



## **APLIKASI SISTEM INFORMASI PELAYANAN JASA LAUNDRY PADA RAMA LAUNDRY DEPOK BERBASIS JAVA NETBEANS**

Tugas Akhir  
diajukan untuk melengkapi  
persyaratan mencapai  
gelar sarjana

NAMA : DEVIANTI NUR SAFITRI  
NPM : 201743501222

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNIK DAN ILMU KOMPUTER  
UNIVERSITAS INDRAPRASTA PGRI  
JAKARTA  
2021**

## **LEMBAR PERSETUJUAN**

Nama : Devianti Nur Safitri  
NPM : 201743501222  
Fakultas : Teknik dan Ilmu Komputer  
Program Studi : Teknik Informatika  
Judul Tugas Akhir : Aplikasi Sistem Informasi Pelayanan Jasa Laundry Pada  
Rama Laundry Depok Berbasis Java Netbeans

Telah diperiksa dan disetujui untuk diujikan

Pembimbing Materi



Dewi Anjani, M.Kom.

0317038702

Pembimbing Teknik



Eddy Saputra, M.Pd.I

0302038203

## **LEMBAR PENGESAHAN**

Nama : Devianti Nur Safitri  
NPM : 201743501222  
Fakultas : Teknik dan Ilmu Komputer  
Program Studi : Teknik Informatika  
Judul Tugas Akhir : Aplikasi Sistem Informasi Pelayanan Jasa Laundry  
Pada Rama Laundry Depok Berbasis Java  
Netbeans

Panitia Ujian

Ketua : Prof. Dr. Sumaryoto  
Sekretaris : Ir. H. Soepardi Harris,M.T.

Anggota :

No	Nama	Tanda Tangan
1.		
2.		
3.		

## **LEMBAR PERNYATAAN**

Yang bertanda tangan dibawah ini :

Nama : Devianti Nur Safitri  
NPM : 201743501222  
Program Studi : Teknik Informatika

Dengan ini menyatakan bahwa tugas akhir dengan judul **Aplikasi Sistem Informasi Pelayanan Jasa Laundry Pada Rama Laundry Depok Berbasis Java Netbeans** beserta seluruh isinya adalah benar-benar karya saya sendiri. Saya tidak melakukan penjiplakan atau pengutipan dengan cara-cara yang tidak sesuai dengan etika ilmu yang berlaku dalam masyarakat keilmuan. Atas pernyataan ini, saya siap menanggung resiko / sanksi apabila di kemudian hari ditemukan adanya pelanggaran etika keilmuan atau ada klaim dari pihak lain terhadap keaslian karya saya ini sesuai dengan Undang-Undang Republik Indonesia Nomor 20 Tahun 2003 tentang Sistem Pendidikan Nasional Bab V Pasal 25 ayat 2.

Demikian Pernyataan ini saya buat untuk dimanfaatkan sesuai dengan keperluan.

Yang menyatakan,



(Devianti.N.S )

## **ABSTRAK**

- A. Devianti Nur Safitri, NPM : 201743501222
- B. **Aplikasi Sistem Informasi Pelayanan Jasa Laundry Pada Rama Laundry Depok Berbasis Java Netbeans.** Tugas Akhir; Jakarta; Fakultas Teknik dan Ilmu Komputer: Program Studi Teknik Informatika; Universitas Indraprasta PGRI, Agustus, 2017
- C. xvi + 5 Bab + 259 halaman
- D. Kata Kunci : Sistem Informasi, Barang, Transaksi, Java Netbeans
- E. Penelitian ini bertujuan untuk membuat aplikasi sistem pelayanan jasa laundry pada Rama Laundry Depok yang masih menggunakan sistem manual dalam pendataan dan pencatatan transaksi, merancang aplikasi sistem pelayanan jasa laundry guna pendataan barang masuk dan transaksi berbasis Java Netbeans. Metode penelitian yang digunakan adalah dengan menggunakan metode Waterfall yang merupakan salah satu metode pengembangan perangkat lunak yang terdiri dari analysis, design, coding, testing, dan maintenance. Hasil yang akan dicapai adalah aplikasi pelayanan jasa laundry yang menjadikan sistem transaksi laundry menjadi terkomputerisasi, serta pencatatan laporan menjadi lebih mudah dan akurat. Selain itu dengan adanya aplikasi ini akan memudahkan pekerjaan operator laundry dalam melakukan pendataan barang, serta memudahkan pemilik laundry dalam melihat pengeluaran dan pemasukan transaksi setiap hari, setiap bulan, dan setiap tahun. Dalam membangun sistem ini digunakan alat bantu perancangan sistem, yaitu Diagram Alur Data (DAD) Konteks, Nol dan Rinci, serta menggunakan bahasa pemrograman Java dengan Netbeans IDE, database MySQL menggunakan XAMPP dan Ireport sebagai pencetak laporannya.
- F. Daftar Pustaka : 10 buku (2005 – 2018)
- G. Pembimbing : **Dewi Anjani, M.Kom.** (Pembimbing Materi)  
: **Eddy Saputra, M.Pd.I.** (Pembimbing Teknik)

## **LEMBAR MOTO**

Namanya juga rahasia, pasti tidak semua orang yang tahu ya kan? Begitu juga dengan rahasia kesuksesan. Intinya, kamu memiliki sesuatu yang tidak banyak orang yang tahu, kemudian hal itu kamu jadikan sebagai kelebihanmu dan kamu manfaatkan sebaik-baiknya.

## **KATA PENGANTAR**

Dengan memanjatkan puji syukur kehadirat Allah SWT yang telah melimpahkan rahmat dan karunia-Nya kepada penulis sehingga akhirnya penulis dapat menyelesaikan skripsi ini dengan tepat waktu.

Skripsi yang berjudul : “ **Aplikasi Sistem Informasi Pelayanan Jasa Laundry Pada Rama Laundry Depok Berbasis Java Netbeans** “ ini ditulis untuk memenuhi salah satu syarat guna memperoleh gelar sarjana di Universitas Indraprasta PGRI.

Pada kesempatan yang baik ini, izinkanlah penulis menyampaikan rasa hormat dan ucapan terima kasih kepada semua pihak yang dengan tulus ikhlas memberikan bantuan dan dorongan kepada penulis dalam menyelesaikan skripsi ini, yaitu kepada :

1. Ibu Dewi Anjani, M.Kom. selaku dosen pembimbing materi skripsi Universitas Indraprasta PGRI.
2. Bapak Eddy Saputra, M.Pd.I. selaku dosen pembimbing teknik skripsi Universitas IndraPrasta PGRI.
3. Bapak Prof. Dr. H. Sumaryoto selaku Rektor Universitas Indraprasta PGRI
4. Bapak Ir. H. Soepardi Harris, M.T. selaku Dekan Fakultas Teknik dan Ilmu Komputer (FTIK) Universitas Indraprasta PGRI
5. Ibu Atie Ernawati, M.T. selaku wakil Dekan Fakultas Teknik dan Ilmu Komputer (FTIK) Universitas Indraprasta PGRI

6. Ibu Mei Lestari, M.Kom, selaku Ketua Program Studi Fakultas Teknik dan Ilmu Komputer (FTIK) Universitas Indraprasta PGRI
7. Ibu Ni Wayan Parwati, M.M, M.Kom selaku Sekretaris Program Studi Fakultas Teknik dan Ilmu Komputer (FTIK) Universitas Indraprasta PGRI
8. Bapak Dani Subardja Selaku Pemilik Rama Laundry
9. Keluarga tercinta yang selalu memberikan dorongan moril maupun materiil agar dapat melanjutkan kejenjang yang lebih tinggi.
10. Monica Regita, S.Kom. MMSI. terima kasih telah memberikan semangat dan motivasi kepada penulis.
11. Rekan-Rekan mahasiswa teknik informatika angkatan 2017 terkhusus kelas RR terima kasih atas doa dan dukungan kalian selama ini.
12. Semua pihak yang tidak dapat penulis sebutkan satu persatu atas segala bantuan untuk memudahkan penulis menyelesaikan skripsi ini.

Penulis menyadari bahwa penulisan laporan kerja praktek ini masih banyak kekurangannya baik bentuk isi maupun teknik penyajiannya, oleh sebab itu kritik yang bersifat membangun dari berbagai pihak penulis terima dengan terbuka serta sangat diharapkan.

Jakarta, 07 Agustus 2021

Penulis

## DAFTAR ISI

<b>HALAMAN JUDUL .....</b>	<b>i</b>
<b>LEMBAR PERSETUJUAN .....</b>	<b>ii</b>
<b>LEMBAR PENGESAHAN .....</b>	<b>iii</b>
<b>LEMBAR PERNYATAAN .....</b>	<b>iv</b>
<b>ABSTRAK .....</b>	<b>v</b>
<b>LEMBAR MOTO .....</b>	<b>vi</b>
<b>KATA PENGANTAR.....</b>	<b>vii</b>
<b>DAFTAR ISI.....</b>	<b>ix</b>
<b>DAFTAR TABEL .....</b>	<b>xi</b>
<b>DAFTAR GAMBAR.....</b>	<b>xii</b>
<b>DAFTAR SIMBOL .....</b>	<b>xv</b>
<b>DAFTAR LAMPIRAN .....</b>	<b>xvi</b>
<b>BAB I PENDAHULUAN.....</b>	<b>1</b>
A. Latar Belakang Masalah.....	1
B. Identifikasi Masalah.....	2
C. Batasan Masalah .....	3
D. Rumusan Masalah.....	3
E. Tujuan Penelitian .....	4
F. Kegunaan Penelitian .....	4
G. Sistematika Penulisan .....	5
<b>BAB II LANDASAN TEORI, PENELITIAN YANG RELAVAN DAN KERANGKA BERFIKIR .....</b>	<b>8</b>
A. Landasan Teori.....	8
B. Penelitian Yang Relevan.....	55
C. Kerangka Berfikir .....	59
<b>BAB III METODE PENELITIAN .....</b>	<b>61</b>
A. Waktu dan Tempat Penelitian .....	61
B. Desain Penelitian .....	62
C. Metode Pengumpulan Data.....	64
D. Langkah-Langkah Pengembangan Sistem .....	66

<b>BAB IV ANALISIS SISTEM BERJALAN DAN RANCANGAN SISTEM YANG DIUSULKAN .....</b>	<b>69</b>
A. Profil Rama Laundry.....	69
B. Visi dan Misi.....	69
C. Struktur Organisasi Perusahaan .....	70
D. Proses Bisnis Sistem Berjalan.....	73
E. Aturan Bisnis Sistem Berjalan .....	75
F. Dekomposisi Fungsi Sistem.....	76
G. Analisa Masukan, Proses dan Keluaran.....	78
H. Diagram Alir Data Sistem Berjalan .....	81
I. Analisis Permasalahan .....	85
J. Alternatif Penyelesaian Masalah.....	86
K. Aturan Bisnis Sistem yang Diusulkan .....	87
L. Dekomposisi Fungsi Sistem yang Diusulkan.....	88
M. Rancangan Masukan (Input), Proses, dan Keluaran (Output) Sistem yang Diusulkan .....	91
N. Diagram Aliran Data (DAD) yang Diusulkan .....	97
O. Kamus Data Sistem yang Diusulkan.....	103
P. Spesifikasi Proses Sistem yang Diusulkan.....	110
Q. Bagan Terstruktur yang Diusulkan .....	115
R. Spesifikasi Modul Sistem yang Diusulkan .....	120
S. Rancangan Basis Data Sistem yang Diusulkan.....	123
T. Rancangan Layar, Rancangan Form Masukan Data, dan Rancangan Keluaran .	131
U. Tampilan dan Penjelasan Layar, Tampilan Format Masukan, dan Tampilan Keluaran.....	144
<b>BAB V PENUTUP .....</b>	<b>158</b>
A. Simpulan .....	158
B. Saran .....	159

## **DAFTAR PUSTAKA**

## **DAFTAR RIWAYAT HIDUP**

## **LAMPIRAN**

## **DAFTAR TABEL**

Tabel 3.1	Tabel Waktu Penelitian
Tabel 4.1	Tabel Data Pelanggan
Tabel 4.2	Tabel Data Karyawan
Tabel 4.3	Tabel Data Jenis Barang
Tabel 4.4	Tabel Data Transaksi

## **DAFTAR GAMBAR**

Gambar 2.4.	Kerangka Berfikir
Gambar 3.1	Metode Waterfall
Gambar 4.1	Struktur Organisasi
Gambar 4.2	Dekomposisi Fungsi Sistem
Gambar 4.3	Diagram Konteks Sistem Berjalan
Gambar 4.4	Diagram Nol Sistem Berjalan
Gambar 4.5	Diagram Rinci Level 1.0 Sistem Berjalan
Gambar 4.6	Diagram Rinci Level 2.0 Sistem Berjalan
Gambar 4.7	Diagram Rinci Level 3.0 Sistem Berjalan
Gambar 4.8	Diagram Rinci Level 4.0 Sistem Berjalan
Gambar 4.9	Dekomposisi Fungsi Sistem Yang Diusulkan
Gambar 4.10	Diagram Konteks Yang Diusulkan
Gambar 4.11	Diagram Nol Yang Diusulkan
Gambar 4.12	Diagram Rinci Level 2.0 Yang Diusulkan
Gambar 4.13	Diagram Rinci Level 3.0 Yang Diusulkan
Gambar 4.14	Diagram Rinci Level 4.0 Yang Diusulkan
Gambar 4.15	Diagram Rinci Level 5.0 Yang Diusulkan
Gambar 4.16	Bagan Terstruktur Proses 1.0 Yang Diusulkan
Gambar 4.17	Bagan Terstruktur Proses 2.1 Yang Diusulkan
Gambar 4.18	Bagan Terstruktur Proses 2.2 Yang Diusulkan
Gambar 4.19	Bagan Terstruktur Proses 2.3 Yang Diusulkan
Gambar 4.20	Bagan Terstruktur Proses 3.1 Yang Diusulkan
Gambar 4.21	Bagan Terstruktur Proses 3.2 Yang Diusulkan
Gambar 4.22	Bagan Terstruktur Proses 3.3 Yang Diusulkan
Gambar 4.23	Bagan Terstruktur Proses 4.1 Yang Diusulkan
Gambar 4.24	Bagan Terstruktur Proses 4.2 Yang Diusulkan
Gambar 4.25	Bagan Terstruktur Proses 5.1 Yang Diusulkan

Gambar 4.26	Bagan Terstruktur Proses 5.2 Yang Diusulkan
Gambar 4.27	Bagan Terstruktur Proses 5.3 Yang Diusulkan
Gambar 4.28	Bagan Terstruktur Proses 5.4 Yang Diusulkan
Gambar 4.29	Bagan Terstruktur Proses 5.5 Yang Diusulkan
Gambar 4.30	Bentuk UnNormalisasi
Gambar 4.31	Bentuk Normalisasi Pertama
Gambar 4.32	Bentuk Normalisasi Kedua
Gambar 4.33	ERD Yang Diusulkan
Gambar 4.34	Rancangan Layar Login
Gambar 4.35	Rancangan Layar Menu Utama
Gambar 4.36	Rancangan Layar Form Data Pelanggan
Gambar 4.37	Rancangan Layar Form Data Pelanggan
Gambar 4.38	Rancangan Layar Form Data Jenis Barang
Gambar 4.39	Rancangan Layar Menu Transaksi
Gambar 4.40	Rancangan Layar Form Menu Barang
Gambar 4.41	Rancangan Keluaran Laporan Pelanggan
Gambar 4.42	Rancangan Keluaran Laporan Karyawan
Gambar 4.43	Rancangan Keluaran Laporan Barang Yang Sudah Diambil
Gambar 4.44	Rancangan Keluaran Laporan Barang Yang Belum Diambil
Gambar 4.45	Rancangan Keluaran Laporan Transaksi
Gambar 4.46	Rancangan Keluaran Bukti Pembayaran
Gambar 4.47	Tampilan Menu Login
Gambar 4.48	Tampilan Menu Utama
Gambar 4.49	Tampilan Data Pelanggan
Gambar 4.50	Tampilan Data Karyawan
Gambar 4.51	Tampilan Data Jenis Barang
Gambar 4.52	Tampilan Menu Transaksi
Gambar 4.53	Tampilan Menu Barang
Gambar 4.54	Laporan Pelanggan

Gambar 4.55	Laporan Karyawan
Gambar 4.56	Laporan Barang Yang Sudah Diambil
Gambar 4.57	Laporan Barang Yang Belum Diambil
Gambar 4.58	Laporan Transaksi
Gambar 4.59	Bukti Pembayaran

## **DAFTAR SIMBOL**

Tabel 2.1	Simbol Diagram Alir Data
Tabel 2.2	Simbol Bagan Terstruktur
Tabel 2.3	Simbol ERD

## **DAFTAR LAMPIRAN**

Lampiran 1	Pengajuan Judul Skripsi / Tugas Akhir
Lampiran 2	Kartu Asistensi
Lampiran 3	Surat Permohonan Penelitian Skripsi / Tugas Akhir
Lampiran 4	Surat Balasan Keterangan Penelitian Skripsi / Tugas Akhir
Lampiran 5	Daftar Riwayat Hidup
Lampiran 6	Listing Program

# **BAB I**

## **PENDAHULUAN**

### **A. Latar Belakang Masalah**

Pada perkembangan zaman saat ini, banyak yang sudah menggunakan teknologi yang maju dan berkembang dengan menggunakan alat bantu komputer. Dengan menggunakan komputer dapat mempermudah setiap aktivitas yang kita lakukan secara cepat dan mudah. Seperti contohnya mempermudah pekerjaan dalam laundry untuk mendata jasa

Namun, masih banyak laundry yang membuat data barang masuk dan keluar dengan menggunakan alat tulis dan menghasilkan laporan berupa lembaran kertas. Semakin banyak pelanggan yang ingin menggunakan jasa ini, namun tidak bisa dengan cepat mendatanya, dan harus mengecek jenis cucian apa yang ingin di laundry, harus mengecek secara manual dan tidak praktis. Melihat-lihat data jenis cucian secara manual untuk mengecek berapa tarif nya yang memakan waktu cukup lama. Hal ini membuat pelanggan kecewa karena banyak membuang waktu dan menunggu lama untuk melakukan pengecekan. Selain pelanggan juga merasa menunggu lama untuk medapatkan bukti pembayaran jika pelanggan menggunakan jasa ini dalam jumlah banyak, karena proses pembuatan pembayaran masih menggunakan sistem manual yang dapat menyebabkan kesalahan dalam perhitungan harga.

Mengetahui berapa tarif cucian yang ingin di laundry merupakan salah satu pelayanan yang cepat untuk memuaskan pelanggan. Sistem yang masih manual dapat menyebabkan banyak permasalahan, seperti dalam pengolahan data, ataupun keterlambatan dalam pencarian data. Persediaan yang masih manual dirasa kurang efektif. Itu bisa berdampak yang cukup besar pada laundry ini. seperti di Rama laundry ini, laundry yang masih manual untuk pengelolaan data barang cucian , bila masih seperti itu akan mendapat banyak kesalahan dan kerugian. Aplikasi Sistem Informasi Pelayanan Jasa Laundry pada Rama Laundry dibuat dengan menggunakan aplikasi yang bernama Netbeans berbasis Java.

Dalam laundry ini maka didapatkan berbagai masalah, maka dari itu penulis ingin membuat sebuah aplikasi yang mampu menyelesaikan permasalahan dengan cara membuat aplikasi “**Aplikasi Sistem Informasi Pelayanan Jasa Laundry Pada Rama Laundry Depok Berbasis Java Netbeans**”.

## B. Identifikasi Masalah

Berdasarkan latar belakang, peneliti dapat mengidentifikasi masalah sebagai berikut :

1. Rama Laundry mencatat data jenis barang masih menggunakan manual dan pembuatan laporan tidak dapat tertata dengan rapih dan mudah hilang.

2. Banyak kendala yang dihadapi saat pelanggan ingin mengambil barang cucian di Rama Laundry sehingga menjadi kurang efektif dan tidak dapat diatasi.

### **C. Batasan Masalah**

Berdasarkan identifikasi masalah diatas, maka penulis hanya membatasi, yaitu :

1. Peneliti merubah cara sistem pencatatan dan laporan menjadi terkomputerisasi akan menjadi lebih efektif.
2. Peneliti membuat aplikasi sistem informasi untuk menghadapi kendala-kendala yang ada pada Rama Laundry Depok.

### **D. Rumusan Masalah**

Setelah melakukan penelitian dan pengamatan langsung di Rama Laundry Depok, dapat ditarik kesimpulan permasalahan yang ada, yaitu :

1. Apakah pencatatan data barang menggunakan Aplikasi Sistem Informasi Pelayanan Jasa Laundry Pada Rama Laundry Depok dapat tertera dengan rapih dan tidak hilang ?
2. Apakah kendala yang dihadapi Rama Laundry Depok dapat teratasi dengan Aplikasi Sistem Informasi Pelayanan Jasa Laundry Pada Rama Laundry Depok Berbasis Java Netbeans ?

## **E. Tujuan Penelitian**

Adapun tujuan dari penelitian ini adalah :

1. Mengetahui pencatatan data jenis barang dan laporan menggunakan Aplikasi Sistem Informasi Pelayanan Jasa Laundry Pada Rama Laundry Depok Berbasis Java Netbeans dapat tertata dengan rapih dan tidak hilang.
2. Agar meningkatkan efisiensi kinerja kerja di Rama Laundry maka diperlukannya database yang selalu terbaharu setiap detiknya disaat karyawan memerlukan data yang diinginkan.

## **F. Kegunaan Penelitian**

Kegunaan Penelitian sistem aplikasi pada laundry ini, yaitu :

### **1. Bagi Peneliti :**

- a. Meningkatkan wawasan peneliti, karena sangat banyak penemuan baru yang pastinya akan di dapat dari penelitian ini.
- b. Menambah pengalaman hidup penulis tentang perancangan suatu sistem aplikasi.
- c. Melengkapi syarat mencapai gelar kesarjanaan.

