3.3.4.1.2 source code javascript chart transaksi

Berikut adalah beberapa potongan *source code* untuk sub-modul *chart* dan tabel daftar transaksi.

```
<body>
<div>
    String data = "SELECT sum( total_harga ) as total, year( tanggal_mulai ) as tahun FROM transaksi
    GROUP BY year( tanggal_mulai ) ASC;";
    Statement stmt = db.getConnection().createStatement();
    ResultSet rs = stmt.executeQuery(data);
    <div id="container" style="border: 1px solid #ccc; padding: 10px; margin: 0 auto;>
    <div class="box-content">
    <table class="table table-condensed" align="center">
    <thead>
    <tr>
    <th align="center">Tahun</th>
    <th align="center">Total Pendapatan</th>
    </tr>
    </thead>
    <tbody>
    <tr>
    <td align="center"><decFormat df = new DecimalFormat();
    DecFormatSymbol dfS = new DecimalFormatSymbol();
    df.setGroupingSeparator(",");
    df.setMonetaryDecimalSeparator(".");
    df.setDecimalFormatSymbols(dfS);
    ((DecimalFormat) df).setDecimalFormatSymbols(dfS);
    JSonArray arr = new JSonArray();
    JSonArray arr = new JSonArray();
    int i = 0;
    while (rs.next()) {
        arr.add(rs.getInt(1));
        arr.add(rs.getInt(2));
    }
    <tbody>
    <tr>
    <td align="center" width="10%"><rs.getInt(1)></td>
    <td align="center" width="90%"><rs.getInt(2)></td>
    </tr>
    </tbody>
    </table>
    </div>
    <FORM ACTION="ChartBulan.jsp" METHOD="POST" target = "_blank">
    <input type="button" value="Batal" />
    </div>
    <div>
    <table border="1">
    <thead>
    <tr>
    <th align="center" colspan="2">Transaksi Tahun</th>
    </tr>
    </thead>
    <tbody>
    <tr>
    <td align="center" colspan="2"><select name="tahun" id="tahun" width="100%">
    <option value="<rs.getInt('maxTahun'); %>"><rs.getInt('maxTahun')>
    </option>
    </select>
    </td>
    </tr>
    </tbody>
    </table>
    </div>
</body>
```

Gambar 3.3.4.1.3 Source code jsp chart dan laporan transaksi

Pada *code* tersebut akan melakukan *action form* ke halaman *ChartBulan.jsp*, berikut *code*-nya.

```
<%@page import="java.text.DecimalFormatSymbols"%>
<%@page import="java.text.DecimalFormat"%>
<%@page import="java.text.DecimalFormat"%>
<%@page import="org.json.simple.JSONArray"%>
<%@page import="org.json.simple.JSONObject"%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@page import="java.io.*"%>
<@connectionBean dc = new ConnectionBean();%>
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Aplikasi Penyewaan Gudang</title>
<meta name="author" content="Itis">
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.1/jquery.min.js"></script>
<script src="https://www.highcharts.com/highcharts.js"></script>
<script src="https://www.highcharts.com/highcharts/themes/darkblue.js"></script>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
</head>
<body>
<script type="text/javascript">
$(document).ready(function() {
    $( "#container" ).highcharts({
        chart: {
            type: 'spline'
        },
        title: {
            text: 'Laporan Pendapatan Transaksi'
        },
        subtitle: {
            text: 'Penyewaan Gudang'
        },
        xAxis: {
            categories: <Out.println(br.toString()); %>,
            labels: {
                formatter: function() {
                    var path = "<Out.println(br.toString()); %>";
                    this.value = "</ao";
                    return path;
                }
            },
            isHTML: true
        },
        yAxis: {
            title: {
                text: 'Total Pendapatan (Rp)'
            }
        },
        tooltip: {
            crosshairs: true,
            shared: true
        },
        plotOptions: {
            spline: {
                marker: {
                    radius: 4,
                    fillColor: "#666666",
                    lineWidth: 1
                }
            }
        },
        series: [ {
            name: 'transaksi',
            data: <Out.println(br.toString()); %>
        } ]
    });
    var s = $('#highcharts-xaxis-labels')[0].childNodes.length;
    for (k = 0; k < params(k); k++) {
        var y = $('#highcharts-xaxis-labels')[k].childNodes[0].childNodes[0].nodeValue;
        y.innerHTML = "<Out.println(br.toString()); %>";
        y.innerHTML = "<Out.println(br.toString()); %>";
        y.innerHTML = "<Out.println(br.toString()); %>";
    }
});
</script>
</body>
</html>
```

Gambar 3.3.4.1.4 Source code chart bulan

Pada code tersebut terhubung ke halaman ChartBulanDetil.jsp. Berikut source code-nya

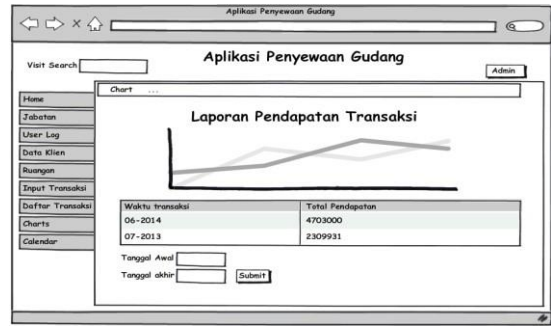
```
<%@page import="java.text.DecimalFormatSymbols"%>
<%@page import="java.text.DecimalFormat"%>
<%@page import="org.json.simple.JSONArray"%>
<%@page import="org.json.simple.JSONObject"%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@page import="java.io.*"%>
<@connectionBean dc = new ConnectionBean();%>
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Aplikasi Penyewaan Gudang</title>
<meta name="author" content="Itis">
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.1/jquery.min.js"></script>
<script src="https://www.highcharts.com/highcharts.js"></script>
<script src="https://www.highcharts.com/highcharts/themes/darkblue.js"></script>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
</head>
<body>
<script type="text/javascript">
<script type="text/javascript">
String tahun = request.getParameter("tahun");
String bulan = request.getParameter("bulan");
String data = "SELECT s.no_mhs, c.cruang, b.tanggalmulai, d.tanggalakhir FROM detiltransaksi a, transaksi b, ruang c, detilstatus d, klik e WHERE b.idtransaksi = a.idtransaksi AND c.idruang = a.idruang AND d.iddetilstatus = a.iddetilstatus AND e.idklik = b.idklik AND monthname(b.tanggalmulai) = (' + bulan + ') AND year(b.tanggalmulai) = (' + tahun + ')";
Statement state = dc.getConnection().createStatement();
ResultSet rs = state.executeQuery(data);
DecimalFormat df = new DecimalFormat();
df.setGroupingSeparator("");
df.setDecimalFormatSymbols(df);
df.setDecimalFormatSymbols(df);
<table border="1">
Nama Klien/Th
 Ruang/Th | Total Harga/Th | Tanggal Mulai/Th | Tanggal Akhir/Th |


```

Gambar 3.3.4.1.5 Source code chart bulan detail

### 3.3.4.2 Sub-modul Chart dan Tabel Daftar Transaksi Berdasarkan Rentang Tanggal

Sub-modul ini hanya dapat diakses oleh superadmin. Sub-modul ini digunakan untuk melihat laporan keuangan untuk setiap transaksi yang dilakukan berdasarkan rentang tanggal yang diinginkan. Namun pada sub-modul ini superadmin hanya dapat melihat diagram laporan keuangan dengan maksimal rentang tanggal selama dua tahun. Berikut sketsa tampilan halaman chart dan tabel daftar transaksi berdasarkan rentang tanggal.



Gambar 3.3.4.2.1 Desain chart berdasarkan rentang tanggal

Berikut adalah beberapa potongan source code untuk sub-modul chart dan tabel daftar transaksi berdasarkan rentang tanggal.

```
<script type="text/javascript">
String data = "";
if (request.getParameter("tanggalPertama") == null && request.getParameter("tanggalKedua") == null) {
    data = "SELECT sum( totalHarga ), monthname( tanggalMulai ) FROM transaksi where year( tanggalMulai ) = (select max( year( tanggalMulai ) ) from transaksi ) GROUP BY monthname( tanggalMulai ) DESC";
} else {
    String tanggalPertama = request.getParameter("tanggalPertama");
    String tanggalKedua = request.getParameter("tanggalKedua");
    data = "SELECT sum( totalHarga ), DATE_FORMAT( tanggalMulai, 'MM-%Y' ) AS tahun, year( tanggalMulai ), month( tanggalMulai ) FROM transaksi where tanggalMulai between (' + tanggalPertama + ') and (' + tanggalKedua + ') GROUP BY year( tanggalMulai ), month( tanggalMulai )";
    Statement state = dc.getConnection().createStatement();
    ResultSet rs = state.executeQuery(data);
<div id="container" style="min-width: 185px; height: 200px; margin: 0 auto"></div>
<div class="box-content">
<table border="1" style="width: 100%; border-collapse: collapse; text-align: center;">

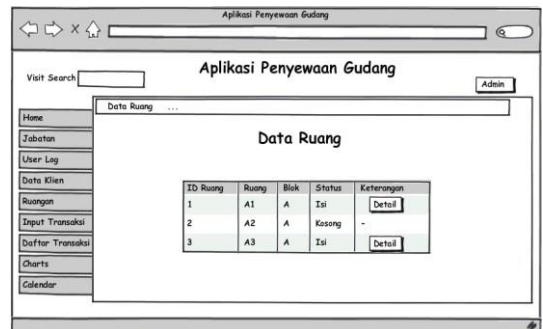



```

Gambar 3.3.4.2.2 Source code jsp chart rentang tanggal

### 3.3.4.3 Sub-modul Chart dan Tabel Daftar Ruangan

Sub-modul ini dapat diakses oleh admin dan pegawai. Sub-modul ini digunakan untuk melihat diagram dari jumlah ruangan yang isi maupun yang kosong beserta detail ruangan yang isi dan ruangan yang kosong serta kita dapat mencarinya berdasarkan tanggal yang ingin kita lihat. Berikut sketsa tampilan halaman chart dan tabel daftar ruangan.