## **2. Bagi Rama Laundry :**

- a. Meningkatkan kinerja Rama Laundry
- b. Mempermudah pekerja Rama Laundry dalam mengelola data yang dihasilkan dari kegiatan mulai dari *input*, *proses* dan *output*.
- c. Mempermudah dalam pembuatan laporan di Rama Laundry.

## **3. Bagi Universitas :**

- a. Menjadi referensi pembaca dalam pengetahuan
- b. Memberikan gambaran bagi pembaca untuk mengetahui bagaimana mengembangkan Aplikasi Sistem Informasi Pelayanan Jasa Laundry Pada Rama Laundry Depok Berbasis Java Netbeans.
- c. Mengetahui kemampuan mahasiswa dalam menerapkan ilmunya dan sebagai bahan evaluasi.

## **G. Sistematika Penulisan**

Untuk mempermudah dan memperjelas dalam pembahasan masalah pada penulisan tugas akhir ini, penulis menyusun tugas akhir ini dengan sistematika penulisan sebagai berikut:

### **BAB I PENDAHULUAN**

Bab ini menguraikan tentang latar belakang masalah, identifikasi masalah, batasan masalah, rumusan masalah,

tujuan penelitian, kegunaan penelitian, dan sistematika penulisan.

**BAB II            LANDASAN TEORI, PENELITIAN YANG RELAVAN  
DAN KERANGKA BERFIKIR**

Bab ini menguraikan tentang landasan teori penelitian yang relevan dan kerangka berpikir.

**BAB III          METODOLOGI PENELITIAN**

Bab ini menguraikan tentang waktu dan tempat penelitian, desain penelitian, metode pengumpulan data, dan langkah-langkah pengembangan sistem

**BAB IV          ANALISIS SISTEM BERJALAN DAN RANCANGAN  
SISTEM YANG DIUSULKAN**

Pada bab ini membahas profil perusahaan, struktur organisasi perusahaan, proses bisnis sistem berjalan, aturan bisnis sistem yang berjalan, diagram aliran data konteks, nol dan rinci sistem berjalan, kamus data sistem berjalan, spesifikasi proses sistem berjalan, analisa basis data sistem berjalan, analisa permasalahan, alternatif penyelesaian masalah, aturan bisnis sistem yang diusulkan, dekomposisi fungsi sistem yang diusulkan, rancangan masukan proses, dan keluaran, diagram aliran data konteks, nol, dan rinci sistem yang diusulkan, kamus data sistem yang diusulkan,

spesifikasi proses sistem yang diusulkan, bagan terstruktur sistem yang diusulkan, spesifikasi modul sistem yang diusulkan, rancangan basis data sistem yang diusulkan, rancangan layar, rancangan form masukan data rancangan form keluaran.

## **BAB V                    SIMPULAN DAN SARAN**

Bab ini berisikan simpulan akhir dari bab-bab sebelumnya serta saran-saran penulis guna pengembangan lebih lanjut.

## **BAB II**

### **LANDASAN TEORI, PENELITIAN YANG RELAVAN DAN KERANGKA BERFIKIR**

#### **A. Landasan Teori**

##### **1. Definisi Aplikasi**

###### **a. Pengertian Aplikasi**

Aplikasi adalah *software* yang dibuat oleh suatu perusahaan komputer untuk mengerjakan tugas-tugas tertentu, misalnya *Microsoft Word* dan *Microsoft Excel*. *Application program* adalah suatu program komputer yang dibuat untuk mengerjakan atau menyelesaikan masalah-masalah khusus. Sedangkan *application package* adalah sebuah paket aplikasi lengkap, terdiri dari program tertentu yang dimasukkan di dalam memori komputernya berikut perangkat keras pendukung yang dibutuhkan (termasuk I/O device).

Menurut ( Chan,2017:4) , “aplikasi adalah koleksi window dan objek-objek yang menyediakan fungsi untuk aktivitas *user*, seperti pemasukan data, proses, dan pelaporan”.

Beberapa aplikasi yang digabung bersama menjadi suatu paket kadang disebut sebagai suatu paket atau *suite* aplikasi (*application suite*). Contohnya adalah *Microsoft Office* yang menggabungkan suatu aplikasi pengolah kata lembar kerja serta beberapa aplikasi lainnya. Aplikasi-aplikasi dalam suatu paket biasanya memiliki antarmuka

pengguna yang memiliki kesamaan sehingga memudahkan pengguna untuk mempelajari dan menggunakan tiap aplikasi.

Dari beberapa penjelasan yang sudah disebutkan, bahwa aplikasi merupakan *software* yang berfungsi untuk melakukan berbagai bentuk-bentuk pekerjaan atau tugas-tugas tetentu seperti penerapan, penggunaan dan penambahan data.

### **b. Klasifikasi Aplikasi**

Pada aplikasi dapat digolongkan menjadi beberapa kelas, diantaranya adalah :

- 1) Perangkat lunak perusahaan
- 2) Perangkat lunak infrastruktur perusahaan
- 3) Perangkat lunak informasi kerja
- 4) Perangkat lunak media dan hiburan
- 5) Perangkat lunak pendidikan
- 6) Perangkat lunak pengembangan media
- 7) Perangkat lunak rekayasa produk

Pada pengertian umumnya aplikasi adalah alat terapan yang difungsikan secara khusus dan terpadu sesuai kemampuan yang dimilikinya serta merupakan suatu perangkat komputer yang siap pakai bagi *user*.

### **c. Perancangan Aplikasi**

Perancangan aplikasi merupakan tahap selanjutnya setelah analis sistem, mendapatkan gambaran dengan jelas tentang apa yang dikerjakan pada analisa sistem, maka dilanjutkan dengan memikirkan bagaimana membentuk sistem tersebut. Perancangan sistem adalah suatu fase dimana diperlukan keahlian perancangan untuk elemen komputer yang menggunakan sistem yaitu pemulihan dan program komputer untuk sistem yang baru.

Mustakini menyatakan bahwa “Perancangan sistem menentukan bagaimana suatu sistem akan menyelesaikan apa yang mesti diselesaikan. Tahap ini menyangkut mengkonfigurasikan dari komponen perangkat lunak dan perangkat keras dari suatu sistem sehingga setelah instalasi dari sistem akan benar-benar memuaskan rancangan bangun yang telah ditetapkan pada akhir tahap analisa sistem”.

Adapun tujuan dari tahap perancangan aplikasi adalah sebagai berikut :

- 1) Untuk memenuhi kebutuhan kepada pemakai sistem
- 2) Untuk memberikan gambaran yang jelas dan rancang bangun yang lengkap kepada pemrograman komputer dan ahli teknik yang terlibat.

## **2. Definisi Sistem**

### **a. Pengertian Sistem**

Struktur sistem merupakan unsur-unsur yang membentuk sistem tersebut. Sedangkan proses sistem menjelaskan cara kerja setiap unsur sistem tersebut dalam mencapai tujuan sistem. Setiap sistem merupakan bagian dari sistem lain yang lebih besar dan terdiri dari berbagai sistem yang lebih kecil, yang disebut sebagai subsistem. Dari uraian tersebut dapat diambil kesimpulan bahwa suatu sistem pada dasarnya adalah sekelompok unsur yang erat hubungannya satu dengan yang lain. Yang berfungsi bersama-sama untuk mencapai tujuan tertentu.

Pengertian sistem menurut ( Indrajit, 2014:1 ) “Sistem mengandung arti kumpulan-kumpulan dari komponen-komponen yang dimiliki unsur keterkaitan antara satu dengan lainnya”.

Menurut ( Jogiyanto, 2014:1 ) “Sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu. Sistem ini menggambarkan suatu kejadian-kejadian dan kesatuan yang nyata adalah suatu objek nyata, seperti tempat benda, dan orang-orang yang betul ada dan terjadi”.

Menurut ( Sidharta, 2014:2 ) “Sistem adalah himpunan dari bagian-bagian yang saling berhubungan yang secara bersama mencapai tujuan-tujuan yang sama”.

Menurut ( Maniah dan Hamidin, 2017:1 ) “Sistem dapat didefinisikan sebagai kumpulan dari elemen-elemen berupa data, jaringan kerja dari prosedur-prosedur yang saling berhubungan, sumber daya manusia, teknologi baik hardware maupun software yang saling berinteraksi sebagai satu kesatuan untuk mencapai tujuan/sasaran tertentu yang sama”.

Menurut ( Romney dan Steinbart, 2016:2 ) “Sistem adalah kumpulan sari dua atau lebih komponen yang saling bekerja dan berhubungan untuk mencapai tujuan tertentu. Dia juga berpendapat bahwa perusahaan adalah sebuah sistem yang terdiri dari beberapa departemen yang bertindak sebagai subsistem yang membentuk sistem perusahaan tersebut”.

### **b. Syarat-Syarat Sistem**

Syarat-syarat yang harus dimiliki oleh suatu sistem, yaitu :

- 1) Sistem harus dibentuk untuk menyelesaikan suatu tujuan
- 2) Elemen sistem harus mempunyai rencana yang ditetapkan
- 3) Adanya hubungan diantara elemen sistem
- 4) Unsur dasar dari proses (arus informasi, energi dan material) lebih penting dari elemen sistem.

### c. Karakteristik Sistem

Menurut ( Jogiyanto H.M, 2010:3 ), “Suatu sistem mempunyai karakteristik tertentu, yaitu mempunyai komponen – komponen (*components*), batas sistem (*boundary*), lingkungan luar sistem (*environment*), penghubung (*interface*), masukan (*input*), keluaran (*output*), pengolahan (*process*) dan sasaran (*objective*) atau tujuan (*goal*)”. Dibawah ini merupakan penjelasan dari masing – masing karakteristik tersebut :

#### 1) Komponen

Suatu sistem terdiri dari sejumlah komponen-komponen yang saling berinteraksi, yang artinya saling bekerja sama membentuk satu kesatuan. Komponen sistem terdiri dari komponen yang berupa subsistem atau bagian dari sistem.

#### 2) Batasan Sistem (Boundary)

Batasan sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lain atau dengan suatu sistem dengan sistem yang lain atau dengan lingkungan luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang sebagai suatu kesatuan. Batasan suatu sistem menunjukkan ruang lingkup (*scope*) dari sistem tersebut.

#### 3) Lingkungan luar sistem (*environment*)

Lingkungan luar sistem (*environment*) adalah diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan dapat

bersifat menguntungkan yang harus tetap dijaga dan yang merugikan yang harus dijaga dan dikendalikan, kalau tidak akan mengganggu kelangsungan hidup dari sistem.

4) Penghubung sistem (*interface*)

Penghubung sistem merupakan media penghubung antara satu subsistem dengan subsistem lainnya. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari subsistem ke subsistem lain. Keluaran (*output*) dari subsistem akan menjadi masukkan (*input*) untuk subsistem lain melalui penghubung.

5) Masukkan Sistem (*input*)

Masukkan adalah energi yang dimasukkan kedalam sistem, yang dapat berupa perawatan (*maintenance input*), dna masukkan sinyal (*signal input*). *Maintenance input* adalah energi yang dimasukkan agar sistem dapat beroperasi. *Signal input* adalah energi yang diproses untuk didapatkan keluaran.

6) Keluaran sistem (*output*)

Keluaran sistem adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan.

7) Pengolahan sistem

Suatu sistem menjadi bagian pengolah yang akan merubah masukkan menjadi keluaran. Sistem produksi akan mengolah

bahan baku menjadi bahan jadi, *system* akuntansi akan mengolah data menjadi laporan-laporan keuangan.

8) Sasaran sistem

Suatu sistem pasti mempunyai tujuan (*goal*) atau sasaran (*objective*). Sasaran dari sistem sangat menentukan input yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem.

**d. Klasifikasi Sistem**

Menurut ( Hutahuaen, 2015:6 ) adapun klasifikasi sistem yang diuraikan sebagai berikut :

1) Sistem abstrak dan sistem fisik

Sistem abstrak merupakan sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik, misalnya sistem telogi. Sedangkan sistem fisik diartikan sebagai sistem yang nampak secara fisik sehingga setiap mahluk dapat melihatnya, misalnya sistem komputer.

2) Sistem alamiah dan sistem buatan manusia

Sistem alamiah merupakan sistem yang terjadi melalui proses alam, tidak dibuat oleh manusia, misalnya sistem tata surya, sistem galaksi, sistem reproduksi dan lain-lain. Sedangkan sistem buatan manusia merupakan sistem yang dirancang oleh manusia. Sistem buatan yang melibatkan interaksi manusia, misalnya sistem akuntansi, sistem informasi, dan lain-lain.

### 3) Sistem deterministik dan sistem probabilistik

Sistem deterministik merupakan sistem yang beroperasi dengan tingkah laku yang sudah dapat diprediksi. Interaksi bagian-bagiannya dapat dideteksi dengan pasti sehingga keluaran dari sistem dapat diramalkan, misalnya sistem komputer, adalah contoh sistem yang tingkah lakunya dapat dipastikan berdasarkan program-program komputer yang dijalankan. Sedangkan sistem robabilistik merupakan sistem yang kondisi masa depanya tidak dapat diprediksi karena mengandung unsur probabilitas, misalnya sistem manusia.

### 4) Sistem terbuka dan sistem tertutup

Sistem terbuka merupakan sistem yang berhubungan dan terpengaruh dengan lingkungan luarnya. Lebih spesifik dikenal juga yang disebut dengan sistem terotomasi, yang merupakan bagian dari sistem buatan manusia dan berinteraksi dengan kontrol oleh satu atau lebih komputer sebagai bagian dari sistem yang digunakan dalam masyarakat modern. Sistem ini menerima masukan dan menghasilkan keluaran untuk sub sistem lainnya, misalnya sistem kebudayaan manusia. Sedangkan sistem tertutup merupakan sistem yang tidak berhubungan dan tidak terpengaruh dengan lingkungan luarnya. Sistem ini bekerja secara otomatis tanpa danya campur tangan dari pihak luar. Secara teoritis sistem tersebut

ada, tetapi kenyataannya tidak ada sistem yang benar-benar tertutup, yang ada hanyalah *relatively closed system* (secara relatif tertutup, tidak benar-benar tertutup).

Dari beberapa penjelasan yang sudah disebutkan, dapat disimpulkan bahwa sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan kegiatan atau untuk melakukan sasaran yang tertentu dan saling bekerjasama dengan tujuan yang sama untuk menghasilkan *output* yang sudah ditentukan sebelumnya.

#### e. Tujuan Sistem

Menurut ( Susanto, 2013:23 ) tujuan sistem merupakan target atau sasaran akhir yang ingin dicapai oleh suatu sistem. Agar supaya target tersebut bisa tercapai, maka target atau sasaran tersebut harus diketahui terlebih dahulu ciri-ciri atau kriterianya. Upaya mencapai suatu sasaran tanpa mengetahui ciri-ciri atau kriteria dari sasaran tersebut kemungkinan besar sasaran tersebut tidak akan pernah tercapai. Ciri-ciri atau kriteria dapat juga digunakan sebagai tolak ukur dalam menilai suatu keberhasilan suatu sistem dan menjadi dasar dilakukannya suatu pengendalian.

Dari pengertian diatas dapat ditarik kesimpulan bahwa sistem merupakan kumpulan suatu komponen sistem yang saling

berhubungan satu dengan yang lain untuk mencapai tujuan suatu kegiatan pokok perusahaan atau organisasi.

### **3. Definisi Informasi**

#### a. Pengertian informasi

Informasi merupakan hal yang sangat mendasar yang sangat diperlukan oleh suatu kegiatan dalam pengambilan suatu keputusan agar tidak terjadi kesalahan. Informasi juga dapat diartikan sebagai data yang telah diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi penerima informasi.

Menurut ( Anggraeni dan Irviani, 2017:13 ) menjelaskan bahwa "informasi adalah sekumpulan data atau fakta yang diorganisasi atau diolah dengan cara tertentu sehingga mempunyai arti bagi penerima".

Menurut ( Subtari dalam Trimahardhika dan Sutinah, 2017:250 ), "informasi suatu data yang telah diolah, di klasifikasikan dan diinterpretasikan serta digunakan untuk proses pengambilan keputusan".

#### b. Karakteristik informasi

- 1) Relevan, informasi harus memiliki makna yang tinggi sehingga tidak menimbulkan keraguan bagi yang menggunakannya dan dapat digunakan secara tepat untuk membuat keputusan.

- 2) Andal, suatu informasi harus memiliki keterandalan yang tinggi, informasi yang dijadikan alat pengambilan keputusan merupakan kejadian nyata dalam aktifitas perusahaan.
  - 3) Lengkap, informasi tersebut harus memiliki penjelasan yang rinci dan jelas dari setiap aspek peristiwa yang diukurnya.
  - 4) Tepat waktu, setiap informasi harus dalam kondisi yang update tidak dalam bentuk yang usang, sehingga penting untuk digunakan sebagai pengambilan keputusan.
  - 5) Dapat dipahami, informasi yang disajikan dalam bentuk yang jelas akan memudahkan orang dalam menginterpretasikannya.
- c. Kualitas informasi
- 1) Akurat (*accurate*)

Informasi harus bebas dari kesalahan dan tidak menyesatkan. Akurat juga berarti bahwa informasi harus jelas mencerminkan maksudnya. Informasi harus akurat karena dari sumber informasi sampai ke penerima informasi mungkin banyak mengalami gangguan (*noise*) yang dapat mengubah atau merusak informasi tersebut.
  - 2) Tepat Waktu (*timelines*)

Informasi yang sampai kepada penerima tidak boleh terlambat. Informasi yang sudah usang tidak akan mempunyai nilai lagi, karena informasi merupakan landasan di dalam pengambilan

keputusan. Bila pengambilan keputusan terlambat maka dapat berakibat fatal bagi organisasi.

3) Relevan (*relevance*)

Informasi tersebut mempunyai manfaat untuk pemakaiannya.

Relevansi informasi untuk setiap orang berbeda.

Menyampaikan informasi tentang penyebab kerusakan mesin produksi kepada akuntan perusahaan tentunya kurang relevan.

Akan lebih relevan bila ditujukan kepada ahli teknik perusahaan.

#### **4. Definisi Pelayanan**

a. Pengertian pelayanan

Pelayanan adalah ”suatu usaha untuk membantu menyiapkan apa yang diperlukan seseorang atau orang lain, terhadap kita yang membentuk suatu keterikatan hak dan kewajiban masing-masing pihak”. (Rusydi, 2017).

Menurut (Poewadarmita dalam Hardiyansyah, 2011:10) dalam (Nurhadi, 2016) menjelaskan bahwa secara etimologis, pelayanan berasal dari kata layan yang berarti membantu menyiapkan atau mengurus apa-apa yang diperlukan seseorang, kemudian pelayanan dapat diartikan sebagai perihal atau cara melayani, service atau jasa, sehubungan dengan jual beli barang atau jasa.

Menurut ( Donald dalam Hardiyansyah, 2011:10 ) dalam (Nurhadi, 2016) mengatakan bahwa pelayanan pada dasarnya adalah

merupakan kegiatan atau manfaat yang ditawarkan oleh suatu pihak kepada pihak lain dan pada hakikatnya tidak terwujud serta tidak menghasilkan kepemilikan sesuatu, proses produksinya mungkin juga tidak dikaitkan dengan suatu produk fisik.

(Hardiyansyah, 2011:11) dalam (Nurhadi, 2016) mendefinisikan pelayanan adalah aktifitas yang diberikan untuk membantu, menyiapkan dan mengurus. Baik itu berupa barang atau jasa dari satu pihak kepada pihak yang lain. Istilah lain yang sejenis dengan pelayanan itu adalah pengabdian dan pengayoman.

Menurut (Barata, 2003:30) dalam (Frimayasa, 2017b), pelayanan adalah suatu kegiatan atau urutan kegiatan yang terjadi dalam interaksi langsung antara seseorang dengan orang lain atau mesin secara fisik, dan menyediakan kepuasan pelanggan.

b. Karakteristik pelayanan

Menurut (Simamora, 2001:175) karakteristik pelayanan terdiri atas empat, yaitu :

1) Intangibility (tidak terwujud)

Layanan yang bersifat intangibility artinya tidak dapat dilihat, dirasa, diraba, dicium, dan didengar sebelum dibeli. Seseorang tidak dapat menilai hasil dari layanan sebelum ia menikmatinya sendiri. Untuk mengurangi ketidakpastian, pembeli akan mencari tanda atau bukti dari kualitas jasa tersebut.

2) Inseparability (tidak terpisahkan)

Layanan biasanya dijual terlebih dahulu baru kemudian diproduksi dan dikonsumsi secara bersamaan. Jika seseorang melakukan layanan maka penyediannya adalah bagian dari layanan. Karena klien juga hadir pada saat layanan itu dilakukan, interaksi antara penyedia layanan dan pelanggan merupakan ciri khusus dalam pemasaran layanan.

3) Variability (bervariasi)

Layanan sangat bersifat variabel karena merupakan nonstandardized output, artinya banyak variasi bentuk, kualitas dan jenis, tergantung pada siapa, kapan dan dimana layanan tersebut dihasilkan.

4) Perishability (mudah lenyap)

Layanan merupakan komoditas yang tidak dapat tahan lama dan tidak dapat disimpan. Sehingga dapat dikatakan bahwa jasa dihasilkan pada saat ada permintaan akan jasa tersebut dan permintaan ini tidak dapat ditunda.

c. Dasar-dasar pelayanan

Dalam memberikan pelayanan biasanya seseorang akan memiliki keterampilan dasar dimana hal ini akan membawa seseorang untuk memiliki pedoman dalam melaksanakan tugasnya. Adapun dasar-dasar yang harus di pahami dan dimengerti :

1) Berpakaian dan Berpenampilan Rapih dan Bersih

Karyawan harus mengenakan baju dan celana yang sepadan dengan kombinasi yang menarik. Harus berpakaian bersih tidak kumal dan bagi baju lengan panjang jangan digulung. Gunakan pakaian seragam jika karyawan diberikan pakaian seragam yang telah ditetapkan.

2) Aroma tubuh yang menyegarkan

Hindari seperti bau badan dan bau mulut yang kurang segar, sehingga dapat mengganggu pelayanan. Oleh karena itu, karyawan perlu menggunakan parfum yang baunya juga tidak terlalu menyengat. Demikian juga untuk pengharum mulut dapat menggunakan permen.

3) Percaya diri, bersikap akrab dan penuh dengan senyuman

Dalam melayani pelanggan, karyawan tidak boleh ragu-ragu. Karyawan harus memiliki keyakinan dan percaya diri yang tinggi. Selain itu juga harus bersikap akrab dengan pelanggan dan juga harus murah senyum dengan raut muka yang menarik hati, serta tidak dibuat-dibuat.

4) Menyapa dengan lembut dan berusaha menyebutkan nama jika kenal

Pada saat pelanggan datang karyawan harus segera menyapa terlebih dahulu dan kalau sudah pernah bertemu sebelumnya

usahaakan menyapa dengan menyebutkan namanya, serta menanyakan apa yang dapat kami bantu.

5) Tenang dan tekun mendengarkan setiap pembicaraan

Usahakan pada saat melayani pelanggan dalam keadaan tenang, tidak terburu-buru. Kondisi yang terburu-buru akan membuat suasana menjadi tidak nyaman. Kemudian tunjukkan sikap menghormati tamu, tekun mendengarkan sekaligus berusaha memahami keinginan pelanggannya.

6) Sopan santun

Sopan santun harus ditunjukkan sebelum dan selama proses pelayanan dilakukan.

7) Hormat

Karyawan harus menghormati keinginan dan kemauan pelanggan. Jangan menimbulkan perasaan tidak senang yang membuat pelanggan tersinggung.

8) Berbicara dengan bahasa yang baik dan benar

Dalam berkomunikasi dengan pelanggan gunakan bahasa indonesia yang baik dan benar atau jika menggunakan bahasa daerah atau bahasa asing gunakan secara benar pula. Suara yang digunakan harus jelas dalam arti mudah dipahami dan jangan menggunakan istilah-istilah yang sulit dipahami.

- 9) Bergairah dalam melayani nasabah dan tunjukkan kemampuannya

Dalam melayani pelanggan jangan terlihat lemas, atau kurang semangat. Tunjukkan pelayanan yang prima seolah-olah memang anda sangat tertarik dengan keinginan dan kemauan pelanggan. Kemampuan karyawan pengetahuan produk dalam memberikan informasi ke pelanggan benar-benar meyakinkan sehingga pelanggan merasa puas dengan pelayanan yang diberikan.

- 10) Jangan menyela atau memotong pembicaraan

Usahakan jangan memotong atau menyela pembicaraan. Hindari kalimat bersikap teguran atau sindiran yang dapat menyinggung pelanggan. Usahakan pelanggan yang berbicara lebih banyak, setelah selesai berbicara barulah karyawan melanjutkan.

- 11) Mampu meyakini nasabah serta memberikan kepuasaan

Setiap pelayanan yang diberikan harus mampu meyakinkan pelanggan dengan argumen-argumen yang masuk akal. Karyawan juga harus mampu memberikan kepuasan atas pelayanan yang diberikannya. Biasanya sering kali pelanggan kurang paham tentang sesuatu, maka jelaskan dengan kalimat yang sejelas-jelasnya, sehingga pelanggan merasa yakin dengan produk yang dibeli atau jasa yang digunakan.

12) Jika tidak sanggup, meminta bantuan

Jika ada pertanyaan atau permasalahan yang tidak sanggup dijawab atau diselesaikan oleh karyawan, maka harus meminta bantuan kepada karyawan yang mampu menanganinya. Tujuan nya adalah jangan sampai karyawan memberikan informasi yang salah yang akan berdampak fatal di kemudian hari.

13) Bila belum dapat melayani, beritahukan kapan akan dilayani

Jika pada saat tertentu, karyawan sibuk dan tidak dapat melayani salah satu pelanggan, maka beritahukan kepada pelanggan kapan akan dilayani dengan simpatik. Hal ini dilakukan untuk menghindari pelanggan menunggu tanpa kepastian dan dapat menimbulkan emosi dari pelanggan.

(Kasmir, 2017)

## 5. Definisi Jasa

### a. Pengertian jasa

jasa adalah suatu aktivitas atau tindakan yang tidak berwujud, tidak dapat diraba tetapi dapat diidentifikasi, yang direncanakan dan dilaksanakan untuk memenuhi permintaan dan kepuasan konsumen.

Menurut ( Kotler, 2013:386 ) mengungkapkan bahwa jasa adalah "setiap tindakan atau kinerja yang ditawarkan oleh satu pihak ke pihak lain yang secara prinsip tidak berwujud dan tidak

menyebabkan perpindahan kepemilikan. Produksi jasa dapat terikat atau tidak terikat pada suatu produk fisik”.

Menurut J. Stanton dikutip dari ( Buchari Alma, 2008:243 ) mengungkapkan bahwa jasa adalah ”sesuatu yang dapat diidentifikasi secara terpisah tidak berwujud, ditawarkan untuk memenuhi kebutuhan. Jasa dapat dihasilkan dengan menggunakan benda-benda berwujud atau tidak”.

b. Karakteristik jasa

Jasa memiliki karakteristik unik yang membedakannya dengan barang, yakni (Tjiptono, 2000: 15-18)

1) Tidak berwujud (*intangibles*)

Jasa bersifat tidak terwujud, artinya tidak dapat dilihat, dirasa, diraba, didengar sebelum dibeli. Untuk mengurangi ketidakpastian, konsumen memperhatikan tanda-tanda atau bukti-bukti kualitas jasa tersebut dari tempat, orang, peralatan, materi komunikasi, simbol dan harga yang mereka amati. Oleh karena itu, perusahaan jasa menghadapi tantangan untuk mengelola keterangan atau informasi untuk mewujudkan produk yang tidak berwujud.

2) Tidak dapat dipisahkan (*inseparability*)

Jasa tidak dapat dipisahkan dari proses produksi dan konsumsinya. Jasa biasanya dijual terlebih dahulu, kemudian diproduksi dan dikonsumsi secara bersamaan. Interaksi antara

penyedia jasa dan konsumen mempengaruhi hasil dari jasa tersebut, sehingga perusahaan jasa perlu memperhatikan proses rekrutmen, kompensasi serta pelatihan dan pengembangan karyawan. Selain itu juga memberi perhatian khusus kepada tingkat partisipasi konsumen dalam proses jasa, penyediaan fasilitas jasa dan pemilihan lokasi yang tepat.

3) Keanekaragaman (*variability*)

Jasa bersifat sangat beraneka ragam karena merupakan *non-standardized output*, artinya banyak variasi bentuk, kualitas dan jenis tergantung pada siapa, kapan, dan dimana jasa tersebut dihasilkan. Para pembeli jasa sangat peduli terhadap variabilitas yang tinggi ini dan seringkali mereka meminta pendapat orang lain sebelum memutuskan untuk memilih penyedia jasa.

4) Tidak tahan lama (*perishability*)

Jasa merupakan komunitas tidak tahan lama dan tidak dapat disimpan, dengan demikian bila suatu jasa tidak dapat digunakan maka jasa tersebut akan berlalu begitu saja. Jika permintaan jasa dari konsumen tidak konstan, maka hal ini akan menjadi masalah. Oleh karena itu, perusahaan jasa harus mengevaluasi kapasitasnya untuk menyeimbangkan penawaran dan permintaan.

c. Klasifikasi jasa

Di dalam melaksanakan pemasaran jasa tidak dapat disamakan antara pemasaran suatu jasa dengan jasa lainnya. Karena industri jasa ini sangatlah beragam. Klasifikasi jasa dapat membantu memahami batasan-batasan dari industri jasa dan memanfaatkan pengalaman industri jasa lainnya yang mempunyai masalah dan karakteristik yang sama untuk diaplikasikan dalam suatu bisnis jasa.

Sejauh ini telah banyak pakar yang mengemukakan skema klasifikasi jasa, dimana masing-masing ahli menggunakan dasar pembedaan yang dapat dilakukan berdasarkan tujuh kriteria pokok.

Menurut ( Lovelock, 2012:23 ) adalah sebagai berikut :

1) Segmen pasar

Berdasarkan segmen pasar, jasa dapat dibedakan menjadi jasa yang ditujukan pada konsumen akhir (misalnya taksi, asuransi jiwa, catering dan jasa tabungan) dan jasa bagi konsumen organisasional (biro, periklanan, jasa akuntansi dan jasa konsultasi manajemen)

2) Tingkat keberwujudan

Kriteria ini berhubungan dengan tingkat keterlibatan produk fisik dengan konsumen. Dalam kaitan dengan aspek pemasaran, secara umum dapat dikatakan bahwa semakin

tidak berwujud sebuah jasa, maka semakin sedikit kesamaan antara pemasaran jasa dan pemasaran barang terwujud.

3) Keterampilan penyedia jasa

Berdasarkan tingkat keterampilan penyedia jasa, terdapat dua tipe pokok jasa. Pertama, *professional service* (dosen, konsultan manajemen, konsultan hukum). *Kedua non-professional service*

4) Tujuan organisasi jasa

5) Regulasi

6) Tingkat intensitas karyawan

7) Tingkat kontak penyedia jasa dan konsumen

## 6. Definisi Diagram Alir Data (DAD)

### a. Pengertian Diagram Alir Data (DAD)

Menurut ( Rosa, dan M. Shalahuddin, 2016:70 ) mengemukakan bahwa, “DFD dapat digunakan untuk mempresentasikan sebuah sistem atau perangkat lunak pada beberapa level yang lebih detail untuk merepresentasikan aliran informasi atau fungsi yang lebih detail”. Sedangkan Menurut ( Ladjamudin, 2013:64 ) “Diagram Aliran Data/Data Flow Diagram (DFD) merupakan model dari sistem untuk menggambarkan pembagian sistem ke modul yang lebih kecil”.

Berdasarkan pendapat para ahli diatas maka dapat disimpulkan bahwa Diagram Aliran Data (DAD) adalah alat

analisis terstruktur yang menggunakan arus data pada suatu sistem secara terstruktur dan jelas dengan menggunakan notasi atau simbol dalam diagram arus data, sehingga dapat membantu komunikasi antara analis sistem dengan pemakai sistem (user).

**b. Tingkatan Atau Level Diagram Alir Data (DAD)**

Gambaran suatu sistem menggunakan diagram alir data (DAD) ini mempunyai tahapan-tahapan yang wajib dilakukan sewaktu menggambar diagram alir data (DAD). Adapun tingkatan atau tahapan dalam penggambaran DAD (Muslihudin dan Oktafianto, 2016:48), yaitu:

1) Diagram Konteks

Gambaran suatu sistem menggunakan diagram alir data (DAD) ini mempunyai tahapan-tahapan yang wajib dilakukan sewaktu menggambar diagram alir data (DAD). Adapun tingkatan atau tahapan dalam Diagram konteks menggambarkan dan dapat mewakili seluruh proses yang terdapat di dalam suatu sistem. merupakan tingkatan tertinggi dalam diagram alir data (DAD) dan biasanya diberi nomor nol (0). Semua entitas eksternal yang ditunjukkan pada diagram konteks berikut aliran-aliran data utama menuju dan dari sistem. Diagram ini sama sekali tidak memuat penyimpanan data dan tampak sederhana untuk diciptakan.penggambaran DAD (Muslihudin dan Oktafianto, 2016:48), yaitu:

## 2) Diagram Nol (Diagram Level-1)

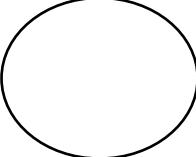
Diagram nol menggambarkan sistem sesuai dengan banyak proses yang terjadi di dalam sistem dan merupakan pemecahan dari diagram konteks. Di dalam diagram ini memuat penyimpanan data. Aliran Data Aliran Data.

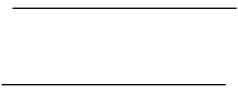
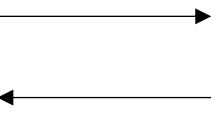
## 3) Diagram Rinci

Diagram nol menggambarkan sistem sesuai dengan banyak proses yang terjadi di dalam sistem dan merupakan pemecahan dari diagram konteks. Di dalam diagram ini memuat penyimpanan data. Aliran Data Aliran Data.

Terdapat empat (4) komponen dalam Diagram Arus Data (DAD) :

**Tabel 2. 1.  
Simbol Diagram Alir Data**

No.	Simbol	Nama Simbol	Keterangan
1		Proses	Suatu proses adalah kegiatan atau kerja yang dilakukan orang, mesin atau komputer dari hasil suatu arus yang masuk kedalam proses untuk dihasilkan arus data yang akan keluar dari proses.

2		Basis Data	Basis data ( <i>database</i> ) digunakan untuk membuat model sekumpulan paket data.
3		Entitas Luar	Terminator mewakili entitas eksternal yang berkomunikasi dalam sistem yang sedang dikembangkan.
4		Data/flow	Suatu alur data digambarkan dengan anak panah, yang menunjukkan arah menuju kedalam dan keluar dari suatu proses.

## 7. Definisi Kamus Data

### a. Pengertian Kamus Data

Menurut ( Kristanto, 2018: 72 ), “Kamus data (data dictionary) adalah kumpulan elemen-elemen atau simbol-simbol yang digunakan untuk membantu dalam penggambaran atau pengidentifikasi setiap field atau file di dalam sistem”.

Kamus Data (KD) atau data dictionary (DD) atau disebut juga dengan istilah system data dictionary adalah catalog fakta

tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi.

Menurut ( Mustakini, 2010:725 ) “Kamus data merupakan katalog fakta tentang data kebutuhan-kebutuhan informasi dari suatu sistem informasi dari suatu sistem informasi”.

**b. Fungsi Kamus Data**

Kamus data ikut berperan dalam perancangan dan pembangunan suatu sistem informasi karena kamus data berfungsi untuk:

- a. Menjelaskan arti aliran data dan menyimpan dalam penggambaran data flow diagram.
- b. Mendeskripsikan komposisi paket data yang bergerak melalui aliran
- c. Mendeskripsikan spesifikasi nilai dan satuan yang relevan terhadap data yang mengalir dalam sistem tersebut

**c. Isi Kamus Data**

Kamus data harus dapat mencerminkan keterangan yang jelas tentang data yang dicatatnya. Isi kamus data adalah sebagai berikut:

- 1) Nama arus data

Kamus data dibuat berdasarkan arus data yang mengalir di DAD, maka perlu menuliskan nama setiap arus data pada

kamus data agar memudahkan siapa saja yang memerlukan penjelasan dan membaca DAD.

2) Alias

Alias atau nama lain dari data dapat dituliskan jika data yang sama mempunyai nama yang berbeda untuk orang atau departemen satu dengan yang lainnya.

3) Bentuk data

Bentuk data yang mengalir ada dua, yaitu: pertama dokumen dasar/formulir yang nantinya dapat digunakan untuk merancang bentuk input system dan yang kedua dokumen hasil cetakan komputer/laporan tercetak yang nantinya akan digunakan untuk merancang bentuk output system.

4) Arus data

Arus data menunjukkan dari mana data mengalir dan kemana data akan menuju.

5) Penjelasan

Untuk lebih memperjelas lagi tentang makna dari arus data yang berisi tentang keterangan-keterangan arus data tersebut.

6) Periode

Periode menunjukkan kapan terjadinya arus data, seperti kapan input data harus dimasukkan ke system, kapan proses dari program harus dilakukan dan kapan laporan dihasilkan.

7) Volume

Untuk menentukan banyaknya rata-rata arus data yang mengalir dalam suatu periode tertentu.

8) Struktur data

Struktur data yang menunjukkan arus data yang dicatat di kamus data terdiri dari item-item data apa saja.

## 8. Definisi Spesifikasi Proses

Spesifikasi Proses menggambarkan kejadian di dalam setiap bubble pada level terbawah pada data flow diagram. Spesifikasi proses mendefinisikan kegiatan yang harus dilakukan untuk mengubah input menjadi output. Spesifikasi proses digunakan untuk mendeskripsikan proses yang terjadi pada level yang paling dasar dalam DFD.

Model ini berfungsi mendeskripsikan apa yang dilakukan ketika masukan ditransformasi menjadi keluaran. Ada berbagai macam tools yang dapat kita gunakan untuk menghasilkan suatu spesifikasi proses: tabel keputusan, Bahasa Inggris terstruktur, pre atau post condition, diagram dan lain sebagainya. Sedangkan kebanyakan analisis sistem mengarah ke Bahasa Inggris terstruktur, anda harus ingat bahwa setiap metode dapat digunakan, selama hal tersebut memuaskan dua keadaan penting :

- 1) Spesifikasi proses harus ditampilkan dalam suatu bentuk atau form yang dapat diverifikasi oleh user dan sistem analis. Kondisi ini tepat untuk alasan ini, dimana kita mengurangi paparan Bahasa Inggris

sebagai sebuah alat spesifikasi hal ini dikenal sebagai ambigu, khususnya apabila menggambarkan tindakan (keoutusan) alternatif dan tindakan berulang (loops). Secara alami, hal tersebut juga cenderung menyebabkan kebingungan yang sangat ketika mengekspresikan bagian-bagian kondisi Boolean(misalkan kombinasi dari operator Boolean AND, OR dan NOT).

- 2) Spesifikasi proses harus ditampilkan dalam suatu bentuk atau form yang dapat mengkomunikasikan secara efektif berbagai keterlibatan berbagai latar belakang pendengar. Sedangkan hal tersebut akan menjadi tipe dari analisis sistem yang menuliskan spesifikasi proses, hal tersebut biasanya menjadi bermacam-macam pendengar dari para pengguna, manager, auditor, personil quality assurance, dan lainnya yang membaca spesifikasi proses.

## **9. Definisi Bagan Terstruktur**

Menurut ( Mustakini, 2010:727 ) “Bagan terstruktur adalah bagan yang digunakan untuk mendefinisikan dan mengilustrasikan organisasi-organisasi dari sistem informasi secara berjenjang dalam bentuk modul dan sub modul”.

Bagan terstruktur (structured chart) digunakan untuk mendefinisikan dan mengilustrasikan organisasi dari system informasi secara berjenjang dalam bentuk modul dan submodul, dengan demikian bagan terstruktur dapat memberikan penjelasan yang lengkap dari

sistem yang dipandang dari elemen data, elemen control, modul dan hubungan antar modulnya.

**Tabel 2. 2.**  
**Simbol Bagan Terstruktur**

No.	Simbol	Nama Simbol	Keterangan
1		Module	Simbol ini menunjukkan suatu modul.
2		Penghubung antar module	Simbol ini digunakan untuk menghubungkan suatu modul dengan modul lainnya.
3		Entitas Luar	Simbol ini menunjukkan suatu data atau elemen kontrol yang dikirimkan dari suatu modul ke modul lainnya. Panah dengan lingkaran kosong menunjukkan data yang dikirimkan, sedangkan panah dengan lingkaran hitam menunjukkan elemen kontrol yang dikirimkan.

## **10. Definisi Database**

### **a. Pengertian Database**

Menurut ( Kristanto, 2018: 79 ) “Database Management System adalah perangkat lunak yang didesain untuk membantu dalam hal pemeliharaan dan utilitas kumpulan data dalam jumlah besar”.

Menurut ( Hutahaean, 2014:50 ) “Basis data merupakan kegiatan sistem program komputer untuk berbagi aplikasi komputer”.

Basis Data terdiri atas 2 kata, yaitu Basis dan Data. Basis kurang lebih dapat diartikan sebagai markas atau gudang, tempat bersarang/berkumpul. Sedangkan Data adalah reseprntasi fakta dunia nyata yang mewakili suatu objek seperti manusia (pegawai, siswa, pembeli, pelanggan), barang, hewan, peristiwa, konsep, keadaan, dan sebagainya, yang diwujudkan dalam bentuk angka, huruf, symbol, teks, gambar, bunti, atau kombinasinya.

Basis data (Database) adalah “Himpunan kelompok data (arsip) yang saling berhubungan yang diorganisasi sedimikian rupa agar kelak dapat di manfaatkan kembali dengan cepat dan mudah”. Prinsip utama Basis Data adalah pengaturan data/arsip. Dan tujuan utamanya adalah kemudahan dan kecepatan dalam pengambilan kembali data/arsip. Perbedaannya hanya terletak pada media penyimpanan yang digunakan.

Dari penjelasan tersebut dapat disimpulkan bahwa basis data merupakan sekumpulan data yang saling berhubungan dan disimpan dalam suatu tempat yang biasanya diimplementasikan dalam bentuk tabel yang tersusun atas bagian yang disebut field dan record. Sebuah field merupakan kesatuan terkecil dan informasi dalam sebuah database, sedangkan record adalah sekumpulan field yang saling berhubungan atau saling berkaitan. Database memiliki tiga macam model data yaitu hirarki, jaringan, dan relasional.

#### **b. Operasi-operasi Dasar Database**

Operasi-operasi dasar yang dapat dilakukan dalam membuat sebuah database adalah pembuatan basis data baru (create database), penghapusan basis data (drop database), pembuatan file/table baru (create table), penghapusan file/table (drop table), penambahan / pengisian data dari sebuah file/table (insert), pengambilan data dari sebuah file/table (retrieve/search), pengubahan data dari sebuah file/table (update), serta penghapusan data dari sebuah file/table (delete).

Operasi yang berkaitan dengan pembuatan objek (basis data dan tabel) merupakan operasi awal yang hanya dilakukan sekali dan berlaku seterusnya. Sedang operasi-operasi yang berkaitan dengan isi tabel (data) merupakan operasi rutin yang berlangsung berulang-ulang dan karena itu operasi-operasi inilah yang tepat

mewakili aktivitas pengelolaan dan pengolahan (processing) data pada basis data.

## **11. Definisi Normalisasi**

Menurut ( Connolly dan Begg, 2010:416 ), “Normalisasi adalah sebuah teknik yang menghasilkan suatu kumpulan relasi dengan property yang diinginkan dengan memberikan suatu kebutuhan data pada perusahaan”. Sedangkan menurut ( Indrajani, 2015:7 ), “Normalisasi adalah teknik dengan melakukan sebuah pendekatan bottom-up yang digunakan dalam membantu mengidentifikasi hubungan”.