Gambar 3.3.4.3.1 Desain chart dan tabel daftar ruangan

Ketika menekan tombol detail, maka akan muncul keterangan data penyewa yang menyewa ruangan tersebut. Berikut tampilan detail penyewa.



Gambar 3.3.4.3.2 desain detail klien chart

Berikut adalah beberapa potongan *source code* untuk sub-modul *chart* dan tabel daftar ruangan.

Pada *code* tersebut terlihat pada *button detail* berfungsi *onclick* ke *detailKlienRuangan* yang merupakan *pop-up* yang menggunakan *jquery* dan *javascript*, berikut *code*-nya.

```
function detailKlienRuangan(e) {
    jQuery(function($) {
        var data = {
            action: "detailKlienRuangan",
            idRuangan: e
        };
        $.ajax({
            url: "transaksi",
            type: "POST",
            data: data,
            success: function(data) {
                console.log(data);
                $("#detail-klien-ruang").html(data);
                $("#detail-klien-ruang").dialog("open");
            }
        });
    });
}
```

Gambar 3.3.4.3.2 source code javascript detail klien

Berikut adalah *code* untuk *popup*-nya

```
$("#detail-klien-ruang").dialog({
    autoOpen: false,
    height: 320,
    width: 320,
    modal: true,
    resizable: false,
    buttons: {
        Tutup: function() {
            $(this).dialog("close");
        }
    }
});
```

Gambar 3.3.4.3.3 Source code jquery popup detail klien

Berikut adalah *servlet* untuk mengambil data *detailKlienRuangan*-nya

```
div class="box-content">
<table class="table table-condensed align="center">
<thead>
<tr>
<th align="center">idblok</th>
<th align="center">ruang</th>
<th align="center">status</th>
<th align="center">keterangan</th>
</tr>
</thead>
<tbody><
String output = "";
String data = "SELECT r.idRuangan, r.ruangan, b.blok, r.status FROM ruang r JOIN blok
b on ( b.idblok = r.idblok )";
Statement state = dc.getConnection().createStatement();
ResultSet rs = state.executeQuery(data);
while (rs.next()) {
<tr>
<td class="center"><%=rs.getString("b.blok")%>/td>
<td class="center"><%=rs.getString("r.ruangan")%>/td>
<td class="center">
<%=rs.getString("r.status")%>
<%=rs.getString("r.keterangan")%>
</td>
</tr>
}
</tbody>
</table>
</div>
<div id="detail-klien-ruang"></div>
```

Gambar 3.3.4.3.4 Source code servlet popup detail klien

```
private String ruang_ambildataKlien(String idRuangan) {
String val = "";
String query = "SELECT k.nama, k.alamat, k.telepon, dt.tanggalakhir FROM klien k JOIN transaksi tr on
(tr.idklien = k.idklien) JOIN detailtransaksi dt on (dt.idtransaksi = tr.idtransaksi) JOIN ruang r on (r.idruang
= dt.idruang) WHERE dt.idruang = " + idRuangan + """;
System.out.println("SELECT k.nama, k.alamat, k.telepon, dt.tanggalakhir FROM klien k JOIN transaksi tr on
(tr.idklien = k.idklien) JOIN detailtransaksi dt on (dt.idtransaksi = tr.idtransaksi) JOIN ruang r on
(r.idruang = dt.idruang) WHERE dt.idruang = " + idRuangan + """);
ResultSet rs;
try {
rs = connection.peek(query);
if (rs.next()) {
val = "<form>";
val = "<table border='0' cellpadding='0' cellspacing='0'>";
val = "<tr>";
val = "<td colspan='3'>";
val = "<td colspan='3'>";
val = "<td colspan='3'>";
val = "<td colspan='3'>";
val = "<td colspan='3'>";
val = "<td colspan='3'>";
val = "<td colspan='3'>";
val = "<td colspan='3'>";
val = "<td colspan='3'>";
val = "<td colspan='3'>";
val = "<td colspan='3'>";
val = "<td colspan='3'>";
val = "<td colspan='3'>";
val = "<td colspan='3'>";
val = "</table>";
}
rs.close();
} catch (SQLException e) {
System.out.println("Error in ruang_ambildataKlien: " + e.getMessage());
}
return val;
} else if (action.equals("detailKlienRuangan")) {
String idRuangan = request.getParameter("idRuangan");
output = ruang_ambildataKlien(idRuangan);
}
```

Gambar 3.3.4.3.6 Source code jsp chart dan laporan daftar ruangan

3.3.4.4 Sub-modul Informasi Ketersediaan Ruangan

Sub-modul ini dapat diakses oleh *admin* dan pegawai. Sub-modul ini digunakan untuk melihat informasi ketersediaan ruangan yang isi maupun ruangan yang kosong dan kita dapat melihat detail pengguna yang sedang menggunakan ruang gudang tersebut. Berikut sketsa tampilan halaman informasi ketersediaan ruangan.



Gambar 3.3.4.4.1 Desain Informasi ketersediaan ruangan

Berikut adalah beberapa potongan *source code* untuk sub-modul *chart* dan tabel daftar ruangan.

```

<body><
String data = "";
String detail = "";
String tanggal = "";
String tanggalCari = "";
if (request.getParameter("tanggalCari") != null) {
    data = "select count(*) as kosong from ruang where idRuang not in (select idRuang from ruang join
    detailTransaksi using (idRuang))";
    detail = "select count(*) as kosong from ruang join detailTransaksi using (idRuang) join transaksi using
    (idTransaksi) where tanggalHulu = (' + tanggalCari + ') || tanggalLahir = (' + tanggalCari + ')";
    detail = "select count(*) as isi from ruang join detailTransaksi using (idRuang) join transaksi using
    (idTransaksi) where tanggalHulu = (' + tanggalCari + ') and tanggalLahir = (' + tanggalCari + ')";
    System.out.println("ISI");
} else {
    tanggalCari = request.getParameter("tanggalCari");
    data = "select count(*) as kosong from ruang where idRuang not in (select idRuang from ruang join
    detailTransaksi using (idRuang))";
    detail = "select count(*) as kosong from ruang join detailTransaksi using (idRuang) join transaksi using
    (idTransaksi) where tanggalHulu = (' + tanggalCari + ') || tanggalLahir = (' + tanggalCari + ')";
    detail = "select count(*) as isi from ruang join detailTransaksi using (idRuang) join transaksi using
    (idTransaksi) where tanggalHulu = (' + tanggalCari + ') and tanggalLahir = (' + tanggalCari + ')";
    System.out.println("ISI");
}
Statement state = db.getConnection().createStatement();
Statement state2 = db.getConnection().createStatement();
Statement state3 = db.getConnection().createStatement();
ResultSet rs = state.executeQuery(detail);
int i = 0;
int k = 0;
int total = 0;
while (rs.next()) {
    i = i + rs.getInt("kosong");
}
while (rs2.next()) {
    j = j + rs2.getInt("kosong");
    total = total + i + j;
}
while (rs3.next()) {
    k = k + rs3.getInt("isi");
}
%><
<table border="1">
<caption>Ketersediaan Ruang</caption>
<thead>
<tr>
<th>Kode Blok</th>
<th>Ruang</th>
<th>Pangsa</th>
<th>Lebar</th>
<th>Tinggi</th>
<th>Status</th>
<th>Keterangan</th>
<th>Aksi</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>A1</td>
<td>200</td>
<td>200</td>
<td>10</td>
<td>Kosong</td>
<td>-</td>
<td><input type="button" value="Detail" /></td>
</tr>
<tr>
<td>B</td>
<td>B1</td>
<td>100</td>
<td>100</td>
<td>10</td>
<td>Isi</td>
<td>-</td>
<td><input type="button" value="Detail" /></td>
</tr>
<tr>
<td>C</td>
<td>C1</td>
<td>250</td>
<td>100</td>
<td>10</td>
<td>Kosong</td>
<td>-</td>
<td><input type="button" value="Detail" /></td>
</tr>
</tbody>
</table>
</div>
</body>
</html>

```

Gambar 3.3.4.2 Source code jsp ketersediaan ruangan

Berikut adalah *code* untuk *chart pie*-nya

```

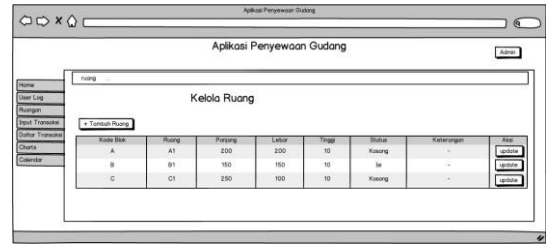
<script type="text/javascript">
function loadChart() {
    jQuery(function($) {
        $.fn.highcharts({
            chart: {
                plotBackgroundColor: null,
                plotBorderWidth: null,
                plotShadow: false
            },
            title: {
                text: 'Ketersediaan Ruang terhadap Penyewaan Gudang'
            },
            tooltip: {
                pointFormat: '{series.name}: {y}%'
            },
            plotOptions: {
                pie: {
                    allowPointSelect: true,
                    cursor: 'pointer',
                    dataLabels: {
                        enabled: false
                    },
                    showInLegend: true
                }
            },
            series: [
                {
                    type: 'pie',
                    name: 'Total Ruang',
                    data: [
                        ['Isi', <out>print(i)</out>],
                        ['Kosong', <out>print(j)</out>]
                    ]
                }
            ]
        });
    });
}
loadChart();
</script>
</body>
</html>

```

Gambar 3.3.4.3 Source code javascript chart pie

### 3.3.5 Modul Ruang

Modul ini hanya bisa diakses oleh *admin*. Modul ini digunakan untuk menambah ruangan baru dan mengubah ruangan yang sudah terdaftar. Berikut sketsa tampilan halaman Ruang setelah berhasil *login*.