Berdasarkan pendapat para ahli diatas maka dapat disimpulkan bahwa normalisasi adalah proses yang dilakukan untuk mengidentifikasi hubungan yang menghasilkan suatu kumpulan relasi yang diinginkan. Tujuan dari normalisasi adalah untuk menghasilkan database yang baik sehingga mempermudah dalam penyimpanan data.terdapat beberapa tahapan dalam normalisasi berikut adalah tahapan-tahapan dalam normalisasi :

- a. Bentuk tidak normal (unnormalized relation), bentuk ini merupakan kumpulan data yang akan direkam tidak ada keharusan mngikuti suatu format tertentu, data yang dicatat mempunyai ciri berulang (redundant).
- b. First Normal Form (1NF), Adalah sebuah relasi yang terdiri dari perpotongan dari setiap baris dan kolom berisi satu dan hanya satu buah nilai saja. Aturan dari 1NF yaitu:

- 1) Tidak ada atribut multi-value, atribut komposit atau kombinasinya.
  - 2) Mendefinisikan atribut kunci
  - 3) Setiap atribut dalam table tersebut harus bernilai atomic (tidak dapat dibagi-bagi lagi)
- c. Second Normal Form (2NF), suatu relasi sudah berada pada bentuk normal kedua jika relasi tersebut sudah memenuhi bentuk normal pertama, dan atribut yang bukan key sudah tergantung penuh terhadap primary keynya.
- d. Third Normal Form (3NF), Adalah relasi yang berada dalam bentuk 1NF dan 2NF di mana tidak ada lagi atribut yang bukan primary key yang bergantung secara transitif kepada primary key.

Aturan dai 3NF yaitu :

- 1) Sudah berada dalam bentuk normal kedua (2NF)
- 2) Tidak ada ketergantungan transitif (dimana atribut bukan kunci tergantung pada atribut bukan kunci lainnya).

## 12. Definisi ERD

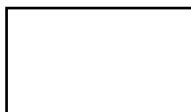
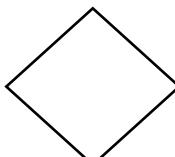
- a. Definisi Relationship Diagram (*ERD*)

Menurut ( Yakub, 2012:60 ) “entity relationship diagram (ERD) untuk mendokumentasikan data perusahaan dengan mengidentifikasi jenis entitas (entity) dan hubungannya ERD merupakan suatu model jaringan yang menggunakan susunan data yang disimpan pada sistem secara abstrak”. Sedangkan menurut (

Sukamto, Ariani Rosa, dan M. Shalahuddin, 2016:50 ) “entity relationship diagram digunakan untuk permodelan basis data relasional. Sehingga jika penyimpanan basis data menggunakan Object Oriented Database Management System (OODBMS) maka perancangan basis data tidak perlu menggunakan ERD”.

b. Komponen Penyusun ERD (*Entity Relationship Diagram*)

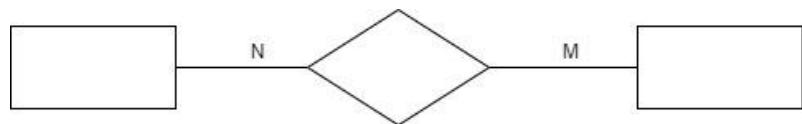
**Tabel 2. 3.  
Simbol ERD**

Simbol	Nama	Keterangan
	Entitas	Kumpulan beberapa objek yang dapat dibedakan atau diidentifikasi secara unik
	Atribut	Sifat atau karakter suatu entitas
	Relationship	Hubungan yang terjadi antara entitas dengan entitas lainnya
	Link	Merupakan garis yang menghubungkan antara banyak cabang atau pemakaian angka

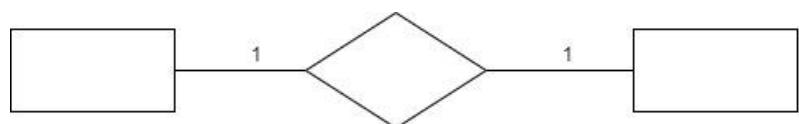
### 13. Definisi Kardinalitas

Menurut ( Muhibib, 2013:10 ) “Kardinalitas relasi menunjukkan jumlah maksimum entitas yang dapat berelasi dengan entitas pada himpunan entitas yang lain”.

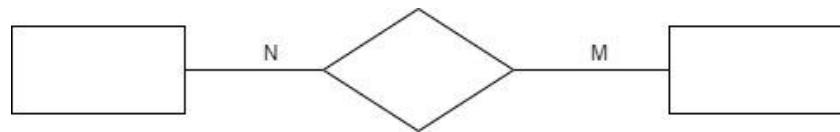
Kardinalitas yang terjadi diantara dua himpunan entitas (misal A dan B) dapat berupa :



- 1) Satu ke Satu (One to One) yang berarti setiap entitas pada himpunan entitas A dapat berhubungan dengan paling banyak dengan satu entitas pada himpunan entitas B, dan begitu juga sebaliknya, setiap entitas pada himpunan entitas B berhubungan dengan paling banyak dengan satu entitas pada himpunan entitas A.
- 2) Satu ke Banyak (One to Many) yang berarti setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B, tetapi tidak sebaliknya, dimana setiap entitas pada himpunan entitas B berhubungan dengan paling banyak dengan satu entitas pada himpunan entitas A.



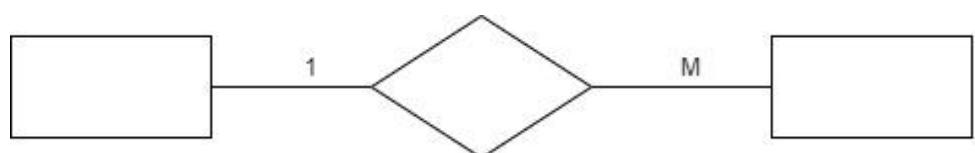
- 3) Banyak ke Satu (Many to One) Setiap entitas pada himpunan entitas A berhubungan dengan paling banyak dengan satu entitas pada satu entitas himpunan entitas B, tetapi tidak sebaliknya, dimana setiap entitas pada himpunan entitas A berhubungan dengan paling banyak satu entitas pada himpunan entitas B.
- 4) Banyak ke Banyak (Many to Many) yang berarti setiap entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B, dan demikian juga sebaliknya, dimana setiap entitas pada himpunan B dapat berhubungan dengan banyak entitas pada himpunan entitas A.



#### **14. Rancangan Dialog**

- 1) Rancangan Layar

Rancangan layar merupakan rancangan bangunan dari percakapan antar pemakai (user) dengan komputer, percakapan ini dapat terjadi pada proses memasukan data ke sistem. Menampilkan output informasi kepada user atau dapat keduanya, layar dialog harus dibentuk sedemikian rupa sehingga informasi, instruktur dan bantuan-bantuan selalu ditampilkan pada area yang dimaksud.



## 2) Rancangan Form Input

Rancangan form input adalah suatu rancangan yang dibuat untuk tempat dimana terjadinya semua proses input yang dibutuhkan dalam proses.

## 3) Rancangan Form Output

Rancangan form output adalah suatu rancangan dimana akan ditampilkan hasil dari suatu proses yang dilakukan.

Berdasarkan keterangan tersebut dapat diambil kesimpulan mengenai rancangan dialog, yang merupakan suatu rancangan dimana dilakukan semua proses input untuk diproses dan menghasilkan rancangan output dari suatu proses.

## 15. Definisi JAVA

### a. Sejarah Singkat JAVA

Menurut ( Dwi Hastuti dan Yusril Arief, 2018: 1 ), Pencipta bahasa pemrograman java adalah James Gosling. Beliau menciptakan bahasa ini saat masih bergabung di Sun Microsystem yang saat ini sudah menjadi bagian dari oracle dan dirilis tahun 1995. Bahasa java dijalankan menggunakan JVM atau Java virtual Machine, ini yang menyebabkan java lebih fleksibel dan bisa dijalankan di System operasi Windows, Mac OS atau Linux. Sesuai dengan slogannya yaitu “Write once, run anywhere”.

Bahasa tersebut haruslah bersifat multiplatform dan tidak tergantung pada vendor yang memanufaktur chip tersebut. Pada

awalnya, bahasa ini disebut Oak, tapi kemudian diubah menjadi Java pada tahun 1995 karena nama Oak telah dijadikan hak cipta dan digunakan sebagai bahasa pemrograman lain.

Sejak pembuatan Oak pada musim gugur tahun 1992 hingga diumumkan ke publik pada musim semi tahun 1995, banyak yang orang terlibat dalam desain dan evolusi bahasa ini. Bill Joy, Arthur Van Hoff, Jonathan Payne, Frank Yellin, dan Tim Lindholm merupakan kontributor kunci yang mematangkan prototipe aslinya. Pada saat ini, Java banyak digemari oleh para programmer karena open source dan dapat dikembangkan ke pemrograman berbasis internet sesuai dengan kebutuhan masa atau zaman.

### **b. Pengertian JAVA**

Sebagai sebuah bahasa pemrogram, java dapat membuat seluruh bentuk aplikasi, dekstop, web dan lainnya, sebagaimana dibuat dengan menggunakan bahasa pemrograman konvensional yang lain.

Menurut ( Supardi, 2010:1 ) “Java merupakan pemrograman yang dikembangkan dari bahasa pemrograman C++, sehingga bahasa pemrograman ini seperti bahasa C++”.

Dari beberapa penjelasan yang sudah disebutkan, dapat disimpulkan bahwa java adalah bahasa pemrograman yang berorientasi objek (OOP) dan dapat dijadikan pada berbagai platform sistem operasi. Perkembangan java tidak hanya terfokus

pada satu sistem operasi, tetapi dikembangkan untuk berbagai sistem operasi dan bersifat open source.

Berdasarkan white paper resmi dari SUN, java memiliki karakteristik berikut :

1) Sederhana (*Simple*)

Bahasa pemrograman java menggunakan sintaks mirip dengan C++ namun sintaks pada java telah banyak diperbaiki terutama menghilangkan penggunaan pointer yang rumit dan multiple inheritance. Java juga menggunakan automatic memory allocation dan memory garbage collection.

2) Berorientasi Objek (*Object Oriented*)

Java menggunakan pemrograman berorientasi objek yang membuat program dapat dibuat secara modular dan dapat dipergunakan kembali.

3) Terdistribusi (*distributed*)

Java dibuat untuk membuat aplikasi terdistribusi secara mudah dengan adanya libraries networking yang terintegrasi pada Java.

4) *Interpreted*

Program java dijalankan menggunakan interpreter yaitu java virtual machine (JVM). Hal ini menyebabkan source code java yang telah dikompilasi menjadi java bytecodes dapat dijalankan platform berbeda-beda.

5) *Secure*

Sebagai bahasa pemrograman untuk aplikasi internet dan terdistribusi, java memiliki beberapa mekanisme kemanan untuk menjaga aplikasi tidak digunakan untuk merusak sistem komputer yang menjalankan aplikasi tersebut.

6) *Architecture Neutral*

Program java merupakan platform independent. Program cukup mempunyai satu buah versi yang dapat dijalankan pada platform berbeda dengan java virtual machine.

7) *Portable*

Source code maupun program java dapat dengan mudah dibawa ke platform yang berbeda-beda tanpa harus dikompilasi ulang.

8) *Perfomance*

Performance pada java sering dikatakan kurang tinggi. Namun performance java dapat ditingkatkan menggunakan kompilasi java lain seperti buatan inprise, microsoft ataupun Sysmantec yang menggunakan Just In Time.

9) *Multithread*

Java mempunyai kemampuan untuk membuat suatu program yang dapat melakukan beberapa pekerjaan secara sekaligus dan simultan.

### c. Keunggulan JAVA

Menurut ( Nofriadi, 2015:1 ) “Kelebihan java dari bahasa pemrograman yang lain adalah bisa dijalankan diberbagai jenis sistem operasi sehingga dikenal juga bahasa pemrograman multiplatform, bersifat pemrograman berorientasi object (PBO), memiliki library yang lengkap”.

Dari beberapa penjelasan yang sudah disebutkan, dapat disimpulkan bahwa keunggulan java yaitu sederhana (simple). Java juga lebih mudah dan Java sangat membantu terutama jika anda mengembangkan software berskala besar, multi platform, dan multi interface. Ini dikarenakan Java memiliki standar yang komprehensif, integral dan jauh lebih terorganisasi dibanding bahasa pemrograman lain.

## 16. Perangkat Lunak yang Digunakan

### a. NetBeans IDE

Pengembangan Netbeans diawali dari Xelfi sebuah proyek tahun 1997 dibawah bimbingan fakultas Matematika dan Fisika, Universitas Charles Praha, sebuah perusahaan kemudian dibentuk untuk proyek tersebut dan menghasilkan versi komersial Netbeans IDE hingga kemudian dibeli oleh Sun Microsystem pada tahun 1999, Sun Microsystem menjadikan Netbeans open source pada bulan Juni tahun 2000, sejak itu komunitas Netbeans terus berkembang.

Netbeans IDE adalah IDE open source yang ditulis sepenuhnya dengan bahasa pemrograman java menggunakan platform Netbeans. Netbeans IDE mendukung pengembangan semua tipe aplikasi java (J2SE, web, EJB, dan aplikasi mobile). Fitur lainnya adalah sistem proyek berbasis Ant, control versi dan repactoring.

Adapun kelebihan Netbean IDE antara lain:

- 1) Mendukung berbagai bahasa pemrograman seperti java C/C++, php, berjalan pada multiflatform sistem operasi termasuk windows, linux, Mac OS, Solaris.
- 2) Berfungsi untuk memberi pengembangan aplikasi mobile menggunakan bahasa java.
- 3) Mendukung untuk memberi pengembangan aplikasi web menggunakan php.
- 4) Terdapat banyak modul untuk mengembangkan lebih lanjut.
- 5) Merupakan produk free tanpa ada batasan penggunaanya.
- 6) Merupakan produk open source atau kode sumber terbuka.

b. Xampp

Menurut ( Nugroho, 2013:1 ), “XAMPP adalah paket program web lengkap yang dapat Anda pakai untuk belajar pemrograman web, khususnya PHP dan MySQL”. Sedangkan menurut ( Buana, 2014:4 ), “XAMPP adalah perangkat lunak

opensource yang diunggah secara gratis dan bisa dijalankan di semua operasi seperti windows, linux, solaris, dan mac”.

XAMPP merupakan singkatan dari X yang dinaggap sebagai empat sistem operasi (Windows, Linux, Mac OS, dan Solaris), Apache, MySQL, PHP, perl. XAMPP merupakan tool yang menyediakan paket perangkat lunak kedalam sebuah paket. Dalam paketnya sudah terdapat Apache atau web server, MySQL atau database, PHP atau scripting, perl, FTP server, phpMyadmin dan berbagai pustaka baru lainnya. Dengan menginstal XAMPP maka tidak perlu lagi melakukan instalasi dan konfigurasi web server Apache, PHP, dan MySQL secara manual.

c. MySQL

MySQL atau SQL yang merupakan kepanjangan dari Structured Query Languag. SQL merupakan bahasa terstruktur yang khusus didefinisikan oleh American National Standards Institute (ANSI) pada tahun 1996. MySQL dibuat dan dikembangkan oleh MySQL AB yang berada di Swedia. MySQL dapat digunakan untuk membuat dan mengelola database beserta isinya. MySQL dimanfaatkan untuk menambahkan, mengubah dan menghapus data yang berada dalam database.

MySQL merupakan sistem manajemen database yang bersifat relational, artinya data-data yang dikelola dalam database akan diletakan dalam beberapa tabel yang terpisah sehingga

manipulasi data akan menjadi lebih cepat. MySQL juga dapat digunakan untuk mengelola database mulai dari yang kecil sampai data yang sangat besar. MySQL juga dapat menjalankan perintah-perintah Structured Query Languag (SQL) untuk mengelola database-database yang ada didalamnya. Untuk mengaktifkan MySQL di lingkungan windows maka harus dijalankan server MySQL. Bahasa yang digunakan pada database yaitu:

1) DDL (*Dual Definition Language*)

Pola database di spesifikasikan dengan set definisi yang di ekspresikan dengan satu bahasa khusus. Contoh perintah : *create, table, alter table, dan drop table.*

2) DML (*Data Manipulation Language*)

Bahasa yang memperbolehkan pemakai untuk akses atau manipulasi data yang telah di organisasikan sebelumnya dalam model yang tepat. Contoh perintah : *select, insert, update, delete. PhpMyAdmin*

d. PhpMyAdmin

Menurut ( Nugroho, 2013:71 ), “phpMyAdmin adalah tools yang dapat digunakan dengan mudah untuk memanajemen database MySQL secara visual dan Server MySQL, sehingga kita tidak perlu lagi harus menulis query SQL setiap akan melakukan perintah operasi database”. Tools ini cukup populer, Anda dapat

mendapatkan fasilitas ini ketika menginstal paket triad phpMyAdmin, karena termasuk dalam xampp yang sudah di instal.

Menurut ( Buana, 2014:2 ) “phpMyAdmin adalah salah satu aplikasi yang digunakan untuk memudahkan dalam melakukan pengelolaan database MySQL. phpMyAdmin merupakan aplikasi web yang bersifat opensource.”

e. iReport

Menurut ( Maya, 2015:197 ) “IReport adalah tools yang memudahkan anda dalam pembuatan laporan. IReport biasanya digunakan dalam bahasa pemograman java. Tools ini bersifat open source, sehingga dapat dengan bebas digunakan tanpa perlu membeli esensi untuk mengembangkan dengan menghubungkan database MySQL ke java”.

## **B. Penelitian Yang Relevan**

Berikut adalah contoh-contoh penelitian yang relevan dalam bentuk skripsi dan jurnal yang penulisannya membahas tentang sistem-sistem informasi pengelolahan data, serta dijelaskan secara singkat tujuan dari penulisan tersebut, Untuk perbandingan dengan penelitian ini, penulis mencoba mempelajari beberapa skripsi dan jurnal yaitu :

### **1. Skripsi pertama**

Nama : Rexi Andriansyah  
Npm : 12163908  
Tahun : 2019  
Fakultas : Teknologi Informasi Universitas Bina Sarana Informatika  
Judul Skripsi : Perancangan Program Pelayanan Jasa Laundry Pada BWS Green Laundry Bekasi  
Tujuan : Penelitian dilakukan untuk merancang sebuah Program untuk membantu proses penyimpanan maupun proses pencatatan pelayanan serta mudahnya dalam pencarian data yang telah tercatat dengan adanya suatu sistem yang terkomputerisasi.

## **2. Skripsi kedua**

Nama : Muhammad Ramadhan  
Npm : 12143043  
Tahun : 2018  
Fakultas : Manajemen Informatika AMIK BSI Bogor  
Judul Skripsi : Perancangan Sistem Informasi Pelayanan Jasa Laundry Pada King Laundry Bogor  
Tujuan : Penelitian dilakukan untuk merancang sebuah Program agar permasalahan yang dihadapi dapat terselesaikan dengan dibuatnya suatu sistem yang sudah terkomputerisasi agar bisa menangani masalah yang terjadi dan kesalahan yang tidak diinginkan, untuk itu produktivitas kerja menjadi lebih efektif dan efisien.

## **3. Skripsi ketiga**

Nama : Bobby Perdana  
Npm : 1410501003  
Tahun : 2018  
Fakultas : Ilmu Komputer  
Judul Skripsi : Perancangan Sistem Informasi Pelayanan Jasa Laundry Berbasis Web Pada Dian Laundry  
Tujuan : Penelitian ini bertujuan untuk menganalisis sistem

yang berjalan dan membuat perancangan sistem informasi Pelayanan Jasa Laundry Berbasis Web pada Dian Laundry, agar menjalankan transaksi dengan efektif dan sistematis, sehingga sangat mendukung orientasi pelayanan usaha terhadap konsumen.

#### **4. Jurnal pertama**

ISSN : 2527-5321  
2527-5941

Nama : Erna Susanti<sup>1</sup>, Alusyanti Primawati<sup>2</sup>, Adhi Susano<sup>3</sup>

Nama Jurnal : Seminar Nasional Riset da Teknologi

Edisi : 14 Januari 2021

Universitas : Indraprasta PGRI

Judul : Perancangan Sistem Informasi Pelayanan Jasa Zifa Laundry Berbasis Java

Situs : <http://www.proceeding.unindra.ac.id/index.php/Semnaristek/article/download/4878-8804>

#### **5. Jurnal kedua**

ISSN : 2527-5321  
2527-5941

Nama : Manin<sup>1</sup>, Rahmawati<sup>2</sup>

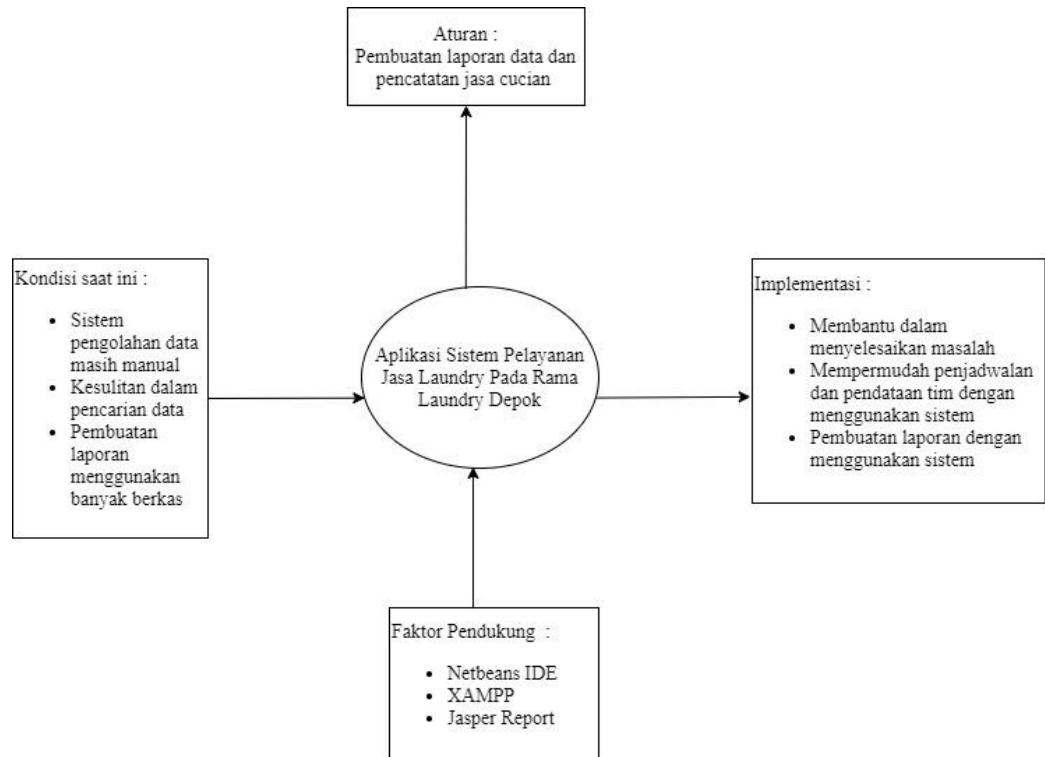
Nama Jurnal : Seminar Nasional Riset dan Teknologi 2021

Edisi : 14 Januari 2021

Universitas : Indraprasta PGRI  
Judul : Perancangan Sistem Aplikasi Laundry Pada Dapoer  
Laundry Ajip 26 Berbasis Java  
Situs : [http://www.proceeding.unindra.ac.id/index.php/  
Semnaristek/article/download/4910-8868](http://www.proceeding.unindra.ac.id/index.php/Semnaristek/article/download/4910-8868)

## C. Kerangka Berfikir

Untuk memudahkan dalam penelitian, berikut kerangka berfikir yang penulis gunakan:



**Gambar 2. 4.**  
**Kerangka Berfikir.**  
Sumber : Penulis (2021)

**Keterangan :**

### 1. Aturan

Berisikan tentang suatu syarat dalam pencapaian hasil yang ditujukan dalam perancangan suatu sistem, dimana dalam pencapaian hasil tersebut harus semaksimal mungkin dapat diterima dan digunakan oleh pengguna dalam Pelayanan Jasa Laundry pada Rama Laundry Depok.

## **2. Aplikasi**

Berisikan tentang materi yang diambil untuk dijadikan bahan penelitian.

## **3. Kondisi saat ini**

Berisikan tentang masalah-masalah yang sebelumnya sudah ada dalam penelitian.

## **4. Implementasi**

Berisi tentang cara penyelesaian masalah-masalah yang sudah ada menjadi suatu kekuatan dalam perancangan aplikasi.

## **5. Faktor pendukung**

Dalam hal ini berisi tentang faktor pendukung dalam perancangan aplikasi, faktor tersebut berupa kemudahan dalam mengakses, bahasa pemrograman serta cara penyampaian dan penggunaan suatu aplikasi.

## **BAB III**

### **METODE PENELITIAN**

#### **A. Waktu dan Tempat Penelitian**

##### **1. Waktu Penelitian**

Penelitian ini berlangsung selama 5 (lima) bulan dimulai dari bulan Maret 2021 sampai dengan Juli 2021. Adapun keuntungan yang diperoleh dalam penelitian adalah untuk mempermudah pengolahan data pada Rama Laundry, agar dapat merencanakan dalam waktu yang ditentukan untuk melengkapi data penulisan tugas akhir. Untuk lebih jelasnya dapat dilihat pada **Tabel 3.1** yang menjelaskan tahapan penelitian berdasarkan pada waktu dan kegiatan yang dilakukan selama proses penelitian sebagai berikut :

**Tabel 3.1**  
**Tabel Waktu Penelitian**

No	Kegiatan	Maret 2021				April 2021				Mei 2021				Juni 2021				Juli 2021			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	Analisis Kebutuhan																				
2	Studi Kepustakaan																				

## **2. Tempat Penelitian**

Penelitian dilaksanakan pada Rama Laundry , Jalan Kenanga Bojong No. 22c Rt. 007/019, Kota Depok, Jawa Barat.

## B. Desain Penelitian

Dalam melakukan penelitian diperlukan suatu metode penelitian yang sesuai dengan pokok permasalahan yang akan diteliti, sedangkan metode penelitian itu sendiri adalah “suatu metode yang digunakan dalam penelitian yang dapat berbentuk metode penelitian survei, exposit facto, action research (penelitian tidakan), evaluasi, dan sejarah”.

Penelitian merupakan suatu proses yang bertujuan untuk menemukan, mengembangkan, dan menguji suatu pengetahuan dengan cara mengumpulkan, mencatat, dan menganalisis informasi data yang dilakukan dengan sabar, hati-hati, terencana dan sistematis serta berdasarkan ilmu pengetahuan.

Dari beberapa definisi diatas maka dapat disimpulkan metodologi penelitian adalah metode yang digunakan mengetahui hasil dari penelitian dengan cara mengumpulkan, mencatat dan menganalisis informasi daya yang ada.

Praswoto (2001:65) menyatakan bahwa “Metode grounded (grounded research) yaitu suatu metode penelitian berdasarkan pada fakta dengan tujuan dapat menetapkan konsep, mengembangkan teori, pengumpulan dan analisis data dalam waktu yang bersamaan”.

Sugiyono (2012:145) menyatakan bahwa “Observasi merupakan suatu proses yang tersusun dari berbagai proses biologis dan psikologis. Dua diantara yang terpenting adalah proses-proses pengamatan dan ingatan”.

Dalam riset ini data merupakan sumber teori atau teori berdasarkan data. Penulis bukan hanya mencari dan mengumpulkan data tetapi juga langsung melakukan klasifikasi terhadap data tersebut, mengolah dan menganalisa data, membangun hipotesis menjadi teori.

## **C. Metode Pengumpulan Data**

Metode pengumpulan data yang dilakukan oleh penulis untuk mendapatkan data-data serta informasi mendukung penyempurnaan hasil dari penelitian ini antara lain adalah :

### **1. Studi Kepustakaan**

Pengumpulan data dan informasi dari kutipan-kutipan berbagai buku, peraturan perundang-undangan, serta hasil laporan dan bahan lainnya yang berkaitan dengan penelitian ini. Dari berbagai referensi tersebut dapat diambil teori yang dapat dijadikan landasan untuk menganalisa masalah yang ditemukan dalam penelitian.

### **2. Studi Lapangan**

#### **a. Wawancara**

Suatu cara pengumpulan data yang digunakan untuk memperoleh informasi langsung dari sumbernya. Wawancara dilaksanakan secara lisan dalam pertemuan tatap muka secara individu.

Wawancara dilakukan pada tanggal 23 Februari 2021, 16.00 WIB dengan Bapak Dani Subardja selaku Pemilik Laundry pada Rama Laundry. Dalam wawancara ini, penulis memberikan beberapa point pertanyaan diantaranya :

- 1) Bagaimana sejarah berdirinya Rama Laundry ini?
- 2) Apa visi dan misi Rama Laundry ?

- 3) Bagaimana struktur organisasi pada Rama Laundry ?
- 4) Bagaimana sistem pencatatan dan transaksi pada Rama Laundry ?

Beberapa point pertanyaan yang diberikan sangat membantu dalam proses pengumpulan data bagi penulis.

#### **b. Survey**

Survey merupakan metode untuk memperoleh data yang ada pada saat penelitian. Metode survey ini penulis lakukan untuk melihat langsung terhadap penerapan sistem pencatatan jenis jasa cucian pada Rama Laundry.

#### **c. Observasi**

Pengertian observasi ialah metode atau cara-cara yang menganalisis dan mengadakan pencatatan secara sistematis mengenai tingkah laku dengan melihat atau mengamati individual atau kelompok secara langsung. Observasi dilaksanakan secara langsung dilaksanakan Rama Laundry, Jalan Kenanga Bojong No. 22c Rt. 007/019, Kota Depok, Jawa Barat.

Observasi yang dilakukan adalah mempelajari dan mengamati sistem kerja yang berjalan sebelumnya dimana mencari titik permasalahan yang ada. Agar dapat menemukan titik masalah yang dapat diselesaikan pada sistem yang baru.

#### **d. Literatur**

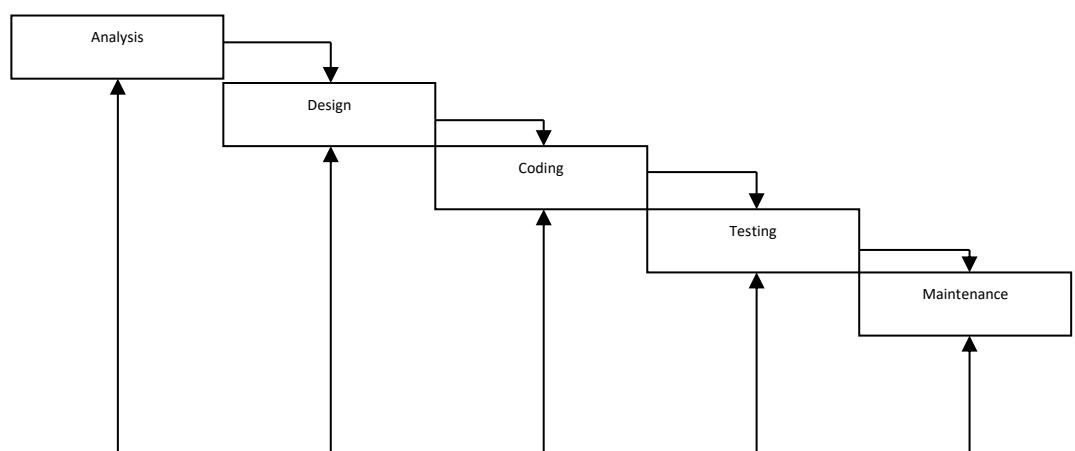
Peneliti melakukan pencarian dan mengumpulkan literatur-literatur dari referensi yang dapat menunjang dalam Aplikasi Sistem Pelayanan Jasa Laundry pada Rama Laundry.

#### **e. Dokumentasi**

Peneliti melakukan pengumpulan data dengan cara melihat dan mencatat data jenis jasa yang dilakukan pada cucian di laundry.

### **D. Langkah-Langkah Pengembangan Sistem**

Perancangan sebuah aplikasi atau sistem, memerlukan metode-metode atau langkah-langkah dalam pembangunan atau pengembangan sistem. Metode waterfall. menurut Saputra (2012) merupakan metode pengembangan perangkat lunak yang secara umum dilakukan oleh para peneliti sistem, dengan melalui beberapa tahapan penelitian yaitu :



**Gambar 3.1.**  
**Metode Waterfall.**  
Sumber : Saputra (2012)

Keterangan :

### **1. Analysis**

Tahap ini merupakan tahap dalam mencari informasi sebanyak-banyaknya mengenai sistem yang diteliti dengan melakukan metode-metode pengumpulan data sehingga ditemukan kelebihan dan kekurangan sistem serta user requirement. Tahap ini juga dilakukan untuk mencari pemecah masalah dan menganalisa bagaimana sistem akan dibangun untuk memecahkan masalah pada sistem sebelumnya.

### **2. Design**

Tahap ini merupakan tahapan perancangan sistem yang didalamnya dilakukan pemodelan sistem dengan use case, relasi tabel, diagram konteks, activity diagram, sequence diagram.

### **3. Coding**

Tahap ini merupakan tahapan dalam pengimplementasian sistem yang sudah dirancang dan dilakukan pengujian secara unit, agar dapat mengetahui kesalahan-kesalahan yang terdapat dalam sistem dan segera dilakukan perbaikan.

### **4. Testing**

Tahap ini merupakan tahap pengujian sistem secara keseluruhan. Tahap ini sistem yang akan dikembangkan menggunakan teknik pengujian black box.

## **5. Maintenance**

Tahap ini merupakan tahapan penggunaan sistem oleh user yang didalamnya harus ada pemeliharaan sistem untuk menjaga proses operasional sistem dan memungkinkan untuk dilakukan pengembangan sistem di kemudian hari.

## **BAB IV**

### **ANALISIS SISTEM BERJALAN DAN RANCANGAN SISTEM YANG DIUSULKAN**

#### **A. Profil Rama Laundry**

Rama Laundry berdiri sejak tahun 2018, laundry ini berada di kawasan Depok-Jawa Barat yang beralamat di Jl. Kenanga Bojong No. 22c Rt. 007 Rw. 019 Abadijaya, Kec. Sukmajaya, Kota Depok, Jawa Barat 16417. Rama Laundry ini terletak di tempat yang strategis, karena berdekatan atau tidak jauh dari pusat kota Depok.

Rama Laundry buka mulai pukul 09.00 pagi dan tutup pukul 20.00 malam. Rama Laundry adalah laundry yang menyediakan banyak jasa cucian mulai dari cuci baju, sepatu, selimut dan lain-lain.

Demi menjamin kepuasan pelanggan, Rama Laundry terus berupaya untuk meningkatkan mutu pelayanannya baik dari segi *service*, maupun kualitas produk-produk yang disediakan.

#### **B. Visi dan Misi**

Visi dan Misi yang dimiliki oleh Rama Laundry adalah :

##### **1. Visi**

Menjadi pengembang usaha Laundry yang berkualitas tinggi.

## **2. Misi**

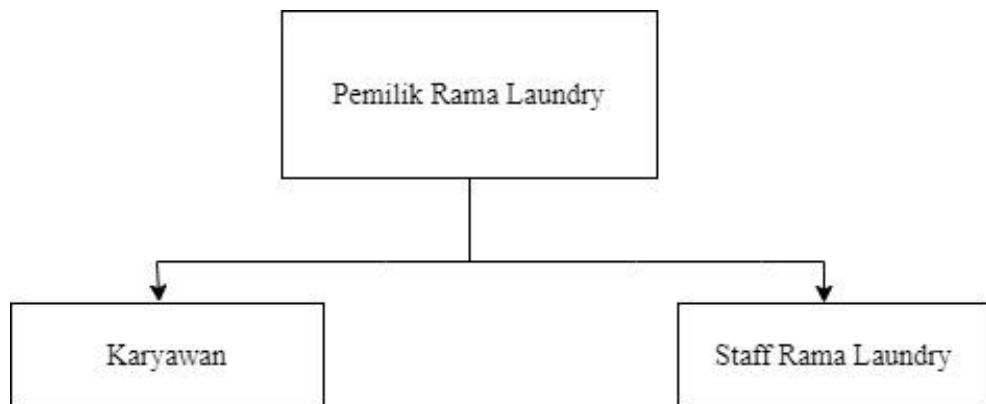
- a. Secara keseluruhan menyedikan jasa yang berkualitas tinggi serta memenuhi kebutuhan pelanggan.
- b. Memperkuat hubungan dengan pelanggan melalui komunikasi yang lebih baik.

## **C. Struktur Organisasi Perusahaan**

Struktur organisasi adalah suatu kerangka kerja yang menunjukkan aktifitas dalam mencapai tujuan suatu organisasi yang saling berhubungan antar fungsi, tugas dan tanggung jawab. Suatu organisasi sangat diperlukan bagi perusahaan baik lingkup kecil maupun lingkup yang lebih besar. Karena dengan adanya organisasi suatu perusahaan dapat memanajemen kegiatan yang berjalan di dalamnya. Organisasi yang baik ialah apabila terciptanya kerjasama antar anggota sesuai dengan prosedur, tugas dan melaksanakan tanggung jawab yang diberikan dalam mendukung pencapaian tujuan yang diharapkan.

## **1. Struktur Organisasi Rama Laundry**

Berikut adalah struktur organisasi Rama Laundry



Gambar 4.1.  
Struktur Organisasi  
Sumber : Rama Laundry (2021)

## **2. Tugas dan Wewenang Struktur Organisasi Rama Laundry**

Berikut adalah uraian fungsi, tugas dan tanggung jawab struktur organisasi Rama Laundry pada masing-masing bagian:

- a. Pemilik Usaha (Pimpinan)
  - 1) Bertanggung jawab atas semua aktifitas yang ada
  - 2) Pengambil keputusan
  - 3) Memberi tugas dan wewenang kepada karyawan yang terkait
  - 4) Mendapatkan semua laporan kegiatan

- b. Operator Laundry
  - 1) Mencatat setiap kegiatan transaksi
  - 2) Menyimpan data-data transaksi pelanggan
- c. Staff Laundry
  - 1) Memberikan Pelayanan yang baik terhadap pelanggan
  - 2) Menerima perintah atau permintaan untuk mengambil cucian kotor
  - 3) Memastikan jumlah, jenis, dan kondisi cucian sesuai dengan laundry list.

## **D. Proses Bisnis Sistem Berjalan**

Dalam sistem penjualan barang pada Rama Laundry pelaksanaannya masih terbilang sangat sederhana. Pencatatan stok barang, penyimpanan data dan transaksi penjualan dilakukan secara manual dan berdasarkan penulisan tangan, oleh sebab itu kemungkinan akan terjadinya kesalahan di dalam pencatatan data sangatlah besar. Begitu pula dengan penyimpanan data, data tersimpan dalam map diletakkan di lemari arsip.

Berikut ini adalah gambaran secara garis besar mengenai penjelasan proses pemesanan barang, proses barang masuk, proses barang keluar, proses barang pengembalian, dan laporan:

### **1. Proses Penerimaan Barang**

Pada proses ini pelanggan datang ke laundry dengan membawa barang cucian yang ingin dicuci. Kemudian karyawan menyortir dan menghitung berdasarkan jenis cucian nya.

### **2. Proses Barang Masuk**

Pada proses ini barang cucian yang diterima oleh karyawan. Kemudian karyawan mencatat data barang dengan keterangan nama pelanggan, tanggal penerimaan barang, dan harga ke dalam nota sebanyak 3 rangkap. Nota biru diserahkan ke pelanggan, nota berwarna putih disimpan di arsip dan nota hijau diserahkan ke staff untuk di sortir.

### **3. Proses Barang Keluar**

Pada bagian ini terjadi proses pembayaran, pelanggan bisa melakukan pembayaran pada saat ingin melakukan laundry. Pelanggan melakukan pembayaran sesuai dengan harga yang sudah tercantum pada nota biru.

### **4. Proses Barang Pengembalian**

Pada proses ini barang yang sudah selesai di laundry, pelanggan bisa mengambil dengan membawa bukti pembayaran dengan keterangan nama pelanggan dan barang yang dilaundry. Berdasarkan nota biru, karyawan melakukan pengecekan ulang pada nota putih yang disimpan di arsip. Setelah itu karyawan menyerahkan barang laundry nya beserta nota putih sebagai bukti pembayaran ke pelanggan.

### **5. Laporan**

Karyawan membuat laporan penjualan barang untuk diserahkan kepada pemilik.

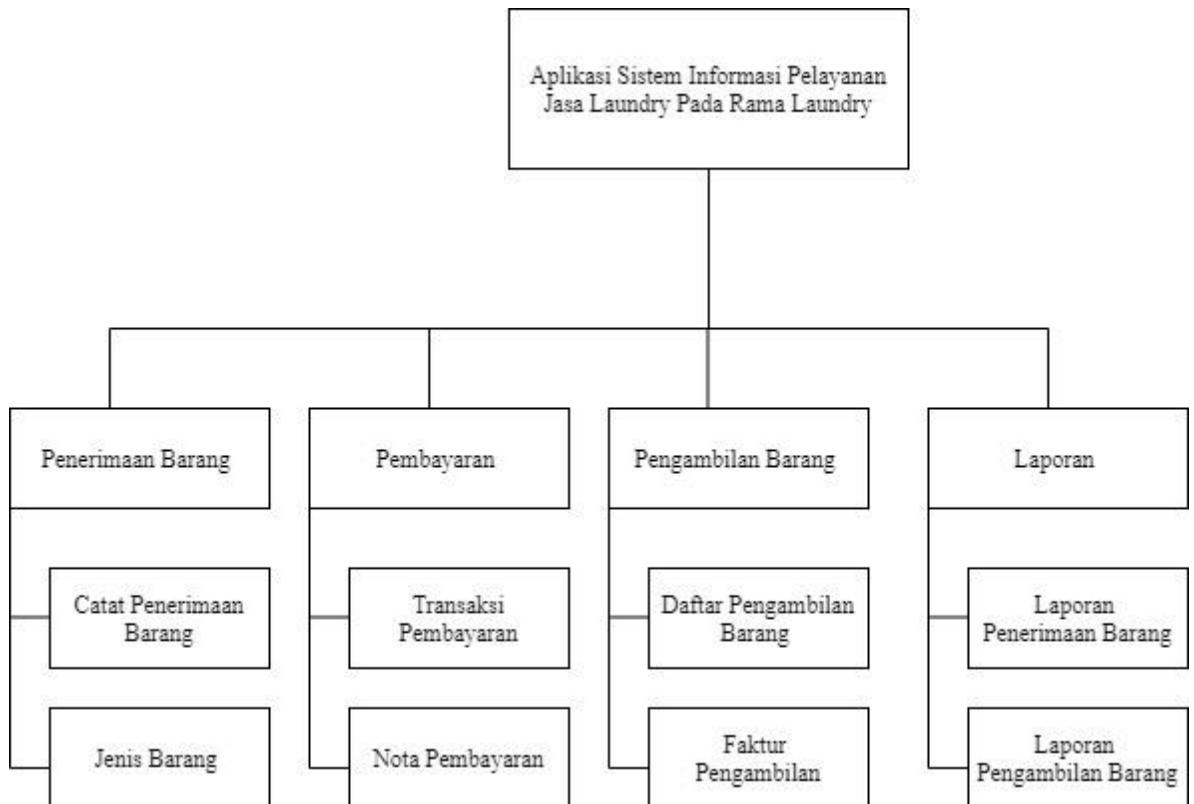
## **E. Aturan Bisnis Sistem Berjalan**

Aturan sistem manajemen penjualan barang yang berjalan pada Rama Laundry adalah sebagai berikut :

1. Karyawan melakukan proses penerimaan barang menggunakan sistem yang masih manual dengan menggunakan data-data yang masih dikelola dalam bentuk arsip atau dokumen, sehingga diperlukan proses yang terkomputerisasi pada Rama Laundry.
2. Dalam penerimaan barang, pelanggan datang ke laundry dengan membawa barang cucian nya yang akan di laundry. Karyawan menghitung dan menyortir barang cucian berdasarkan jenis cucian nya dan mencatat data barang ke dalam buku besar.
3. Dalam proses pembayaran, karyawan mencatat transaksi dengan cara manual yang kemudian bukti transaksi tersebut diserahkan ke pelanggan dan diarsipkan dalam pembayaran sebagai bahan laporan pembayaran yang nanti nya akan diberikan kepada pemilik laundry.
4. Dalam proses barang keluar dari gudang, bagian karyawan mencatat data barang yang telah dikeluarkan dan dikirimkan. Kemudian bagian gudang mengirimkannya ke toko dan pelanggan.
5. Dalam proses pengembalian barang dari rama laundry, pelanggan membawa bukti transaksi kemudian karyawan mengecek sesuai dengan arsip pembayaran yang tersimpan di laundry.
6. Karyawan membuat laporan untuk diserahkan kepada pemilik.