Gambar 3.3.5.1 Desain Halaman Ruang

Beberapa Potongan *Source Code* Modul Ruang

```

<table cellpadding="0" cellspacing="0" border="0" class="table table-bordered table-striped datatable">
<thead>
<tr>
<th>Kode Blok</th>
<th>Ruang</th>
<th>Pangsa</th>
<th>Lebar</th>
<th>Tinggi</th>
<th>Status</th>
<th>Keterangan</th>
<th>Aksi</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>A1</td>
<td>200</td>
<td>200</td>
<td>10</td>
<td>Kosong</td>
<td>-</td>
<td><input type="button" value="Detail" /></td>
</tr>
<tr>
<td>B</td>
<td>B1</td>
<td>100</td>
<td>100</td>
<td>10</td>
<td>Isi</td>
<td>-</td>
<td><input type="button" value="Detail" /></td>
</tr>
<tr>
<td>C</td>
<td>C1</td>
<td>250</td>
<td>100</td>
<td>10</td>
<td>Kosong</td>
<td>-</td>
<td><input type="button" value="Detail" /></td>
</tr>
</tbody>
</table>
</div>
</body>
</html>

```

Gambar 3.3.4.1 Source code jsp ruangan

```

public class ruang {
    int idRuang;
    int idBlok;
    String ruang;
    int pangsa;
    int lebar;
    int tinggi;
    int status;
    int harga;
    String keterangan;
    public int getIdRuang() {
        return idRuang;
    }
    public void setIdRuang(int idRuang) {
        this.idRuang = idRuang;
    }
}

```

Gambar 3.3.4.2 source code dao Ruang

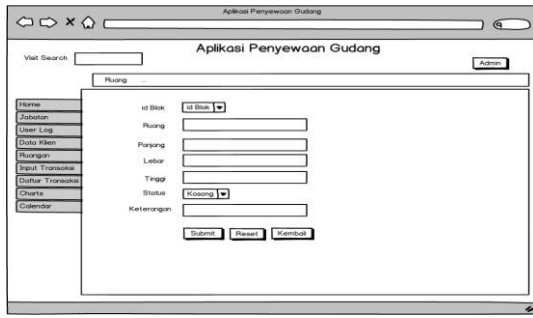
```

public List<ruang> getallRuang();
public boolean addRuang(ruang ruang);
public boolean updateRuang(ruang ruang);

```

Gambar 3.3.5.2 source code Model Ruang

### 3.3.5.1 Sub-Modul Input Ruangan



Gambar 3.3.5.1.1 Desain input ruangan

Potongan Source code-nya

```
if (request.getParameter("action").equals("add")) {
    ruang.setIDBlok(Integer.parseInt(request.getParameter("idblok")));
    ruang.setRuang(request.getParameter("ruang"));
    ruang.setPanjang(Integer.parseInt(request.getParameter("panjang")));
    ruang.setLebar(Integer.parseInt(request.getParameter("lebar")));
    ruang.setTinggi(Integer.parseInt(request.getParameter("tinggi")));
    ruang.setStatus(Integer.parseInt(request.getParameter("status")));
    ruang.setKeterangan(request.getParameter("keterangan"));
    log("ruangcontroller?action=add");
    try {
```

Gambar 3.3.5.1.4 source code insert ruangan pada servlet

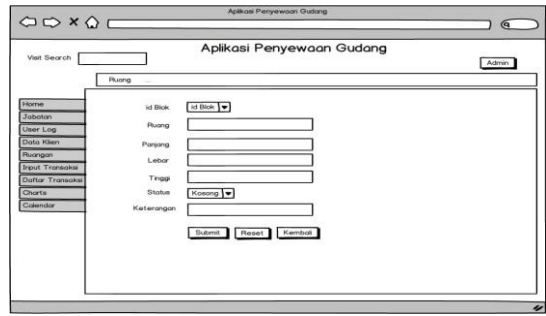
```
public boolean addRuang(Ruang ruang) {
    try {
        String query = "insert into ruang ("
            + "idblok, panjang, lebar, tinggi, status, keterangan)"
            + " values (?, ?, ?, ?, ?, ?)";
        PreparedStatement preparedStatement = (PreparedStatement) conn.prepareStatement(query);
        Statement statement = conn.prepareStatement(query);
        preparedStatement.setString(1, ruang.getIDBlok());
        preparedStatement.setString(2, ruang.getRuang());
        preparedStatement.setString(3, ruang.getPanjang());
        preparedStatement.setString(4, ruang.getLebar());
        preparedStatement.setString(5, ruang.getTinggi());
        preparedStatement.setString(6, ruang.getStatus());
        preparedStatement.setString(7, ruang.getKeterangan());
        int execute = preparedStatement.executeUpdate();
        System.out.println("execute sql " + query);
        System.out.println("Param : " + ruang.getIDBlok() + ", " + ruang.getRuang() + ", " +
            ruang.getRuang() + ", " + ruang.getPanjang() + ", " + ruang.getLebar() + ", " + ruang.getLebar() + ", " +
            ruang.getTinggi() + ", " + ruang.getStatus() + ", " + ruang.getKeterangan());
        if (execute == 1) {
            success = true;
        } catch (SQLException e) {
            e.printStackTrace();
        }
        return success;
    }
}
```

Gambar 3.3.5.1.3 source code insert ruangan pada ruangan dao imp

```
<form name="inputRuang" method="POST" action="">
<table>
<tr>
<td><input type="text" placeholder="ruang" required id="ruang" name="ruang" /></td>
</tr>
<tr>
<td><input type="text" placeholder="panjang" pattern="[0-9]{1,6}" required id="panjang" name="panjang" /></td>
</tr>
<tr>
<td><input type="text" placeholder="lebar" pattern="[0-9]{1,6}" required id="lebar" name="lebar" /></td>
</tr>
<tr>
<td><input type="text" placeholder="tinggi" pattern="[0-9]{1,6}" required id="tinggi" name="tinggi" /></td>
</tr>
<tr>
<td><input type="text" placeholder="keterangan" id="keterangan" name="keterangan" /></td>
</tr>
```

Gambar 3.3.5.1.2 Source Code jsp Tambah Ruangan

### 3.3.5.2 Sub-Modul Update Ruangan



Gambar 0.1 Desain Update Ruangan

Beberapa potongan Source Code-nya

```
if (request.getParameter("action").equals("edit")) {
    ruang.setIDruang(Integer.parseInt(request.getParameter("idruang")));
    ruang.setIDBlok(Integer.parseInt(request.getParameter("idblok")));
    ruang.setRuang(request.getParameter("ruang"));
    ruang.setPanjang(Integer.parseInt(request.getParameter("panjang")));
    ruang.setLebar(Integer.parseInt(request.getParameter("lebar")));
    ruang.setTinggi(Integer.parseInt(request.getParameter("tinggi")));
    ruang.setStatus(Integer.parseInt(request.getParameter("status")));
    ruang.setKeterangan(request.getParameter("keterangan"));
    log("ruangcontroller?action=edit");
    try {
        ruangdao.updateRuang(ruang);
        success = true;
    } catch (Exception e) {
        success = false;
    }
}
```

Gambar 0.2 source code update ruangan pada servlet

```
String query = "update ruang set idblok=?, ruang=?, panjang=?, lebar=?,"
    + "tinggi=?, status=?, keterangan=? where idruang=?";
try {
    PreparedStatement preparedStatement = (PreparedStatement) conn.prepareStatement(query);
    Statement statement = conn.prepareStatement(query);
    preparedStatement.setInt(1, ruang.getIDruang());
    preparedStatement.setString(2, ruang.getIDBlok());
    preparedStatement.setString(3, ruang.getRuang());
    preparedStatement.setInt(4, ruang.getPanjang());
    preparedStatement.setInt(5, ruang.getLebar());
    preparedStatement.setInt(6, ruang.getTinggi());
    preparedStatement.setString(7, ruang.getKeterangan());
    preparedStatement.executeUpdate();
} catch (SQLException e) {
    e.getMessage();
}
return success;
```

Gambar 0.3 source code update ruangan pada daoimp

```
<table>
<tr>
<td><input type="hidden" readonly="" hidden="true" placeholder="id ruang" required id="idruang" name="idruang" value="" /></td>
</tr>
<tr>
<td><input type="text" placeholder="id blok" value="" /></td>
</tr>
<tr>
<td><input type="text" placeholder="ruang" value="" /></td>
</tr>
<tr>
<td><input type="text" placeholder="panjang" pattern="[0-9]{1,6}" required id="panjang" name="panjang" value="" /></td>
</tr>
<tr>
<td><input type="text" placeholder="lebar" pattern="[0-9]{1,6}" required id="lebar" name="lebar" value="" /></td>
</tr>
<tr>
<td><input type="text" placeholder="tinggi" pattern="[0-9]{1,6}" required id="tinggi" name="tinggi" value="" /></td>
</tr>
<tr>
<td><input type="text" placeholder="keterangan" id="keterangan" name="keterangan" value="" /></td>
</tr>
```

Gambar 0.4 Source Code jsp Update ruangan

### 3.3.6 Modul Klien

Modul ini hanya bisa diakses oleh pegawai. Modul ini digunakan untuk menambah klien baru dan mengubah klien yang sudah terdaftar. Berikut sketsa tampilan halaman Klien setelah berhasil *login*.