## F. Dekomposisi Fungsi Sistem

Dekomposisi fungsi yang berjalan pada Rama Laundry terbagi dalam 4 proses yaitu penerimaan barang, pembayaran, pengambilan barang dan laporan.



Gambar 4.2.  
Dekomposisi Fungsi Sistem  
Sumber : Rama Laundry (2021)

**Keterangan :**

**1. Penerimaan Barang**

Data Barang : Untuk mendata barang yang ingin di laundry

Jenis Barang : Karyawan melihat jenis barang sesuai dengan cucian yang pelanggan laundry

**2. Pembayaran**

Transaksi Pembayaran : Untuk melakukan pembayaran dengan barang yang sudah di laundry

Nota Pembayaran : Bukti pengambilan barang yang diberikan karyawan kepada pelanggan

**3. Pengambilan Barang**

Data Barang : Untuk mendata barang yang keluar sudah terjual.

**4. Pengembalian Barang**

Daftar Pengembalian : Untuk melihat data pelanggan yang ingin mengambil barang yang sudah di laundry

Faktur Pengembalian : Bukti pengambilan barang yang diberikan dari karyawan dan diterima oleh laundry

## 5. Laporan

Penerimaan Barang : Penerimaan barang selama sebulan yang dibuat karyawan untuk diserahkan ke pemilik usaha.

Pengambilan Barang : Pengambilan barang selama setiap hari yang dibuat karyawan dalam laporan sebulan untuk diserahkan ke pemilik usaha.

## G. Analisa Masukan, Proses dan Keluaran

### 1. Analisa Masukan

- a. Nama Masukan : Penerimaan Barang
  - Fungsi : Untuk mengisi data barang cucian dan jenis cucian
  - Sumber : Karyawan
  - Media : Kertas
  - Frekuensi : Setiap ada barang cucian masuk
  - Keterangan : Berisi data penerimaan barang cucian yang ingin dilaunder oleh pelanggan
- b. Nama Masukan : Nota Pembayaran
  - Fungsi : Untuk bukti pembayaran cucian
  - Sumber : Pelanggan
  - Media : Kertas
  - Frekuensi : Bila ada transaksi pelayanan jasa

Keterangan : Berisi bukti pembayaran cucian yang di Laundry

## 2. Analisa Proses

### a. Proses Penerimaan Barang

Pada proses ini ini berkaitan dengan pendataan barang cucian yang dilakukan antara pelanggan dan karyawan. Karyawan melayani pelanggan dalam pendataan barang yang di serahkan oleh pelanggan untuk di laundry.

### b. Proses Pembayaran

Pada proses ini dilakukan oleh bagian karyawan yang bertugas dalam hal pembayaran barang dan pembuatan nota untuk diberikan kepada pemilik. Pemilik menerima daftar pembayaran barang dan membuat faktur pembayaran barang kemudian pemilik memberikan kepada karyawan rama laundry.

### c. Proses Pengambilan Barang

Pada proses ini berkaitan dengan pengembalian barang yang dilakukan antara pelanggan dan karyawan. Karyawan melayani pelanggan dalam pengambilan barang yang di serahkan oleh pelanggan hingga proses pencucian dan perhitungan nota pembayaran, salinan nota pembayaran untuk pembayaran transaksi barang cucian yang akan diberikan kepada pelanggan.

### d. Proses Laporan

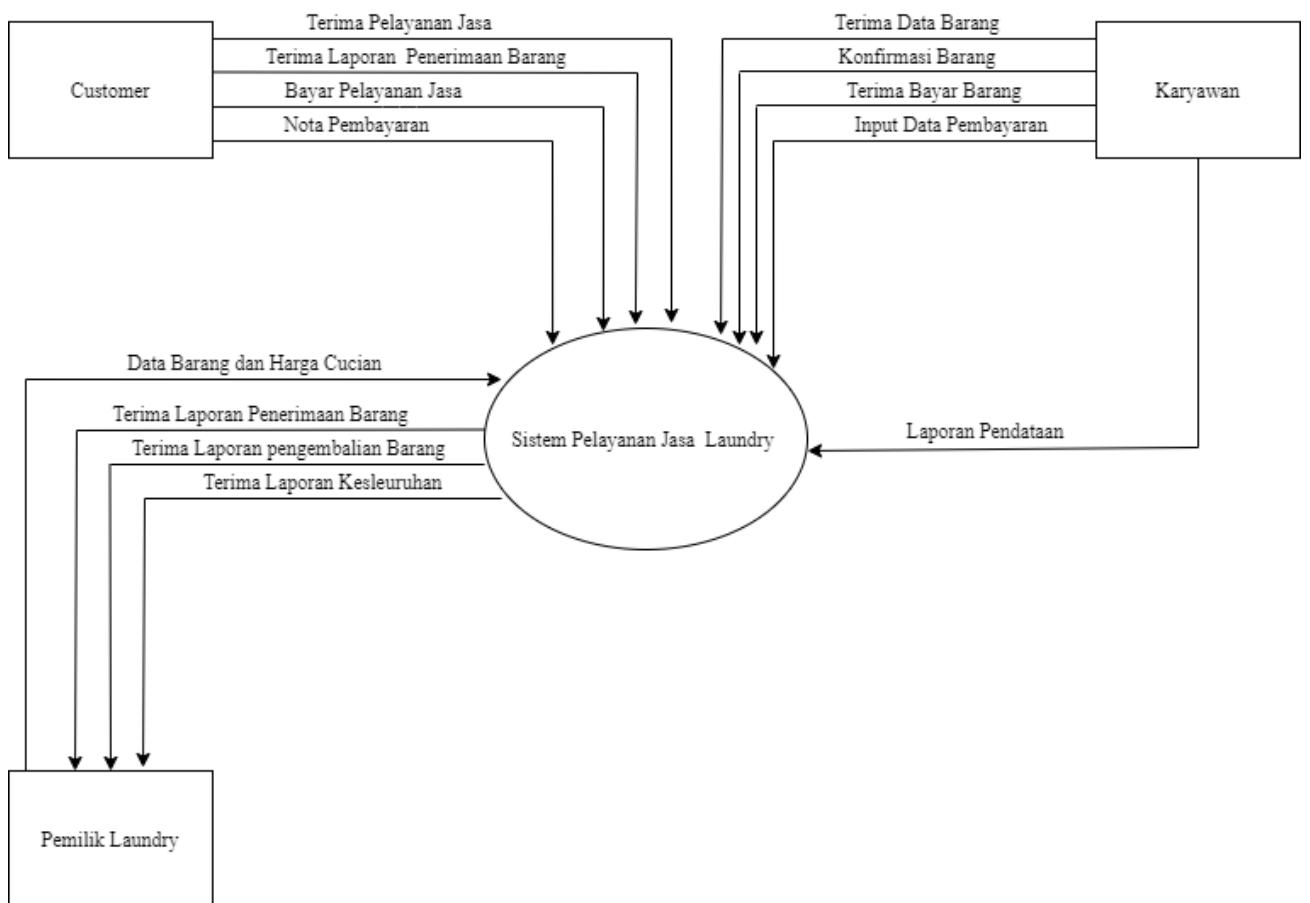
Pada proses ini karyawan membuat laporan yang berisi tentang penerimaan barang dan pengambilan barang dalam laundry selama sebulan.

### **3. Analisa Keluaran**

- e. Nama Keluaran : Laporan Penerimaan Barang  
Fungsi : Untuk mengetahui data cucian masuk yang ingin di laundry
- Sumber : Karyawan  
Media : Kertas  
Frekuensi : Setiap sebulan sekali  
Keterangan : Berisi tentang data penerimaan barang
- b. Nama Keluaran : Laporan Pengambilan Barang  
Fungsi : Untuk mengetahui data barang yang diambil setelah di laundry
- Sumber : Karyawan  
Media : Kertas  
Frekuensi : Setiap sebulan sekali  
Keterangan : Berisi tentang data pengambilan barang

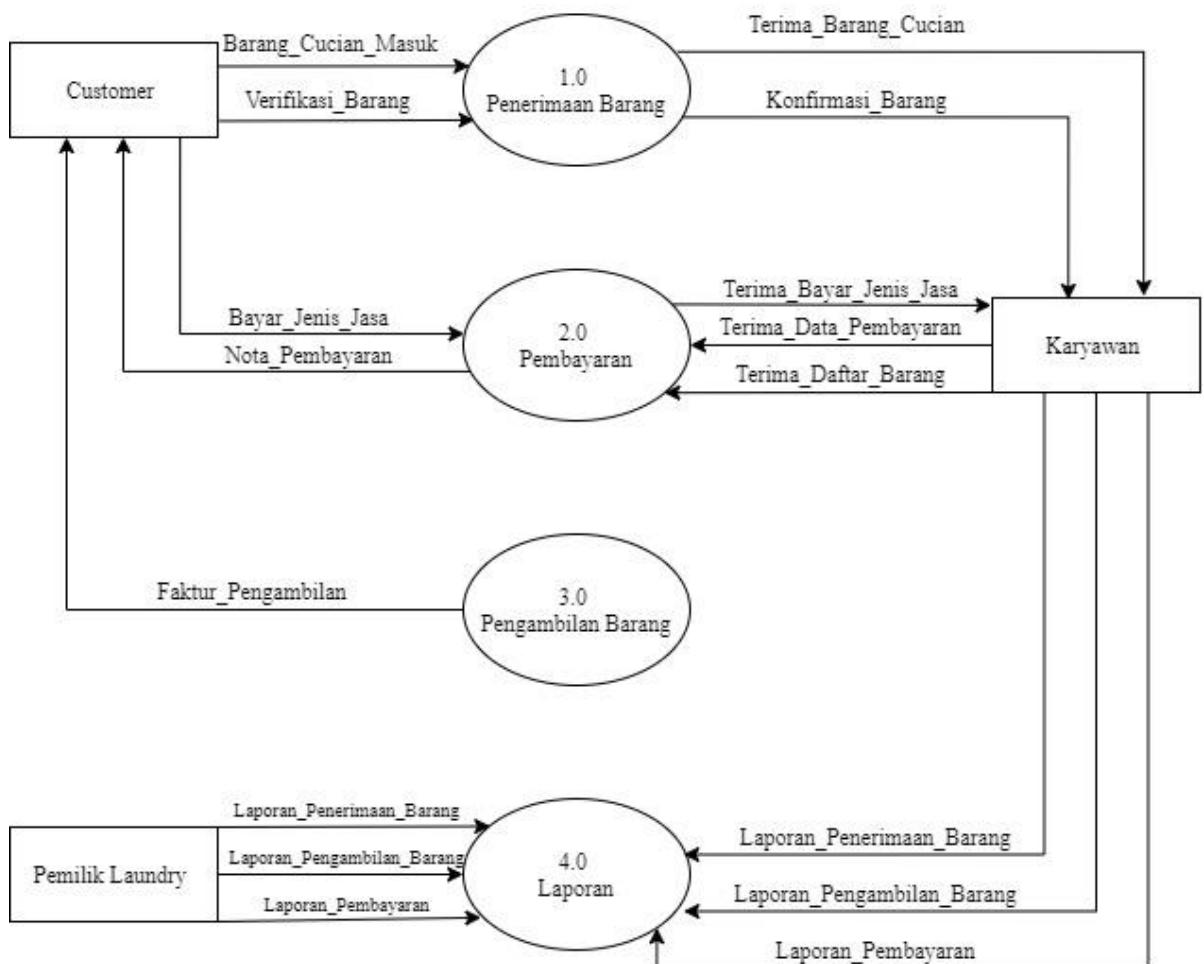
## H. Diagram Alir Data Sistem Berjalan

### 1. Diagram Konteks Sistem Berjalan



Gambar 4.3.  
Diagram Konteks Sistem Berjalan  
Sumber : Rama Laundry (2021)

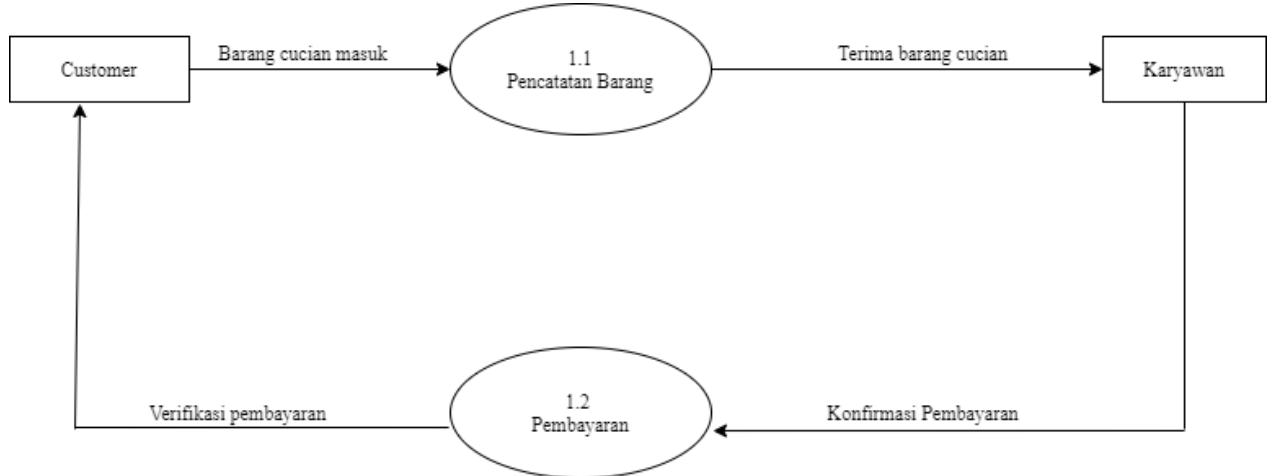
## 2. Diagram Nol Sistem Berjalan



Gambar 4.4.  
Diagram Nol Sistem Berjalan  
Sumber : Rama Laundry (2021)

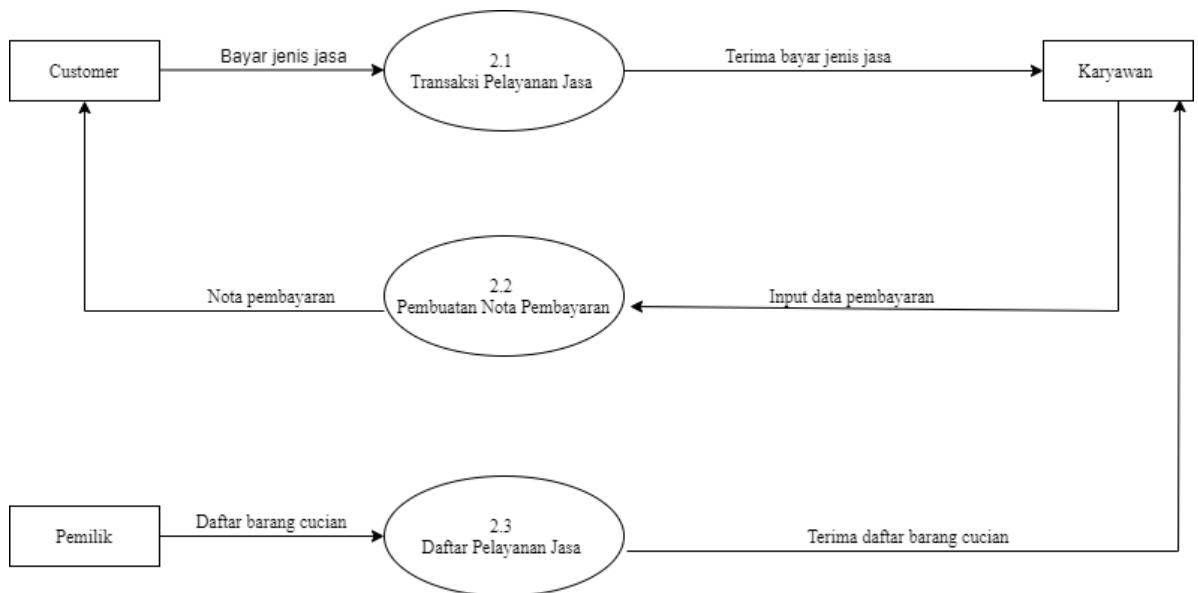
### 3. Diagram Rinci Sistem Berjalan

#### a. Level 1.0 Penerimaan Barang



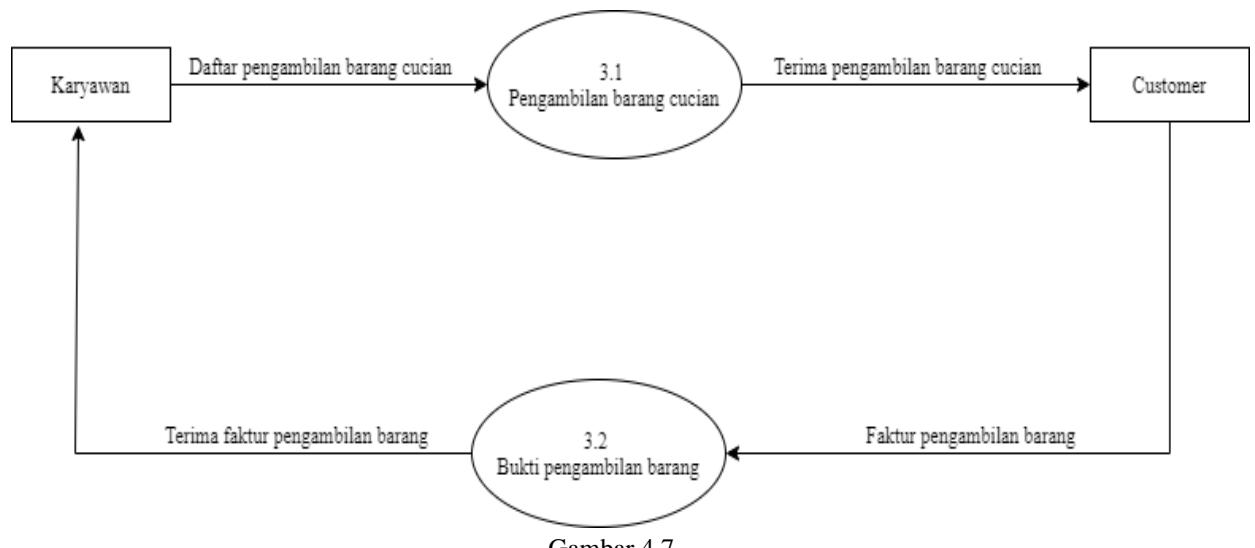
Gambar 4.5.  
Diagram Rinci Level 1.0 Sistem Berjalan  
Sumber : Rama Laundry (2021)

#### b. Level 2.0 Pembayaran



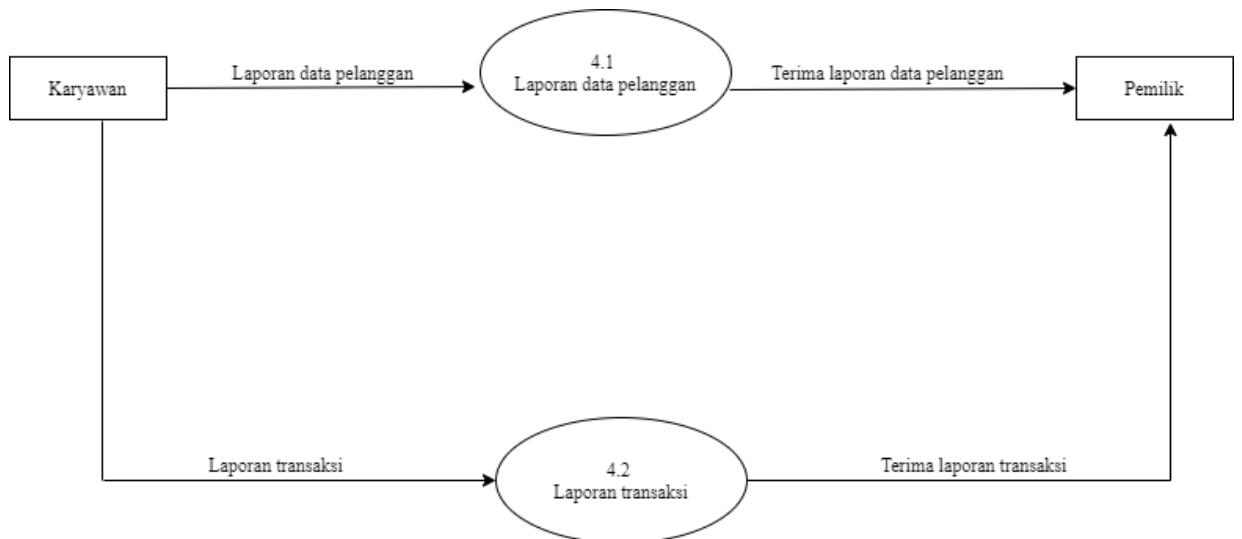
Gambar 4.6.  
Diagram Rinci Level 2.0 Sistem Berjalan  
Sumber : Rama Laundry (2021)

c. Level 3.0 Pengambilan barang



Gambar 4.7.  
Diagram Rinci Level 3.0 Sistem Berjalan  
Sumber : Rama Laundry (2021)

d. Level 4.0 Laporan



Gambar 4.8.  
Diagram Rinci Level 4.0 Sistem Berjalan  
Sumber : Rama Laundry (2021)

## **I. Analisis Permasalahan**

Sistem manajemen pada Rama Laundry masih tergolong sederhana.