Gambar 3.3.6.1 Desain data klien

Beberapa potongan *Source Code*-nya

```
public class klient {
    int idklient;
    String nama;
    String alamat;
    String telepon;
    String email;
    String keterangan;
}
```

Gambar 3.3.6.2 Source Code Model Klien

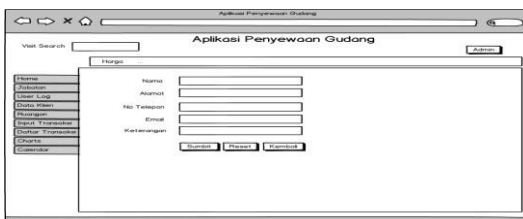
```
<thead>
<tr>
<th>nama lengkap</th>
<th>alamat</th>
<th>nomor telepon</th>
<th>email</th>
<th>keterangan</th>
<th>aksi</th>
</tr>
</thead>
<tbody>
<t>
<tr>
<td>rs.getString(1)</td>
<td>rs.getString(2)</td>
<td>rs.getString(3)</td>
<td>rs.getString(4)</td>
<td>rs.getString(5)</td>
<td>rs.getString(6)</td>
<td><a href="edit_klien.jsp?id=<rs.getInt(1)>">
<input type="submit" value="update" id="update" name="update" />
</td>
</tr>
</tbody>
</table>
} catch (Exception e) {
    e.printStackTrace();
}
```

Gambar 3.3.6.4 Source Code jsp Klien

```
public boolean addklien(klient klient);
public boolean updateklien(klient klient);
```

Gambar 3.3.6.3 Source Code dao Klien

#### 3.3.6.1 Sub-Modul Input Klien



Gambar 3.3.6.1.1 Desain input klien

Beberapa potongan *source code* untuk *input* klien

```
if (request.getParameter("action") != null) {
    if (request.getParameter("action").equals("add")) {
        klient.setName(request.getParameter("nama"));
        klient.setAlamat(request.getParameter("alamat"));
        klient.setTelepon(request.getParameter("telepon"));
        klient.setEmail(request.getParameter("email"));
        klient.setKeterangan(request.getParameter("keterangan"));
        log("klientcontroller?action=add");
        try {
            success = klientdao.addklien(klient);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

Gambar 3.3.6.1.3 source code Input Klien pada servlet

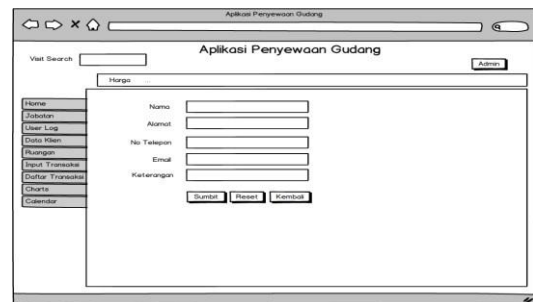
```
try {
    String query = "insert into klien ("
        + "nama, alamat, telepon, email, keterangan)"
        + "values (?, ?, ?, ?, ?)";
    PreparedStatement preparedStatement = conn.prepareStatement(query);
    preparedStatement.setString(1, klient.getName());
    preparedStatement.setString(2, klient.getAlamat());
    preparedStatement.setString(3, klient.getTelepon());
    preparedStatement.setString(4, klient.getEmail());
    preparedStatement.setString(5, klient.getKeterangan());
    int execute = preparedStatement.executeUpdate();
    System.err.println("Param : " + klient.getIdklien() + ", " + klient.getNama() + ", " + klient.getAlamat() + ", " + klient.getTelepon() + ", " + klient.getEmail() + ", " + klient.getKeterangan());
    if (execute == 1) {
        success = true;
    }
} catch (SQLException e) {
    e.printStackTrace();
}
```

Gambar 3.3.6.1.2 source code input Klien pada dsomip

```
<td>nama</td>
<td><input type="text" placeholder="nama" required id="nama" name="nama"/></td>
</tr>
<td>alamat</td>
<td><input type="text" placeholder="alamat" required id="alamat" name="alamat"/></td>
</tr>
<td>Telepon</td>
<td><input type="text" pattern="[0-9]{10,12}" placeholder="no telepon" required id="telepon" name="telepon"/></td>
</tr>
<td>Email</td>
<td><input type="email" placeholder="e-mail" required id="email" name="email"/></td>
</tr>
<td>keterangan</td>
<td><input type="text" placeholder="keterangan" id="keterangan" name="keterangan"/></td>
</tr>
```

Gambar 3.3.6.1.2 source code Input Klien pada jsp

#### 3.3.6.2 Sub-Modul Update Klien



Gambar 3.3.6.2.1 Desain update klien

Beberapa potongan *source code*-nya

```
if (request.getParameter("action").equals("edit")) {
    klient.setIdklien(Integer.parseInt(request.getParameter("idklient")));
    klient.setName(request.getParameter("nama"));
    klient.setAlamat(request.getParameter("alamat"));
    klient.setTelepon(request.getParameter("telepon"));
    klient.setEmail(request.getParameter("email"));
    klient.setKeterangan(request.getParameter("keterangan"));
    log("klientcontroller?action=edit");
    try {
        klientdao.updateklien(klient);
        success = true;
    } catch (Exception e) {
        success = false;
        e.printStackTrace();
    }
}
```

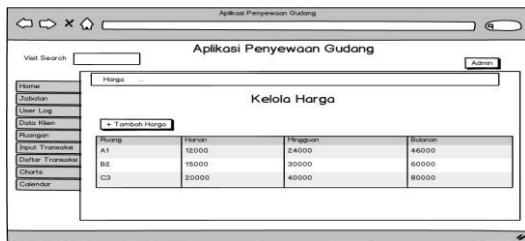
Gambar 3.3.6.2.2 source code update pengguna pada servlet

```
<form name="inputKlien" method="POST" action="">
<table>
<tr>
<td>ID Klien</td>
<td><input type="hidden" placeholder="Id Klien" required id="idKlien" name="idKlien" value="#{idKlien}"/></td>
</tr>
<tr>
<td>Nama</td>
<td><input type="text" placeholder="nama" required id="nama" name="nama" value="#{nama}"/></td>
</tr>
<tr>
<td>Alamat</td>
<td><input type="text" placeholder="alamat" required id="alamat" name="alamat" value="#{alamat}"/></td>
</tr>
<tr>
<td>No Telepon</td>
<td><input type="text" placeholder="no telepon" pattern="[0-9]{10,12}" required id="noTelepon" name="noTelepon" value="#{noTelepon}"/></td>
</tr>
<tr>
<td>Email</td>
<td><input type="email" placeholder="e-mail" required id="email" name="email" value="#{email}"/></td>
</tr>
<tr>
<td>Keterangan</td>
<td><input type="text" placeholder="keterangan" id="keterangan" name="keterangan" value="#{keterangan}"/></td>
</tr>
</table>
</form>
```

Gambar 3.3.6.2.3 Source code update klien pada servlet

3.3.7 Modul Harga

Modul ini hanya bisa diakses oleh *admin*. Modul ini digunakan untuk menambah harga ruangan baru dan mengubah harga ruangan yang sudah terdaftar. Berikut sketsa tampilan halaman Harga setelah berhasil *login*.



Gambar 3.3.7.1 Desain data harga

3.3.7.1 Sub-Modul Input Harga



Gambar 3.3.7.1.1 Desain tambah harga

Beberapa potongan Source code-nya

```
if (request.getParameter("action").equals("add")) {
    harga.setIDRuang(Integer.parseInt(request.getParameter("idRuang")));
    harga.setHarian(Integer.parseInt(request.getParameter("harian")));
    harga.setMingguan(Integer.parseInt(request.getParameter("mingguan")));
    harga.setBulanan(Integer.parseInt(request.getParameter("bulanan")));
    harga.setTahunan(Integer.parseInt(request.getParameter("tahunan")));
    log("HargaController.addAction()");
    try {
        hargaDao.addHarga(harga);
        success = true;
    } catch (Exception e) {
        success = false;
    }
}
if (request.getParameter("action").equals("edit")) {
    harga.setIDRuang(Integer.parseInt(request.getParameter("idRuang")));
    harga.setHarian(Integer.parseInt(request.getParameter("harian")));
    harga.setMingguan(Integer.parseInt(request.getParameter("mingguan")));
    harga.setBulanan(Integer.parseInt(request.getParameter("bulanan")));
    harga.setTahunan(Integer.parseInt(request.getParameter("tahunan")));
    log("HargaController.addAction()");
    try {
        hargaDao.addHarga(harga);
        success = true;
    } catch (Exception e) {
        success = false;
    }
}
```

Gambar 3.3.7.1.2 source code input harga pada servlet

Beberapa Potongan Source code-nya

```
int idHarga;
int idRuang;
int harian;
int mingguan;
int bulanan;
int tahunan;
```

Gambar 3.3.7.4 source code model harga

```
<thead>
<tr>
<th>id Ruang</th>
<th>Harian</th>
<th>Mingguan</th>
<th>Bulanan</th>
<th>Tahunan</th>
<th>aksi</th>
</tr>
</thead>
<tbody>
<tr>
<td><input type="button" value="Tambah Harga" /></td>
</tr>
<tr>
<td><table border="1">
<thead>
<tr>
<th>id Ruang</th>
<th>Harian</th>
<th>Mingguan</th>
<th>Bulanan</th>
<th>Tahunan</th>
<th>aksi</th>
</tr>
</thead>
<tbody>
<tr>
<td>B1</td>
<td>15000</td>
<td>30000</td>
<td>60000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>B2</td>
<td>18000</td>
<td>36000</td>
<td>72000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C3</td>
<td>20000</td>
<td>40000</td>
<td>80000</td>
<td></td>
<td></td>
</tr>
</tbody>
</table></td>
</tr>
```

Gambar 3.3.7.3 source code pada jsp harga

```
public boolean addHarga(Harga harga);
public boolean updateHarga(Harga harga);
```

Gambar 3.3.7.2 source code pada dao harga

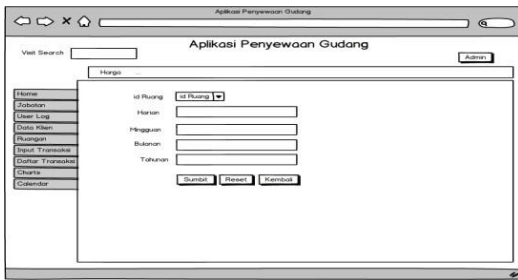
```
<form name="inputKlien" method="POST" action="">
<table>
<tr>
<td>ID Ruang</td>
<td><input type="text" value="#{idRuang}" /></td>
</tr>
<tr>
<td>Harian</td>
<td><input type="text" value="#{harian}" /></td>
</tr>
<tr>
<td>Mingguan</td>
<td><input type="text" value="#{mingguan}" /></td>
</tr>
<tr>
<td>Bulanan</td>
<td><input type="text" value="#{bulanan}" /></td>
</tr>
<tr>
<td>Tahunan</td>
<td><input type="text" value="#{tahunan}" /></td>
</tr>
</table>
</form>
```

Gambar 3.3.7.1.4 Source code jsp input klien

```
try {
    String query = "insert into harga ("
        + "idRuang, harian, mingguan, bulanan, tahunan) "
        + "values (?, ?, ?, ?, ?)";
    PreparedStatement preparedStatement = conn.prepareStatement(query);
    preparedStatement.setInt(1, harga.getIDRuang());
    preparedStatement.setInt(2, harga.getHarian());
    preparedStatement.setInt(3, harga.getMingguan());
    preparedStatement.setInt(4, harga.getBulanan());
    preparedStatement.setInt(5, harga.getTahunan());
    int execute = preparedStatement.executeUpdate();
    System.err.println("execute = " + query);
    harga.getIDRuang() + ", " + harga.getHarian() + ", " + harga.getMingguan() + ", "
        + harga.getBulanan() + ", " + harga.getTahunan();
    if (execute == 1) {
        success = true;
    }
}
```

Gambar 3.3.7.1.3 source code input harga pada daoimp

3.3.7.2 Sub-Modul *Update Harga*



Gambar 3.3.7.2.1 Desain *update harga*

Beberapa potongan *Source code*-nya

```

if (request.getParameter("action").equals("edit")) {
    log("EditPegguna.action-edit");
    pengguna.setIduser(Integer.parseInt(request.getParameter("iduser")));
    pengguna.setUsername(request.getParameter("username"));
    pengguna.setNama(request.getParameter("nama"));
    pengguna.setPassword(request.getParameter("password"));
    pengguna.setAlamat(request.getParameter("alamat"));
    pengguna.setTelepon(request.getParameter("telepon"));

    if (request.getParameter("action").equals("edit")) {
        log("EditPegguna.action-edit");
        pengguna.setIduser(Integer.parseInt(request.getParameter("iduser")));
        pengguna.setUsername(request.getParameter("username"));
        pengguna.setNama(request.getParameter("nama"));
        pengguna.setPassword(request.getParameter("password"));
        pengguna.setAlamat(request.getParameter("alamat"));
        pengguna.setTelepon(request.getParameter("telepon"));
    }
}
    
```

Gambar 3.3.8.2 *source code update profile* pada *servlet*

Gambar 3.3.7.2.2 *source code update harga* pada *servlet*

```

<table>
<tr>
<td><input type="hidden" readonly="" hidden="true" placeholder="idharga" required id="idharga" name="idharga" value=""></td>
</tr>
<tr>
<td><input type="hidden" name="idkuang" id="idkuang" value=""></td>
<td><input type="hidden" name="idkuang" id="idkuang" value=""></td>
</tr>
</table>
Database conn = new Database();
String idlevel = null;
String level = null;
Statement statement = conn.koneksi();
ResultSet rs = statement.executeQuery("select * from ruang");
%>
<select readonly="true" disabled="">
<tr>
<td><input type="hidden" name="idkuang" id="idkuang" value=""></td>
<td><input type="hidden" name="idkuang" id="idkuang" value=""></td>
</tr>
</table>
    
```

Gambar 3.3.7.2.4 *source code update harga* pada *jsp*

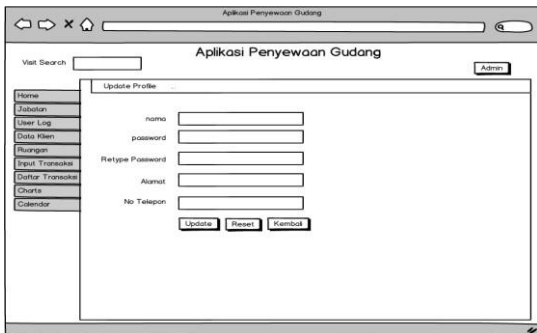
```

String query = "update harga set idkuang = ?, harian = ?, mingguan = ?, bulanan = ?, tahunan = ? where idharga = ?";
try {
    PreparedStatement preparedstatement = conn.prepareStatement(query);
    preparedstatement.setInt(1, harga.getIdkuang());
    preparedstatement.setInt(2, harga.getHarian());
    preparedstatement.setInt(3, harga.getMingguan());
    preparedstatement.setInt(4, harga.getBulanan());
    preparedstatement.setInt(5, harga.getTahunan());
    preparedstatement.executeUpdate();
}
    
```

Gambar 3.3.7.2.3 *source code update harga* pada *daoimp*

3.3.8 Modul *Update Profile*

Modul ini bisa diakses oleh semua pengguna. Modul ini digunakan untuk mengubah data pengguna yang sudah terdaftar. Berikut sketsa tampilan halaman *Update Profile*.



Gambar 3.3.8.1 Desain *update profile*



```

<form name="inputpengguna" method="POST" action="" onsubmit="return validate()">
<table>
<tr>
<td><input type="hidden" placeholder="id user" required id="idUser" name="idUser" value="(idUser%)" /></td>
</tr>
<tr>
<td><input type="hidden" placeholder="username" required id="username" name="username" value="(Busername%)" /></td>
</tr>
<tr>
<td>Nama</td>
<td><input type="text" placeholder="nama" required id="nama" name="nama" value="(Bnama%)" /></td>
</tr>
<tr>
<td>Password</td>
<td><input type="password" placeholder="password" maxlength="10" min="6" required id="password" name="password" /></td>
</tr>

```

Gambar 3.3.8.4 source code update profile pada jsp