Pencatatan data barang cucian serta penyimpanan dilakukan secara manual dan berdasarkan penulisan tangan, sehingga besar kemungkinan terjadinya kesalahan di dalam pembuatan laporan. Begitu pula dalam hal penyimpanan data, data tersimpan dalam map dan diletakkan dilemari arsip.

Dari pembahasan tersebut, penulis mengambil kesimpulan atas permasalahan yang ada diantaranya :

1. Secara keseluruhan proses pencatatan barang masih menggunakan cara manual yaitu menggunakan buku dan ditulis dengan tangan.
2. Karena keseluruhan data ditulis dengan media penyimpanan buku maka akan memerlukan banyak ruang untuk menyimpan data-data yang telah ditulis sebelumnya.
3. Karena menggunakan banyak ruang maka pencarian arsip datapun akan memerlukan banyak waktu.
4. Untuk proses pencatatan laporan transaksi masih ditulis dalam pencatatan nota dengan cara manual yang dapat menimbulkan kesalahan data pada saat mencatat transaksi.

## **J. Alternatif Penyelesaian Masalah**

Setelah mengetahui masalah-masalah yang terdapat dalam proses penginputan data serta pembuatan laporan pada Rama Laundry maka penulis mengusulkan pemecahan masalah untuk meningkatkan mutu pelayanan pada laundry tersebut, alternatif penyelesaian masalah tersebut secara garis besar adalah sebagai berikut :

1. Agar tingkat kesalahan dalam penulisan data-data serta laporan dapat dikurangi, alangkah baiknya Rama Laundry menerapkan sistem baru yang telah terkomputerisasi.
2. Jika telah berhasil menerapkan sistem yang telah terkomputerisasi maka media penyimpanan data-data serta laporan akan berubah yang semula adalah buku menjadi file-file yang pastinya lebih efisien.
3. Alangkah baiknya komputer yang digunakan pada sistem baru harus memiliki spesifikasi yang mendukung untuk menyimpan data yang jumlahnya banyak.
4. Untuk meminimalisir tingkat kesalahan pada pencatatan transaksi, sebaiknya media pencatatan transaksi diubah yang semula adalah nota menjadi file-file yang lebih efisien.

Dengan dibuatnya sistem manajemen pelayanan jasa laundry pada Rama Laundry dengan sistem terkomputerisasi ini, diharapkan dapat mempermudahkan proses penginputan dan pengecekan data khususnya pada proses pembuatan laporan dan stok barang.

## **K. Aturan Bisnis Sistem yang Diusulkan**

Setelah menganalisa masalah-masalah yang timbul maupun berpotensi timbul dalam pelaksanaan sistem berjalan di Rama Laundry inti khususnya dalam hal pencatatan, maka penulis mempunyai pemikiran untuk merancang suatu aplikasi, aplikasi ini akan menerima data barang masuk, keluar, pengembalian dan transaksi penjualan dalam suatu basis data dan menghasilkan laporan yan sesuai data.

### **1. Pendataan**

Bagian operator laundry bisa meng data pelanggan kedalam sistem database yang telah dibuat dan bisa meng data pelanggan.

### **2. Pendataan Barang Masuk**

Bagian operator laundry bisa meng barang cucian yang sudah masuk dan tercatat oleh karyawan dari pemesanan yang dilakukan oleh karyawan.

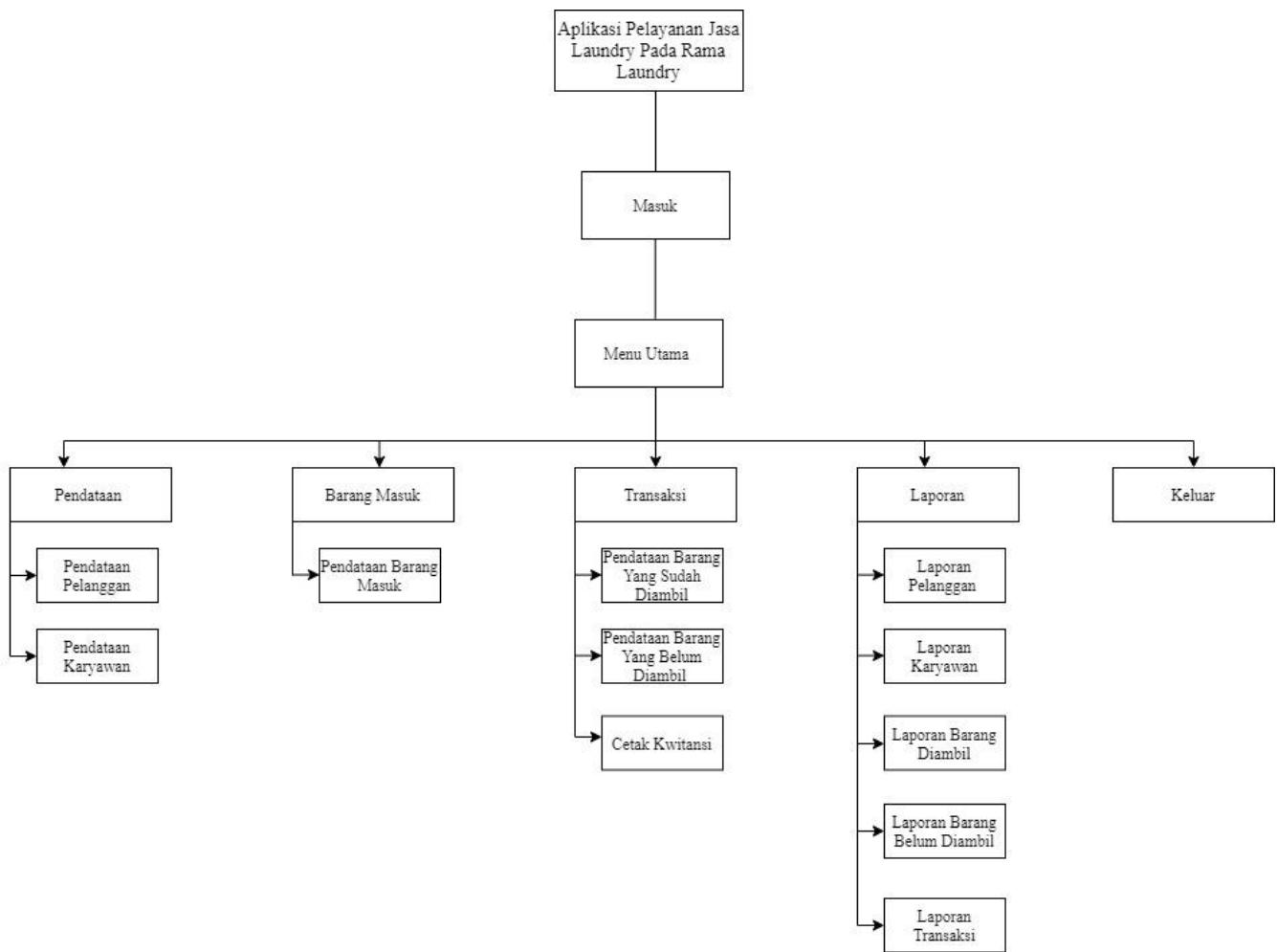
### **3. Pendataan Transaksi**

Bagian operator laundry bisa mendata barang yang keluar.

### **4. Laporan**

Data keseluruhan dari pelanggan, karyawan, barang yang sudan diambil, barang yang belum diambil dan data transaksi barang yang masih ada di gudang dan diserahkan kepada pemilik laundry.

## L. Dekomposisi Fungsi Sistem yang Diusulkan



Gambar 4.9.  
Dekomposisi Fungsi Sistem yang Diusulkan  
Sumber : Rama Laundry (2021)

Keterangan :

1. Masuk : Untuk masuk kedalam sebuah aplikasi ini harus masuk dengan nama pengguna dan kata sandi yang dapat dibuka oleh operator laundry.

2. Pendataan

Pendataan Pelanggan : Dalam pendataan ini diisi dengan pendataan pelanggan yang terdiri dari id pelanggan, nama pelanggan, nomor telepon dan alamat.

Pendataan Karyawan : Dalam pendataan ini diisi dengan pendataan karyawan yang terdiri dari id karyawan, nama karyawan, nomor telepon, dan password.

3. Data Barang Masuk

Pendataan Barang

Masuk : Dalam pendataan ini diisi dengan pendataan barang masuk yang terdiri dari id pelanggan, nama pelanggan, id barang, nama barang.

4. Transaksi

Barang diambil : Dalam pendataan ini diisi dengan pendataan transaksi yang terdiri dari no transaksi, tanggal, id pelanggan, nama pelanggan, id barang, nama barang, tarif, berat, total, diskon.

Barang belum diambil : Dalam pendataan ini diisi dengan pendataan transaksi yang terdiri dari no transaksi, tanggal, id pelanggan, nama pelanggan, id barang, nama barang, tarif, berat, total, diskon.

Cetak Kwitansi : Dalam aplikasi ini dapat mencetak bukti kwitansi pembayaran yang sudah dibayar oleh pelanggan.

## 5. Laporan

Data Pelanggan : Laporan ini berisi tentang data pelanggan yang sudah sering melaundry di Rama Laundry

Data Karyawan : Laporan ini berisi tentang data karyawan yang sudah bekerja di Rama Laundry

Data Barang Yang  
sudah diambil : Laporan ini berisi tentang data barang yang sudah diambil oleh Pelanggan ke Rama Laundry.

Data Barang Yang  
Belum Diambil : Laporan ini berisi tentang data barang yang belum diambil oleh Pelanggan ke Rama Laundry.

Transaksi : Laporan ini berisi tentang transaksi Pelanggan ke Rama Laundry.

Keluar : Untuk Keluar dari aplikasi.

## **M. Rancangan Masukan (Input), Proses, dan Keluaran (Output)**

### **Sistem yang Diusulkan**

#### **1. Rancangan Masukan (*Input*)**

Data yang menjadi masukan dalam sistem diusulkan antara lain sebagai berikut:

- a. Nama Masukan : Masuk
  - Fungsi : Untuk mengakses aplikasi pelayanan jasa
  - Sumber : Operator Laundry
  - Media : Komputer
  - Frekuensi : Setiap ada kegiatan penerimaan barang dan transaksi pada Rama Laundry
  - Keterangan : Operator laundry mengakses sistem pelayanan jasa dan transaksi pembelian, setiap ada kegiatan penerimaan barang dan transaksi di Rama Laundry
  
- b. Nama Masukan : Pendataan Pelanggan
  - Fungsi : Untuk mengetahui biodata pelanggan
  - Sumber : Pelanggan
  - Media : Komputer
  - Frekuensi : Setiap ada pencucian

- Keterangan : Operator laundry mendata pelanggan yang ingin melaundry di Rama Laundry
- c. Nama Masukan : Pendataan Karyawan  
Fungsi : Untuk mengetahui karyawan yang bekerja di Rama Laundry  
Sumber : Pemilik  
Media : Komputer  
Frekuensi : Setiap bulan  
Keterangan : Operator laundry mendata karyawan yang bekerja di Rama Laundry
- d. Nama Masukan : Pendataan Barang Masuk  
Fungsi : Mendata jenis barang cucian yang ingin di Laundry  
Sumber : Operator Laundry  
Media : Komputer  
Frekuensi : Setiap hari  
Keterangan : Operator Laundry mendata jenis barang yang ada di Rama Laundry
- e. Nama Masukan : Data Transaksi  
Fungsi : Mengetahui data transaksi dari pelanggan ke Rama Laundry

Sumber : Operator laundry  
Media : Komputer  
Frekuensi : Setiap ada pembeli  
Keterangan : Operator laundry mendata transaksi barang yang ingin di laundry.

f. Nama Masukan : Cetak Kwitansi  
Fungsi : Mengetahui bukti pembayaran yang telah dibayar oleh pelanggan  
Sumber : Operator laundry  
Media : Komputer  
Frekuensi : Setiap hari  
Keterangan : Sebagai bukti pembayaran

g. Nama Masukan : Laporan  
Fungsi : Mengetahui data-data yang sudah ada di Laundry  
Sumber : Operator Laundry  
Media : Komputer  
Frekuensi : Setiap bulan  
Keterangan : Mengetahui transaksi barang di Laundry

## **2. Rancangan Proses**

Pada bagian ini akan menguraikan secara umum mengenai proses yang terjadi pada sistem yang diusulkan.

a. Proses Pendataan Pelanggan, Karyawan

Proses pendataan ini admin meng data pelanggan, karyawan pada formulir aplikasi yang telah dibuat. Formulir ini nantinya akan berfungsi saat toko ingin melakukan pendataan pelanggan, karyawan.

b. Proses Barang Masuk

Bagian karyawan menerima barang cucian kotor dari pelanggan ke laundry. Dan operator laundry mendata barang cucian tersebut pada formulir aplikasi yang telah dibuat. Barang tersebut yang nantinya akan dilaundry.

c. Proses Transaksi

Bagian Operator laundry melakukan pendataan barang yang ingin di laundry oleh pelanggan ke laundry. Setelah pelanggan melakukan pembayaran, maka bagian operator laundry akan mencetak bukti transaksi.

d. Proses Laporan

Setelah melakukan pendataan barang cucian dan transaksi pelanggan selanjutnya bagian operator laundry akan membuat laporan transaksi yang akan diserahkan kepada pemilik.

### **3. Rancangan Keluaran (*Output*)**

- b. Nama Keluaran : Laporan Pelanggan  
Fungsi : Mengetahui Pelanggan  
Sumber : Pelanggan  
Media : Komputer  
Frekuensi : Setiap hari  
Keterangan : Untuk pemilik melihat data pelanggan
- c. Nama Keluaran : Laporan Karyawan  
Fungsi : Mengetahui Karyawan  
Sumber : Karyawan  
Media : Komputer  
Frekuensi : Setiap bulan  
Keterangan : Untuk pemilik melihat data karyawan
- d. Nama Keluaran : Laporan Barang Yang Sudah Diambil  
Fungsi : Mengetahui Barang Yang Sudah Diambil  
Sumber : Operator Laundry  
Media : Komputer  
Frekuensi : Setiap minggu  
Keterangan : Untuk pemilik melihat data barang yang  
sudah diambil
- e. Nama Keluaran : Laporan Barang Yang Belum Diambil  
Fungsi : Mengetahui barang yang belum diambil

Sumber : Operator Laundry

Media : Komputer

Frekuensi : Setiap minggu

Keterangan : Untuk pemilik tentang barang yang belum diambil

f. Nama Keluaran : Laporan Transaksi

Fungsi : Mengetahui transaksi yang ada di laundry

Sumber : Operator Laundry

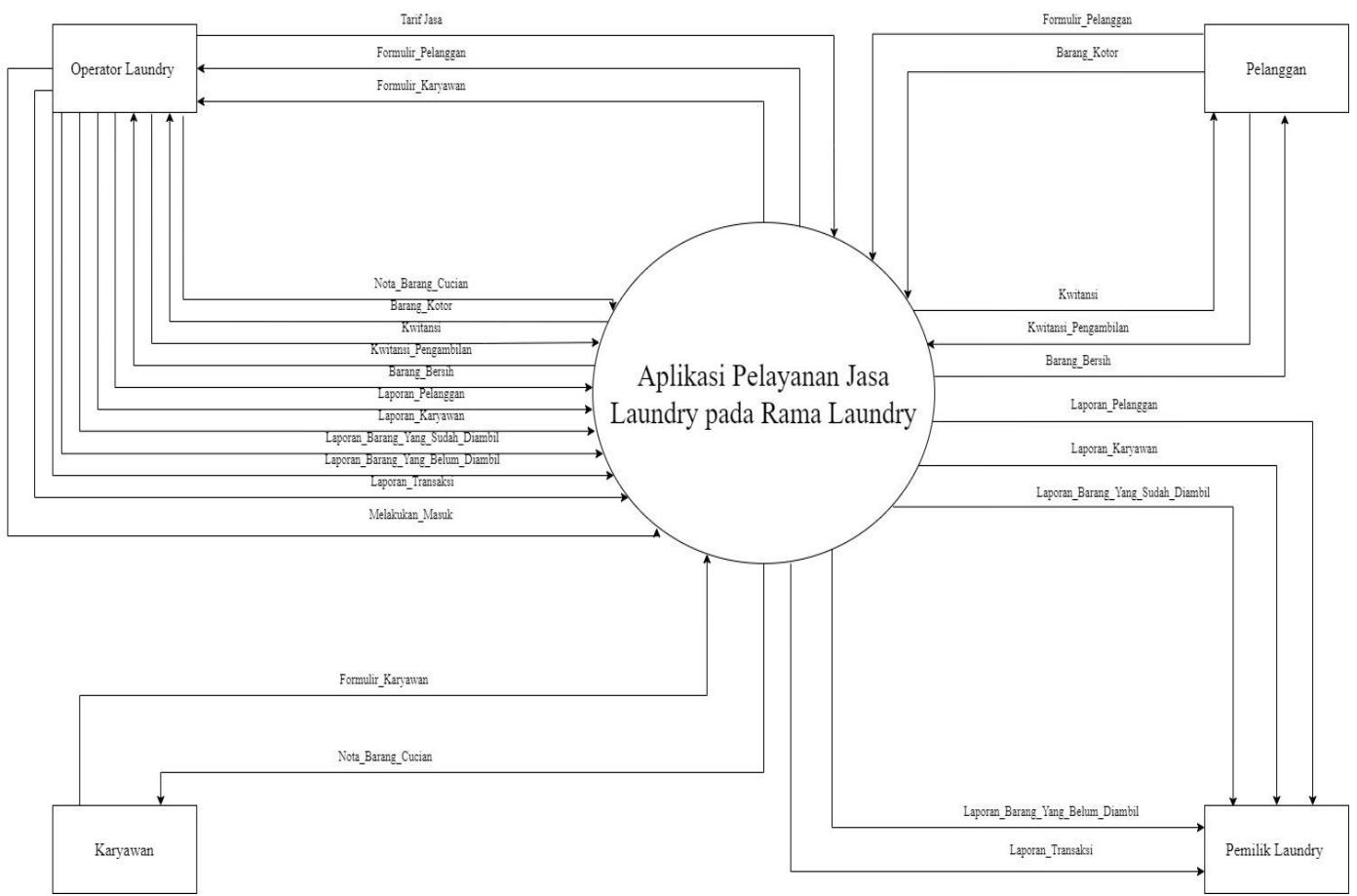
Media : Komputer

Frekuensi : Setiap bulan

Keterangan : Untuk pemilik melihat data transaksi di laundry

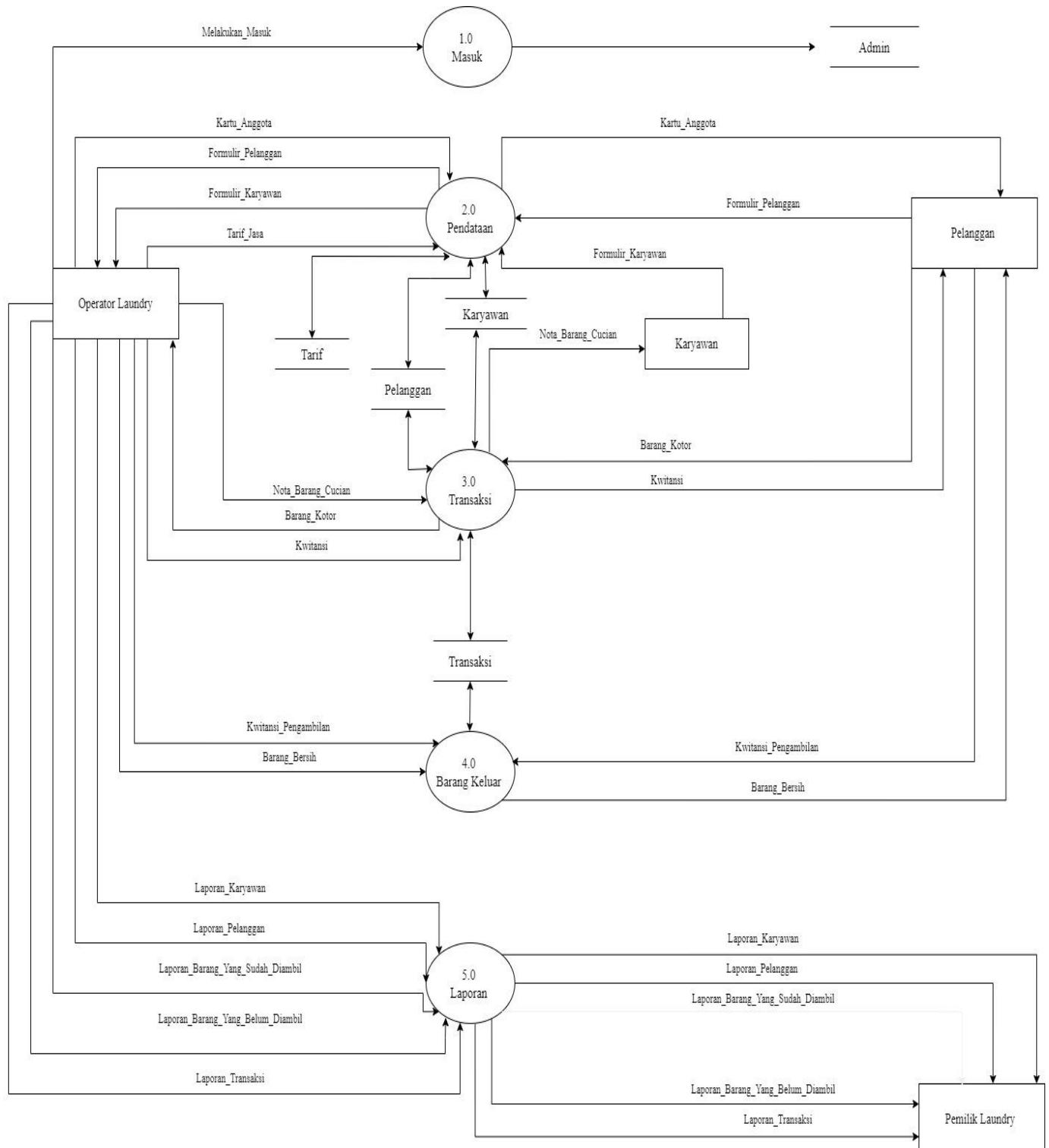
## N. Diagram Aliran Data (DAD) yang Diusulkan