```

String query = "update user set username = ?, nama = ?, password = md5(?),"
+ " alamat = ?, telepon = ? where iduser = ?";
try {
    PreparedStatement preparedStatement = conn.prepareStatement(query);
    Statement statement = conn.prepareStatement(query);
    preparedStatement.setString(1, pengguna.getUsername());
    preparedStatement.setString(2, pengguna.getUserName());
    preparedStatement.setString(3, pengguna.getName());
    preparedStatement.setString(4, pengguna.getPassword());
    preparedStatement.setString(5, pengguna.getAlamat());
    preparedStatement.setString(6, pengguna.getTelepon());
    preparedStatement.executeUpdate();
}

```

Gambar 3.3.8.3 source code update profile pada daoimp

## Bab 4 Penggunaan Produk

### 4.1 Instalasi Produk

- Untuk dapat menjalankan aplikasi berbasis *web* ini, diperlukan *server/localserver* yang memuat Apache dan MySQL. Bisa menggunakan XAMPP yang didalamnya telah didukung dengan Apache dan MySQL seperti xampp-1.8.1. Untuk instalasinya bisa diunduh di *link* berikut [www.filehorse.com/download-xampp](http://www.filehorse.com/download-xampp) dan cara instalasinya dapat diikuti pada *link* berikut [www.indrabt.com/panduan-menginstall-xampp.html](http://www.indrabt.com/panduan-menginstall-xampp.html).
- Untuk menjalankan aplikasi ini juga memerlukan *editor* NetBeans IDE. Untuk instalasinya NetBeans IDE 7.4 bisa diunduh di *link* berikut <https://netbeans.org/downloads> dan cara instalasinya dapat diikuti pada *link* berikut <https://netbeans.org/community/releases/74/install.html>
- Dapatkan *source* aplikasi dari *link* berikut [www.example.com/aplikasipenyewaan-gudang.war](http://www.example.com/aplikasipenyewaan-gudang.war)
- Ekstrak file tersebut di folder xampp/htdocs/Aplikasi penyewaan gudang. Pada folder tersebut terdapat basis data "selfstorage.sql" dan "AplikasiGudang.war"
- Buka sebuah *web browser*, pada *address bar* ketik localhost/phpmyadmin untuk persiapan basis datanya. Pilih *import*, lalu *browse* ke folder "Aplikasi penyewaan gudang" yang telah diunduh dan sebelumnya telah diekstrak. Cari file "selfstorage.sql". Tekan *GO*, maka seharusnya basis data berhasil di-*import*.
- Buka *editor* NetBeans IDE yang telah diinstall. Kemudian pada NetBeans IDE pilih menu *open project*. Cari file AplikasiGudang.war yang sebelumnya telah diunduh.
- Setelah file sukses dibuka, klik kanan pada file "AplikasiGudang" tersebut, kemudian pilih *run file* untuk menjalankan aplikasi tersebut.
- Jangan lupa untuk mengaktifkan apache dan MySQL di XAMPP dan mengaktifkan Databases dan Apache Tomcat Server pada NetBeans di menu *services*.

### 4.2 Petunjuk Penggunaan Produk

#### 1. Halaman Login



Gambar 4.2.1 Halaman Login

Pada saat pertama kali aplikasi dijalankan, maka akan tampil halaman *login* seperti gambar di atas. Kita bisa *login* sebagai superadmin, admin dan admin pegawai. Kita harus *login* sebagai superadmin dahulu agar dapat melihat siapa saja yang bisa *login* di aplikasi ini. *Login* superadmin sementara dengan *username*: ilham dan *password*: ilham.

Pertama user menginputkan *username* dan *password* yang sudah terdaftar di *database*. Jika *username* telah tidak aktif, pemberitahuan akan muncul seperti "maaf anda tidak dapat mengakses halaman ini" dan jika *username* atau *password* salah, pemberitahuan akan muncul seperti "maaf *username* atau *password* anda salah". Kemudian jika *username* dan *password* benar, user akan masuk ke halaman *dashboard* sesuai dengan *level* dari *username* tersebut.

#### 2. Halaman Superadmin

No	Nama Admin	Ruang	Tanggal Awal Sewa	Tanggal Akhir Sewa	Duit Sewa	Keterangan	Total	Aksi
1	A1		2014-07-13	2014-07-15	aktif	penyewaan ruang pada hari ini	75000	<a href="#">Detail</a>
2	A2		2014-07-13	2014-07-17	aktif	penyewaan ruang pada hari ini	75000	<a href="#">Detail</a>
3	C5		2013-09-30	2014-07-05	aktif	penyewaan ruang pada hari ini	1200000	<a href="#">Detail</a>

Gambar 4.2.2 halaman superadmin

Ini adalah tampilan halaman superadmin jika sudah berhasil *login* dengan level superadmin. Superadmin dapat mengakses *home*, *chart*, grafik keuangan, mengelola data jabatan, mengelola data user, mengelola data gudang, mengelola data harga, mengelola daftar transaksi. Yang dimaksud mengelola data disini seperti tambah dan ubah. Halaman ini berfungsi untuk melihat daftar masa penyewaan ruang gudang yang akan habis kurang dari 7 hari. User dapat mengubah status dari masa sewa tersebut, dengan menekan tombol ganti status pada setiap baris yang ada di tabel daftar masa sewa yang akan habis. Jika masa sewa sudah habis dan tidak ada pemberitahuan dari klien maka user mengganti masa sewa dengan memilih "closed by landlord" yang berarti penutupan penyewaan oleh pemilik ruang gudang. Jika klien ingin menutup masa sewa sebelum masa sewa habis maka user bisa memilih "closed by tenant" yang berarti penutupan penyewaan oleh klien. Setelah masa sewa ditutup, data yang sudah ditutup tidak akan muncul lagi di tabel.

3. Halaman admin



Gambar 4.2.3 Halaman admin

Ini adalah tampilan halaman admin jika sudah berhasil login dengan level admin. admin dapat mengakses *home*, *chart*, mengelola data gudang, mengelola data harga dan mengelola data daftar transaksi. Yang dimaksud mengelola data disini seperti tambah dan ubah.

Halaman ini berfungsi untuk melihat daftar masa penyewaan ruang gudang yang akan habis kurang dari 7 hari. User dapat mengubah status dari masa sewa tersebut, dengan menekan tombol ganti status pada setiap baris yang ada di tabel daftar masa sewa yang akan habis. Jika masa sewa sudah habis dan tidak ada pemberitahuan dari klien maka user mengganti masa sewa dengan memilih "closed by landlord" yang berarti penutupan penyewaan oleh pemilik ruang gudang. Jika klien ingin menutup masa sewa sebelum masa sewa habis maka user bisa memilih "closed by tenant" yang berarti penutupan penyewaan oleh klien. Setelah masa sewa ditutup, data yang sudah ditutup tidak akan muncul lagi di tabel.

4. Halaman admin pegawai



Gambar 4.2.4 Halaman admin pegawai

Ini adalah tampilan halaman admin jika sudah berhasil login dengan level pegawai. Admin pegawai dapat mengakses *home*, melihat status gudang, mengelola data klien, menambahkan transaksi, melihat daftar transaksi dan detail transaksi serta cetak transaksi.

Halaman ini berfungsi untuk melihat daftar masa penyewaan ruang gudang yang akan habis kurang dari 7 hari. User dapat mengubah status dari masa sewa tersebut, dengan menekan tombol ganti status pada setiap baris yang ada di tabel daftar masa sewa yang akan habis. Jika masa sewa sudah habis dan tidak ada pemberitahuan dari klien maka user mengganti masa sewa dengan memilih "closed by landlord" yang berarti penutupan penyewaan oleh pemilik ruang gudang. Jika klien ingin menutup masa sewa sebelum masa sewa habis maka user bisa memilih "closed by tenant" yang berarti penutupan penyewaan oleh klien. Setelah masa sewa ditutup, data yang sudah ditutup tidak akan muncul lagi di tabel.

5. Halaman untuk mengelola data user



Gambar 4.2.5 Halaman kelola data user

Pada halaman ini, user atau pengguna dapat dikelola oleh superadmin. Superadmin dapat menambah user baru, mengubah data dari user, mengubah status user dari aktif menjadi tidak aktif begitu juga sebaliknya.

Halaman ini terdapat tabel data detail user untuk melihat user yang masih aktif atau tidak aktif. Jika user ingin menambahkan user, tekan tombol yang ada diatas tabel yaitu "tambah user" dan jika user ingin mengubah data user, tekan tombol "ubah" yang ada di setiap baris tabel data detail user.

6. Halaman untuk menambah data user



Gambar 4.2.6 Halaman tambah data user

Pada halaman ini, yang hanya bisa mengakses adalah *superadmin*. *Superadmin* dapat menambahkan user yang berhak mengakses aplikasi ini. Pada halaman ini user memasukkan username, nama, *password*, validasi *password*, id level yaitu tingkatan level hak akses *login*, alamat dari user, no telepon dan status yaitu aktif atau tidak.

Setelah menekan tombol "tambah user" yang ada di halaman data user, user akan masuk ke halaman tambah user. Disini user menambahkan *username* yang belum ada di *database*, jika user menambahkan *username* yang sudah ada maka disamping kotak penambahan akan ada pemberitahuan seperti "username telah tersedia". Setelah itu user menambahkan nama yang harus menambahkan dengan huruf, menambahkan alamat, *password*, dan *re-type password*. *Re-type password* disini mengulang *password* yang sudah ditambahkan jika cocok akan ada pemberitahuan "password cocok". Kemudian menambahkan nomor telepon yang harus menambahkan lebih dari 10 digit. Lalu user menambahkan *level* dengan pilihan *superadmin*, *admin* dan pegawai. User juga menambahkan status dari user itu sendiri seperti aktif atau tidak aktif. Setelah data sudah benar ditambahkan, user menekan tombol "submit" kemudian akan masuk ke halaman data detail user.

7. Halaman untuk mengubah data user



Gambar 4.2.7 Halaman mengubah data user

Setelah menekan tombol “ubah” yang ada di halaman data user pada setiap baris tabel data user, user akan masuk ke halaman ubah user. Disini user tidak bisa mengubah *username* karena *username* tidak bisa diubah. Setelah itu user menambahkan nama yang harus menambahkan dengan huruf, menambahkan alamat, *password*, dan *re-type password*. *Re-type password* disini mengulang *password* yang sudah ditambahkan jika cocok akan ada pemberitahuan “*password cocok*”. Kemudian menambahkan nomor telepon yang harus menambahkan lebih dari 10 digit. Lalu user menambahkan *level* dengan pilihan *superadmin*, *admin* dan pegawai. User juga menambahkan status dari user itu sendiri seperti aktif atau tidak aktif. Setelah data sudah benar ditambahkan, user menekan tombol “*submit*” kemudian akan masuk ke halaman data detail user.

8. Halaman untuk *chart* pendapatan semua tahun



Gambar 4.2.8 Halaman *chart* dan laporan pendapatan semua tahun

Pada halaman ini, yang bisa mengakses hanya *superadmin*. *Chart* yang digunakan disini adalah *chart line*. Disini *superadmin* dapat melihat grafik pendapatan pada penyewaan gudang dalam semua tahun. User dapat melihat transaksi per tahun dengan memilih pada *box* yang ada dibawah *chart* pendapatan transaksi, lalu user menekan tombol “*submit*” kemudian akan masuk ke halaman *chart* pendapatan per tahun.

9. Halaman *chart* pendapatan per tahun



Gambar 4.2.9 Halaman *chart* dan laporan pendapatan per tahun

Pada halaman *chart* pendapatan per tahun, yang bisa mengakses hanya *superadmin*. Halaman ini tidak ditampilkan pada menu utama dan hanya bisa diakses pada saat memilih tahun yang akan dilihat pada halaman *chart*

pendapatan semua tahun. Halaman ini menampilkan tabel detail pendapatan perbulan beserta total pendapatan per tahun. *Chart* yang digunakan disini adalah *chart line*. Pada saat user menekan nama bulan yang ada di *chart*, user dapat melihat detail dari bulan tersebut. Berikut tampilan halaman detail *chart* per bulan.



Gambar 4.2.10 Halaman *chart* detail per bulan

10. Halaman daftar transaksi

ID Transaksi	Nama Klien	Tanggal Awal Sewa	Total Harga	Jumlah Aktif
10	Titi	2014-05-20	80000	1
11	Adina	2014-05-20	240000	1

Gambar 4.2.11 Halaman daftar transaksi

Pada halaman daftar transaksi, semua level user dapat mengakses halaman ini, seperti *superadmin*, *admin* dan pegawai. Pada saat mengakses halaman ini, tabel menampilkan daftar transaksi pada tanggal saat halaman ini diakses. User dapat mencari tanggal awal transaksi dengan menggunakan kotak pencarian lalu tekan *submit* agar dapat muncul di tabel daftar transaksi. Pada tabel daftar transaksi, terdapat informasi id transaksi, nama klien yang telah melakukan transaksi, tanggal awal sewa yaitu tanggal pada saat klien transaksi, total harga, dan jumlah aktif yaitu status aktif gedung yang sedang disewa. User juga dapat melihat informasi detail transaksi dengan menekan tombol id transaksi yang diinginkan pada tabel daftar transaksi.

11. Halaman informasi detail transaksi

Nama Klien	Ruang	Tanggal Awal Sewa	Tanggal Akhir Sewa	Detil Status	Keterangan	Total	Aktif
Adi	A2	2014-05-20	2014-05-20	aktif	penyewaan ruang gudang (200) Transaksi aktif	240000	aktif
Adi	A2	2014-05-20	2014-05-20	aktif	penyewaan ruang gudang (200) Transaksi aktif	240000	aktif
Adi	A2	2014-05-20	2014-05-20	aktif	penyewaan ruang gudang (200) Transaksi aktif	240000	aktif
Adi	A2	2014-05-20	2014-05-20	aktif	penyewaan ruang gudang (200) Transaksi aktif	240000	aktif

Gambar 4.2.12 Halaman informasi detail transaksi

Halaman ini tidak ditampilkan di menu utama. Halaman informasi detail transaksi hanya bisa diakses saat user memilih id transaksi pada halaman daftar transaksi. Halaman ini menampilkan detail transaksi berdasarkan id transaksi. User dapat melihat detail informasi klien yang berfungsi jika status masa sewa gedung sudah mau habis, user dapat menghubungi klien dengan melihat informasinya. Berikut tampilan *popup* detail klien.



Gambar 4.2.13 Halaman popup detail klien

Setelah user melihat detail klien, user dapat menutup proses sewa gudang dari klien tersebut. Penutupan proses sewa dapat dilakukan oleh klien yang bersangkutan dan bisa juga dilakukan oleh pemilik dari gudang tersebut. Penutupan proses sewa dilakukan dengan cara mengubah status pada detail klien yang ada di tabel informasi detail klien dengan menekan tombol ganti status pada setiap barisnya. Berikut tampilan popup ganti status.

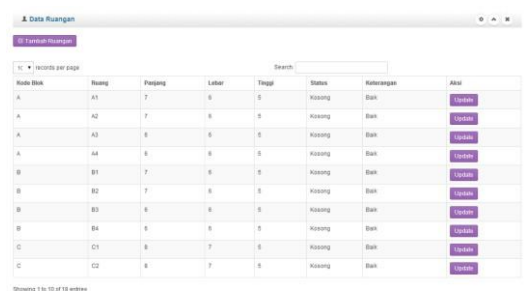


Gambar 4.2.14 Halaman popup ganti status

12. Halaman untuk menampilkan data blok dan data ruangan



Gambar 4.2.15 Halaman data blok



Gambar 4.2.16 Halaman data ruangan

Pada halaman ini, yang bisa mengakses adalah superadmin dan admin. Halaman ini menampilkan data dari blok dan ruangan. User dapat menambahkan atau mengubah data blok dan ruangan. Pada tabel data blok menampilkan kode blok, blok dan keterangan dari blok tersebut. Sedangkan pada tabel data ruangan menampilkan kode blok, ruang berdasarkan

blok, panjang ruangan, lebar ruangan, tinggi ruangan, status dari ruangan tersebut dan keterangan.

Jika user ingin menambahkan data blok, tekan tombol “tambah blok” yang ada diatas tabel data blok dan jika user ingin mengubah data blok, tekan tombol “ubah” yang ada pada setiap baris tabel data blok. Jika user ingin menambahkan data ruangan, tekan tombol “tambah ruangan” yang ada diatas tabel data ruangan dan jika user ingin mengubah data ruangan, tekan tombol “ubah” yang ada pada setiap baris tabel data ruangan. Disini jika blok belum ditambahkan maka ruangan tidak bisa ditambahkan, karena ruangan diisi berdasarkan blok yang sudah ada.

13. Halaman untuk menambah blok



Gambar 4.2.17 Halaman tambah blok

Halaman tambah blok tidak ditampilkan pada menu utama. Halaman ini dapat diakses pada saat user menekan tombol tambah blok pada halaman data blok dan ruangan. User menambahkan blok dan keterangan. Pada saat user menambahkan blok, jika blok sudah tersedia pada database, maka akan muncul pemberitahuan disamping kotak tambah blok seperti “blok sudah tersedia”. Setelah data sudah benar ditambahkan, user menekan tombol “submit”. Pada saat user menekan tombol submit akan pindah ke halaman data blok dan data ruangan, dan user dapat melihat data yang baru saja ditambahkan.

14. Halaman untuk mengubah data blok



Gambar 4.2.18 Halaman mengubah data blok

Setelah user menambah data blok, user dapat mengubah data blok dengan menekan tombol update yang ada di baris tabel. Pada saat user menambahkan blok, jika blok sudah tersedia pada database, maka akan muncul pemberitahuan disamping kotak tambah blok seperti “blok sudah tersedia”. Setelah data sudah benar ditambahkan, user menekan tombol “submit”. Pada saat user menekan tombol submit akan pindah ke halaman data blok dan data ruangan, dan user dapat melihat data yang baru saja diubah.

15. Halaman untuk menambah ruangan



Gambar 4.2.19 Halaman tambah data ruangan

Halaman tambah ruangan tidak ditampilkan pada menu utama. Halaman ini dapat diakses pada saat user menekan tombol tambah ruangan pada halaman data blok dan ruangan. User menambahkan ruangan berdasarkan blok yang sudah ada.

Disini user menambahkan blok yang sudah terdaftar di *database*. Lalu user juga menambahkan ruangan, jika ruangan sudah terdaftar maka akan ada pemberitahuan seperti "ruang sudah tersedia". User menambahkan panjang, lebar dan tinggi dari ukuran ruangan, user harus menambahkan dengan angka jika huruf maka tidak akan masuk ke *database*. Setelah data benar ditambahkan, user menekan tombol *submit*. Pada saat user menekan tombol *submit* akan pindah ke halaman data blok dan data ruangan, dan user dapat melihat data yang baru saja ditambahkan.

#### 16. Halaman untuk mengubah data ruangan



Gambar 4.2.20 Halaman mengubah data ruangan

Setelah user menambahkan data ruangan, user juga dapat mengubah data ruang. Disini user hanya bisa mengubah panjang, lebar, tinggi dan status ruangan. User menambahkan panjang, lebar dan tinggi dari ukuran ruangan, user harus menambahkan dengan angka jika huruf maka tidak akan masuk ke *database*. Setelah data benar ditambahkan, user menekan tombol *submit*. Pada saat user menekan tombol *submit* akan pindah ke halaman data blok dan data ruangan, dan user dapat melihat data yang baru saja ditambahkan.

#### 17. Halaman untuk menampilkan data harga

ID Ruang	Ruang	Mingguan	Bulanan	Tahunan	Aksi
10	12000	7000	20000	200000	Aksi
11	12000	7000	20000	200000	Aksi
12	12000	7000	20000	200000	Aksi
13	12000	7000	20000	200000	Aksi
14	12000	7000	20000	200000	Aksi
15	12000	7000	20000	200000	Aksi
16	12000	7000	20000	200000	Aksi
17	12000	7000	20000	200000	Aksi
18	12000	7000	20000	200000	Aksi
19	12000	7000	20000	200000	Aksi
20	12000	7000	20000	200000	Aksi

Gambar 4.2.21 Halaman data harga

Halaman ini dapat diakses oleh *superadmin* dan *admin*. Pada halaman ini di user dapat melihat data harga berdasarkan id ruang. Pada tabel harga menampilkan id ruang, harga harian, harga mingguan, harga bulanan dan harga tahunan. User juga dapat mengubah dan menambahkan data harga berdasarkan id ruang.

Jika user ingin menambah data harga maka tekan tombol "tambah harga" yang ada diatas tabel data harga. Dan jika user ingin mengubah data harga maka tekan tombol "ubah harga" yang ada pada setiap baris tabel data harga.

#### 18. Halaman untuk menambah harga



Gambar 4.2.22 Halaman tambah harga

Halaman ini tidak ditampilkan pada menu utama. Halaman ini dapat diakses pada saat user menekan tombol tambah harga pada halaman data harga. User menambahkan harga berdasarkan ruang yang belum mempunyai harga. Saat user menambahkan harga, maksimal angkanya adalah 10 *digit*. Setiap tambah harga akan ada validasi angka, jika user menginputkan huruf, tombol *submit* tidak akan berfungsi. Setelah user berhasil menambahkan harga, halaman akan pindah ke halaman data harga, dan user dapat melihat data yang baru saja ditambahkan.

#### 19. Halaman untuk mengubah data harga



Gambar 4.2.23 Halaman mengubah data harga

Setelah user menambahkan data harga, user dapat mengubah data harga dengan menekan tombol pada baris tabel data harga. User disini hanya bisa mengubah data harga saja. Saat user menambahkan harga, maksimal angkanya adalah 10 *digit*. Setiap tambah harga akan ada validasi angka, jika user menginputkan huruf, tombol *submit* tidak akan berfungsi. Setelah user berhasil menambahkan harga, halaman akan pindah ke halaman data harga, dan user dapat melihat data yang baru saja ditambahkan.

#### 20. Halaman *chart* pendapatan per tanggal



Gambar 4.2.24 Halaman *chart* dan laporan pendapatan per tanggal

Halaman ini hanya dapat diakses oleh *superadmin*. Halaman ini menampilkan *chart* dan laporan berdasarkan rentang tanggal yang dimasukkan pada kotak pencarian.

User dapat melihat *chart* dan laporan dengan memilih tanggal awal dan akhir yang ada diatas *chart*. Setelah user memilih tanggal, user tekan tombol *submit*. Pada saat user menekan tombol *submit*, *chart* dan laporan akan berubah berdasarkan tanggal yang dipilih.

21. Halaman status ruangan



Gambar 4.2.25 Halaman status ruangan

Halaman ini dapat diakses oleh admin dan pegawai. Pada halaman ini menampilkan data status ruangan yang kosong atau status ruangan yang isi. Baris tabel pada ruangan yang kosong akan berwarna merah, sedangkan baris tabel pada ruangan yang isi akan berwarna kuning. User dapat melihat detail klien dari ruangan yang statusnya isi. Berikut tampilan halaman *popup* detail klien.



Gambar 4.2.26 Halaman *popup* ruangan detail klien

22. Halaman *chart* ketersediaan ruangan



Gambar 4.2.27 Halaman *chart* dan laporan ketersediaan ruangan

Halaman ini hanya dapat diakses oleh admin. Pada halaman ini menampilkan *chart* ketersediaan ruangan berdasarkan tanggal yang dimasukkan pada kotak pencarian. Pada tabel laporan ketersediaan ruangan, user dapat melihat detail dari ketersediaan ruangan berdasarkan tanggal. User mencari tanggal yang ada diatas *chart* dan laporan, setelah user mencari tanggal kemudian user menekan tombol *submit*. Setelah menekan tombol *submit*, laporan dan *chart* akan berubah sesuai tanggal yang dimasukkan.

23. Halaman untuk menampilkan data klien

Gambar 4.2.28 Halaman data klien

Halaman ini hanya dapat diakses oleh pegawai. Pegawai bertugas untuk menginputkan klien yang akan transaksi sewa gudang. Pada halaman ini user dapat menambahkan atau mengubah data dari klien. Pada tabel klien menampilkan nama lengkap dari klien, alamat, nomor telepon, email dan keterangan dari klien tersebut.

Jika user ingin menambahkan data klien maka tekan tombol "tambah klien" yang ada diatas tabel detail data klien. Dan jika user ingin mengubah data klien maka tekan tombol "ubah klien" yang ada di setiap baris tabel detail data klien.

24. Halaman untuk menambah klien

Gambar 4.2.29 Halaman tambah klien

Halaman ini tidak ditampilkan pada menu utama. Halaman ini hanya bisa diakses pada saat user menekan tombol tambah klien pada halaman data klien. Setiap penambahan data akan ada validasi seperti nama dan alamat tidak bisa ditulis dengan angka, nomor telepon hanya bisa diinputkan dengan angka, dan email harus sesuai penulisan email. Setelah berhasil validasi, user menekan tombol submit dan halaman akan pindah ke halaman data klien, dan user dapat melihat data yang baru ditambahkan.

25. Halaman untuk mengubah data klien

Gambar 4.2.30 Halaman mengubah data klien

Pada halaman ini user dapat mengubah data klien yang telah diinputkan. User dapat mengakses halaman ini dengan menekan tombol *update* yang ada di tabel data klien yang ada pada halaman kelola klien. Setiap mengubah data akan ada validasi seperti nama dan alamat tidak bisa ditulis dengan angka, nomor telepon hanya bisa diinputkan dengan angka, dan email harus sesuai penulisan email. Setelah berhasil validasi, user menekan tombol submit dan halaman akan pindah ke halaman data klien, dan user dapat melihat data yang baru diubah.

26. Halaman untuk menambahkan transaksi



Gambar 4.2.31 Halaman tambah transaksi

Halaman ini hanya dapat diakses oleh pegawai. Pegawai bertugas untuk *input* transaksi klien. Pertama user harus mencari nama klien yang sudah terdaftar di database, kemudian tekan pada nama yang bersangkutan lalu akan muncul detail data klien pada kotak nama, alamat dan telepon sesuai dengan yang ada di database atau data klien yang diinputkan pada halaman tambah klien. Kemudian untuk menambahkan transaksi, user menekan tombol tambah. Berikut tampilan popup tambah transaksi.



Gambar 4.2.32 Halaman *popup* tambah transaksi

Pada *popup* tambah data transaksi, nama ruang disini berdasarkan nama blok. Jika nama blok tidak dipilih maka nama ruang akan kosong. Tipe sewa disini yaitu hari, minggu, bulan dan tahun. Dan total adalah kalkulasi dari harga per ruangan dari tipe sewa dijumlahkan dengan lama sewa. Setiap penambahan transaksi akan ditampung pada tabel yang ada di halaman tersebut, jika ingin mengubah penambahan transaksi, user menekan tombol ganti yang ada di setiap baris tabel. Setelah penambahan transaksi selesai user menekan tombol simpan data, kemudian akan pindah ke halaman daftar transaksi dan cetak transaksi sesuai data transaksi yang baru saja diinputkan.

27. Halaman untuk cetak transaksi



Gambar 4.2.33 Halaman cetak transaksi

Halaman ini tidak ditampilkan pada menu utama. Halaman ini dapat diakses pada saat user menekan tombol simpan data yang ada di halaman tambah transaksi. Cetak transaksi menampilkan data dari klien yang bertransaksi, data transaksi yaitu berupa ruangan yang disewa, tanggal akhir sewa, status dari masa sewa, keterangan status sewa, total harga ruangan dan total keseluruhan dari transaksi. Kemudian user dan klien diberikan kolom untuk tanda tangan transaksi.

Bab 5  
Penutup

5.1 Hambatan yang Dialami

Hambatan yang dialami selama pembuatan produk ini adalah sebagai berikut.

1. Pengetahuan penulis mengenai *javascript, jquery, jsp, datatable, highchart, ajax* yang masih sedikit, sehingga terkadang menghambat pekerjaan.
2. Penyatuan aplikasi antar modul membutuhkan waktu yang cukup lama, karena sering terjadi perubahan atau penambahan terhadap masing-masing modul yang dikerjakan.

5.2 Solusi

Solusi dari hambatan yang dialami selama pembuatan produk ini adalah sebagai berikut.

1. Penulis mencari referensi-referensi dari internet dan berbagai buku tentang *javascript, jquery, jsp, datatable, highchart* dan *ajax* serta bertanya kepada *trainer* terhadap kesulitan yang dialami.
2. Melakukan penyatuan aplikasi diakhir ketika setiap modul benar-benar selesai dikerjakan.

5.3 Saran Pengembangan

Untuk pengembangan produk selanjutnya, maka disarankan beberapa hal sebagai berikut.

1. Aplikasi ini disarankan menggunakan database oracle karena membutuhkan tempat penyimpanan data yang besar.
2. Aplikasi ini disarankan membuat sistem *booking* agar klien mudah untuk menyewa gudang sebelum waktu masa penyewaan berlangsung.
3. Aplikasi ini disarankan untuk menambahkan modul pembayaran agar klien dapat melakukan pembayaran ruang gudang yang jumlahnya besar.
4. Aplikasi ini disarankan untuk menambahkan modul keuangan agar laporan keuangan menjadi terstruktur terhadap pemasukkan dan pengeluaran.

Daftar Pustaka

[1] M. Huda, Membuat Aplikasi Database dengan Java, MySQL, dan NetBeans, Jakarta: PT. Elex Media Komputindo, 2010.

[2] G. S. Hartati, Pemrograman GUI Swing Java dengan NetBeans 5, Yogyakarta: Penerbit Andi, 2008.

[3] K. S. Jonathan Chaffer, Learning JQuery, American: Packt, 2007.

[4] H. Alatas, Responsive Web Design dengan PHP dan Bootstrap, Jakarta: Lokomedia, 2010.

- [5] J. Kuan, Learning Highchart, American: Packt, 2012.
- [6] Anonim, Naskah Kursus Web Programming Using PHP and MySQL, Depok: Lembaga Pengembangan Komputerisasi Universitas Gunadarma, 2002.
- [7] Apache, "Apache," Apache, Jumat Juni 2014. [Online]. Available:  
[http://httpd.apache.org/ABOUT\\_APACHE.html](http://httpd.apache.org/ABOUT_APACHE.html).  
[Accessed Friday Juni 2014].
- [8] M. A. Sunyoto, Ajax Membangun Web dengan Teknologi Asynchronous JavaScript & XML, Jakarta: Andi Publisher, 2009.
- [9] S. H. Wijono, Pemograman Java Servlet dan JSP dengan NetBeans, Yogyakarta: Andi Publisher, 2007.
- [10] A. Prasetio, Buku Pintar Pemrograman Web, Jakarta: MediaKita, 2012.