### 1. Diagram Konteks



Gambar 4.10.  
Diagram Konteks yang Diusulkan  
Sumber : Penulis (2021)

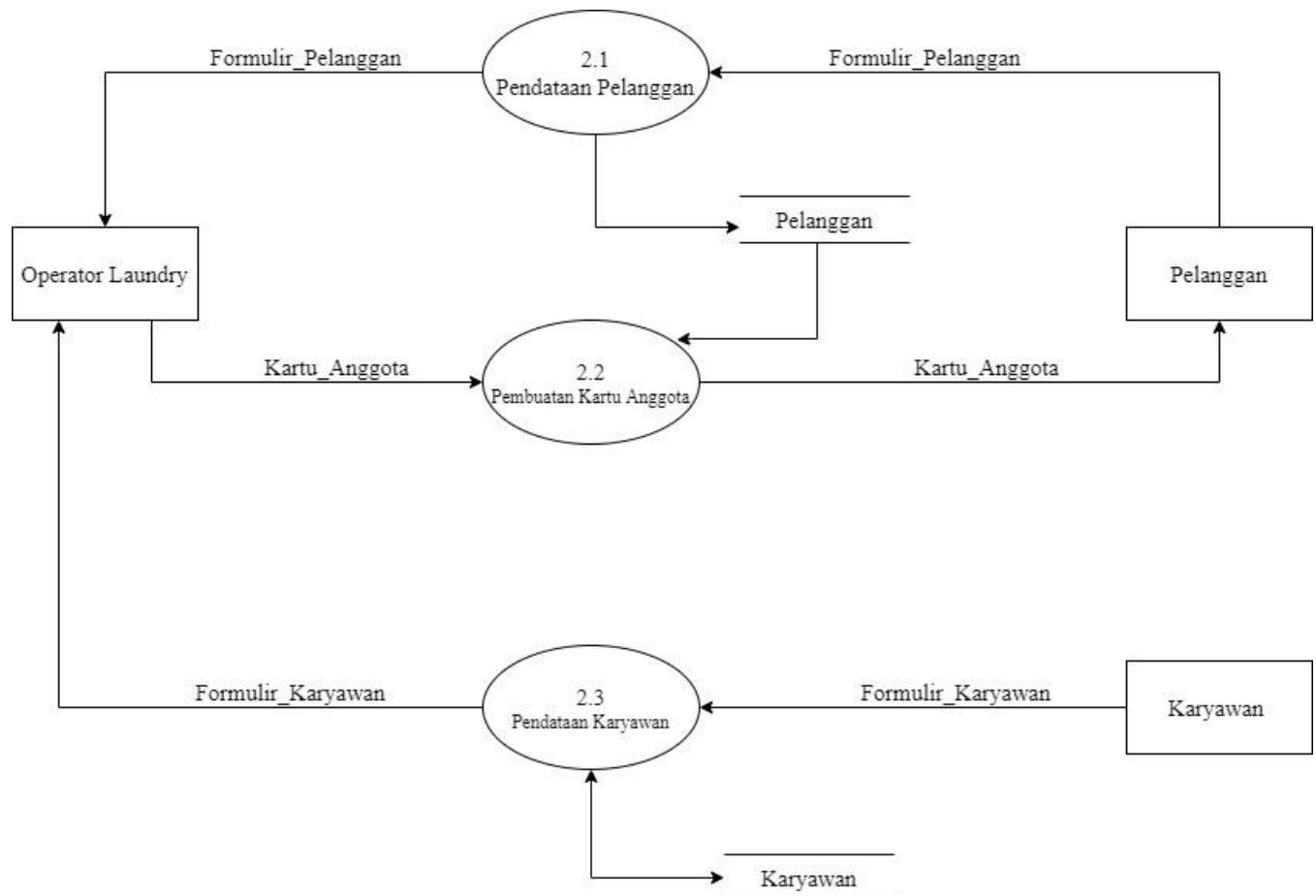
## 2. Diagram Nol



Gambar 4.11.  
Diagram Nol yang Diusulkan  
Sumber : Penulis (2021)

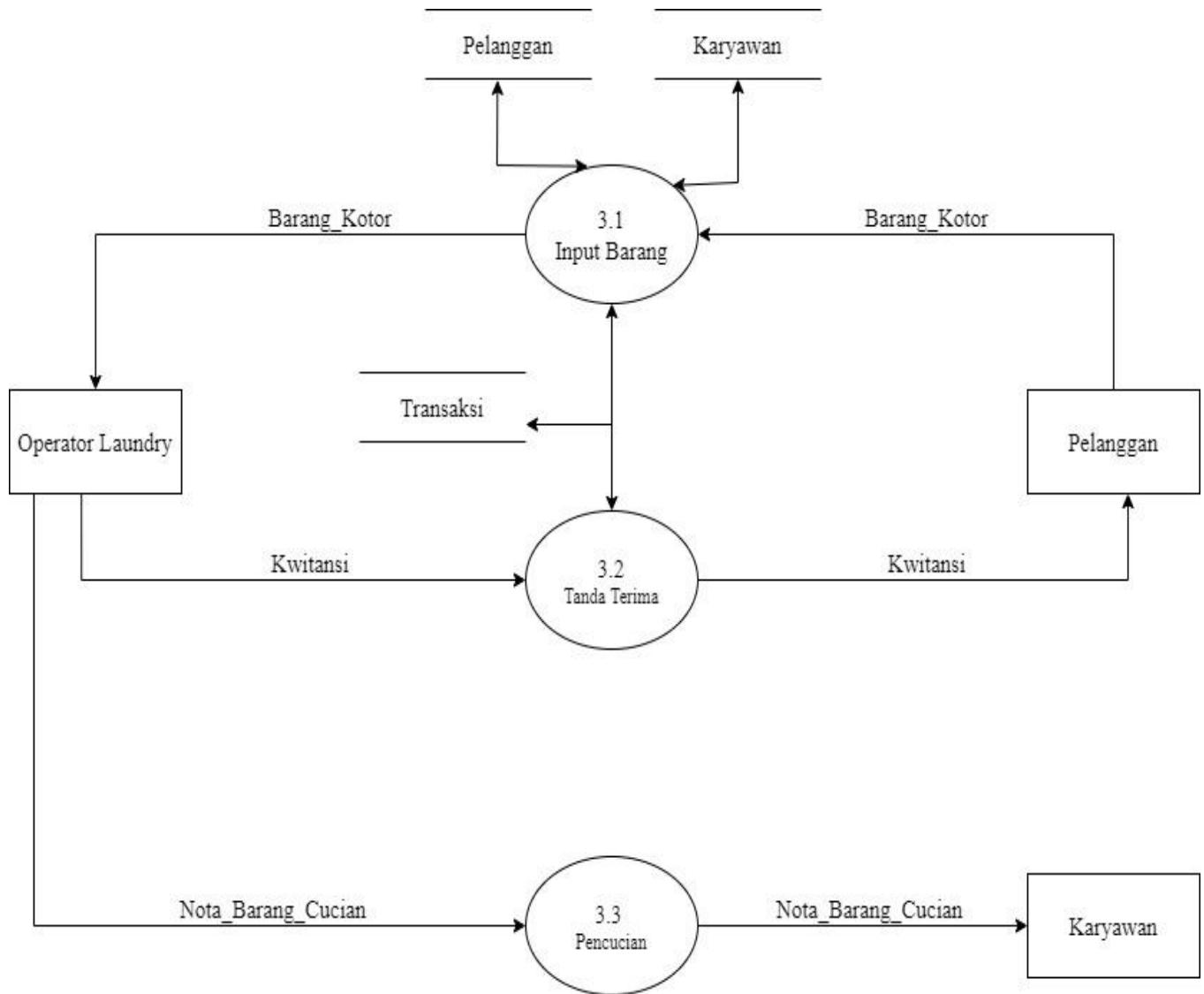
### 3. Diagram Rinci

#### 1. Level 2.0 Pendataan



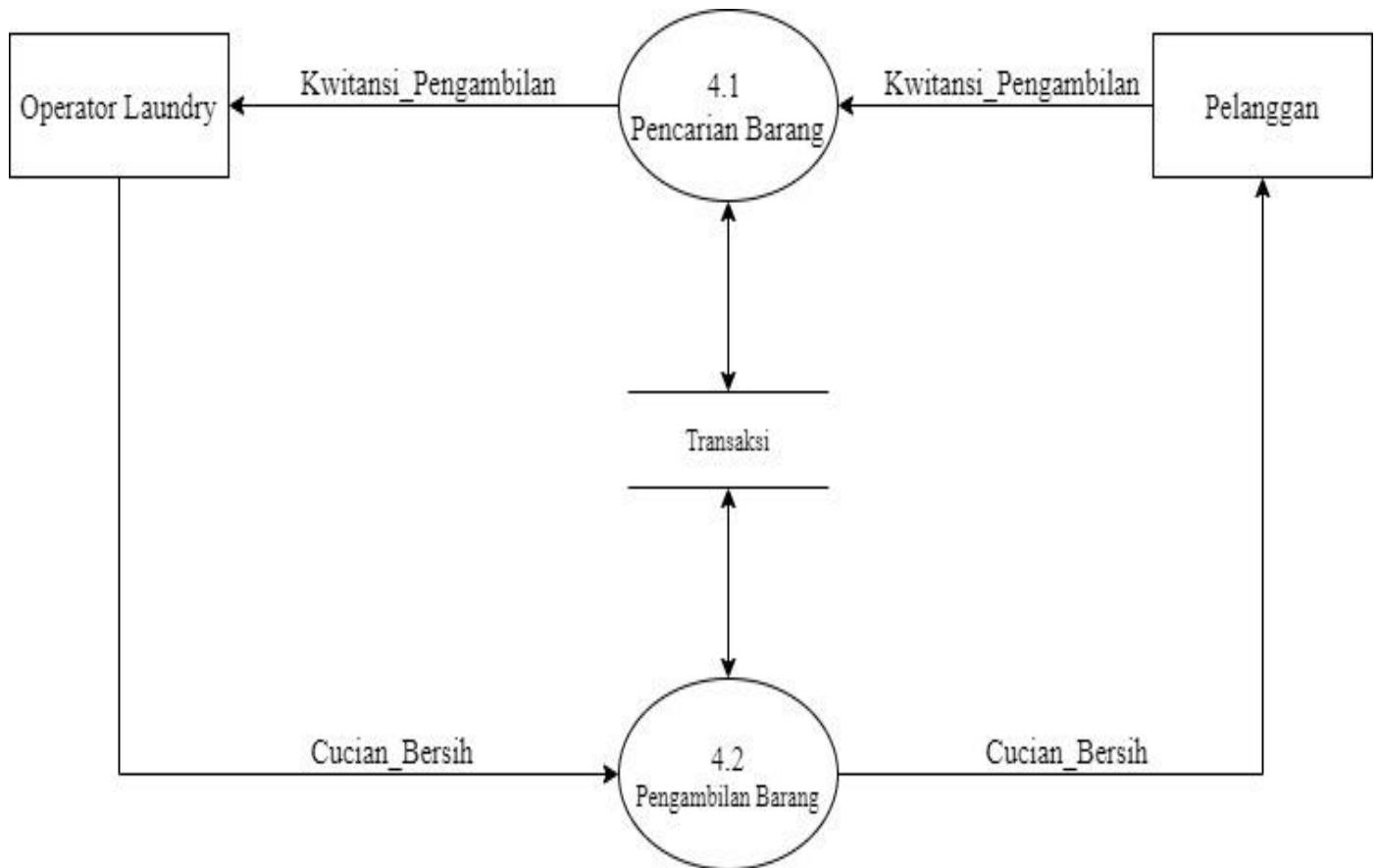
Gambar 4.12.  
Diagram Rinci Level 2.0 yang Diusulkan.  
Sumber : Penulis (2021)

## 2. Level 3.0 Transaksi



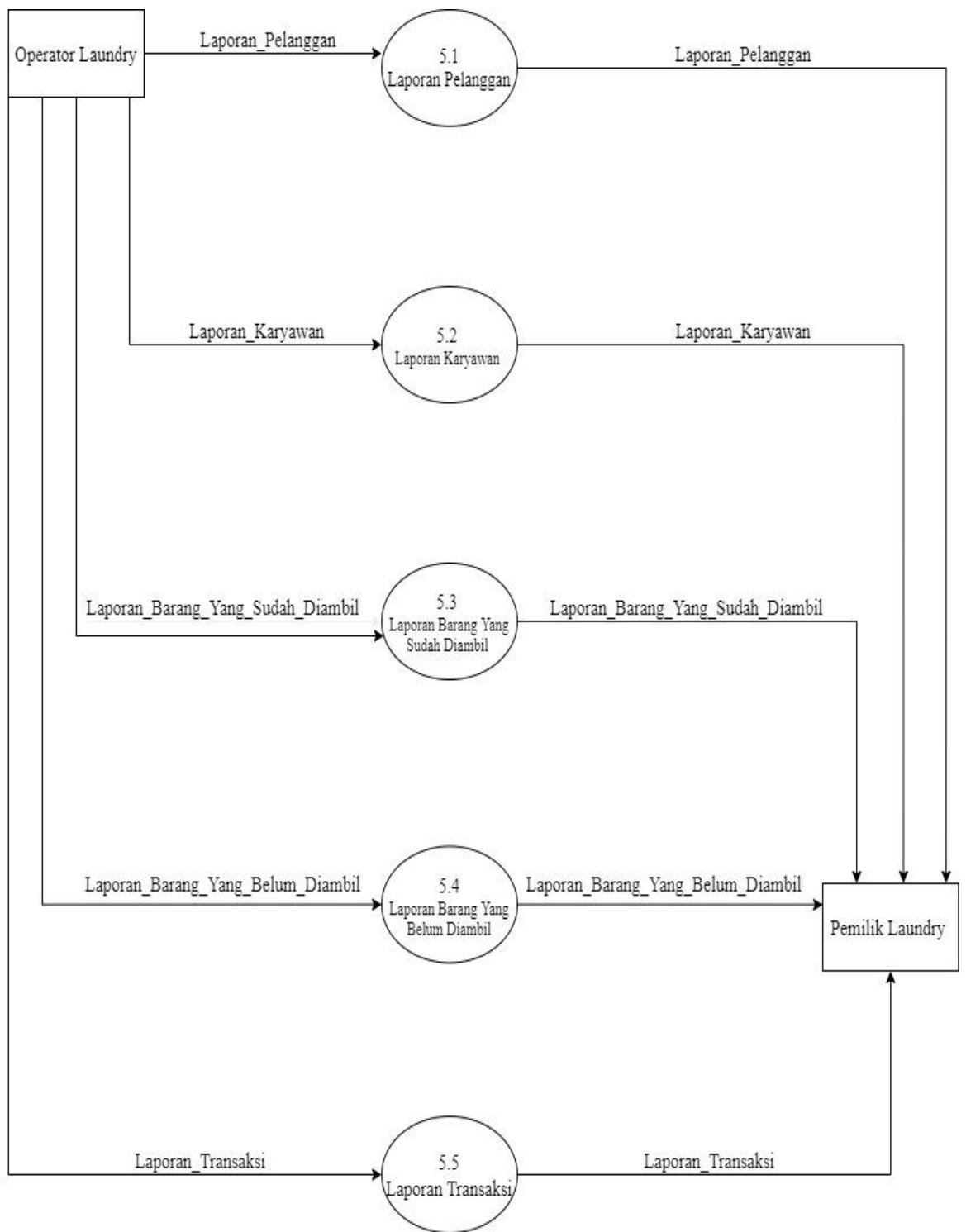
Gambar 4.13.  
Diagram Rinci Level 3.0 yang Diusulkan.  
Sumber : Penulis (2021)

### 3. Level 4.0 Barang Keluar



Gambar 4.14.  
Diagram Rinci Level 4.0 yang Diusulkan.  
Sumber : Penulis (2021)

#### 4. Level 5.0 Laporan



Gambar 4.15.  
Diagram Rinci Level 5.0 yang Diusulkan.  
Sumber : Penulis (2021)

## O. Kamus Data Sistem yang Diusulkan

Kamus data merupakan data penjelasan tertulis mengenai data yang berada didalam *database*. Adapun kamus data yang digunakan dalam merancang sistem yang akan berjalan adalah sebagai berikut :

### 1. Kamus Data Masukan

a. Nama Arus Data : Data Pelanggan

Alias : DP

Bentuk Data : *File*

Arus Data : Pelanggan-Proses 2.1-*Operator laundry*

Penjelasan : Bagian *Operator laundry* meng-*input* data pelanggan ke dalam *database* pelanggan

Periode : Setiap ada pelanggan baru

Volume : Setiap Bulan

Struktur Data : ID\_Pelanggan+Nama\_Pelanggan  
+No\_Telepon+Alamat

b. Nama Arus Data : Kartu Anggota

Alias : KA

Bentuk Data : *File*

Arus Data : Operator Laundry-Proses 2.2-*Pelanggan*

Penjelasan : Bagian karyawan memberikan data diri kepada *operator laundry*

Periode : Setiap ada karyawan baru

- Volume : Setiap bulan  
 Struktur Data : ID\_Pelanggan+Nama\_Pelanggan  
                   +No\_Telepon+Alamat
- c. Nama Arus Data : Pendataan Karyawan  
 Alias : PK  
 Bentuk Data : *File*  
 Arus Data : *Karyawan-Proses 2.3-Operator Laundry*  
 Penjelasan : Bagian Karyawan memberikan data diri kepada operator laundry  
 Periode : Setiap ada karyawan baru  
 Volume : Setiap bulan  
 Struktur Data : ID\_Karyawan + Nama\_Karyawan + No\_Telepon  
                   + Password
- d. Nama Arus Data : *Input Barang*  
 Alias : IB  
 Bentuk Data : *File*  
 Arus Data : *Pelanggan-Proses 3.1-Operator laundry*  
 Penjelasan : Pelanggan membawa barang cucian kotor yang ingin di laundry  
 Periode : Setiap ada pelanggan  
 Volume : Setiap hari  
 Struktur Data : No\_Transaksi + Tanggal + Id\_Pelanggan +  
                   Nama\_Pelanggan + Id\_Karyawan +

Nama\_Karyawan + Id Barang + Nama Barang +  
Tarif + Berat + Total + Diskon + Keterangan

- e. Nama Arus Data : Tanda Terima
- Alias : TT
- Bentuk Data : Dokumen
- Arus Data : Operator laundry-Proses 3.2-*Pelanggan*
- Penjelasan : Operator laundry memberikan tanda terima kepada *Pelangan*
- Periode : Setiap ada transaksi barang
- Volume : Setiap hari
- Struktur Data : -
- f. Nama Arus Data : Pencucian
- Alias : P
- Bentuk Data : *File*
- Arus Data : Operator laundry-Proses 3.3-Karyawan
- Penjelasan : Operator laundry memberikan nota barang cucian kepada karyawan
- Periode : Setiap ada barang cucian masuk
- Volume : Setiap hari
- Struktur Data : -
- g. Nama Arus Data : *Pencarian Barang*
- Alias : PB

Bentuk Data : *File*

Arus Data : Pelanggan-Proses 4.1-*File* Operator toko

Penjelasan : Pelanggan memberikan kwitansi pengambilan kepada operator laundry untuk mencari barang cucian yang sudah selesai di laundry

Periode : Setiap ada pencarian barang

Volume : Setiap hari

Struktur Data : No\_Transaksi + Tanggal + Id\_Pelanggan + Nama\_Pelanggan + Id\_Karyawan + Nama\_Karyawan + Id\_Barang + Nama\_Barang + Tarif + Berat + Total + Diskon + Keterangan

h. Nama Arus Data : Pengambilan Barang

Alias : PB

Bentuk Data : Barang

Arus Data : Operator laundry-Proses 4.2-Pelanggan

Penjelasan : Operator laundry memberikan barang cucian yang sudah bersih kepada pelanggan

Periode : Setiap ada pengambilan dari pelanggan

Volume : Setiap hari

Struktur Data : -

## **2. Kamus Data Keluaran**

- a. Nama Arus Data : Cetak Laporan Data Pelanggan
- Alias                 : CLDP
- Bentuk Data          : Dokumen
- Arus Data            : Operator Laundry-Proses 5.1- Pemilik
- Penjelasan           : Sebagai laporan data pelanggan
- Periode             : Setiap pemilik meminta laporan data pelanggan
- Volume              : Setiap hari
- Struktur Data       : ID\_Pelanggan+Nama\_Pelanggan  
                          +No\_Telepon+Alamat
- b. Nama Arus Data : Cetak Laporan Data Karyawan
- Alias                 : CLDK
- Bentuk Data          : Dokumen
- Arus Data            : Operator Laundry-Proses 5.2- Pemilik
- Penjelasan           : Sebagai laporan data karyawan
- Periode             : Setiap pemilik meminta laporan data karyawan
- Volume              : Setiap bulan
- Struktur Data       : ID\_Karyawan+Nama\_Karyawan+No\_Telepon  
                          +Password
- c. Nama Arus Data : Cetak Laporan Barang Yang Sudah Diambil
- Alias                 : CLBYSD
- Bentuk Data          : Dokumen
- Arus Data            : Operator Laundry-Proses 5.3- Pemilik

Penjelasan : Sebagai laporan barang yang sudah diambil

Periode : Setiap pemilik meminta laporan data barang yang  
sudah diambil

Volume : Setiap hari

Struktur Data : No\_Transaksi + Tanggal + ID\_Pelanggan +  
Nama\_Pelanggan + ID\_Karyawan +  
Nama\_Karyawan + ID\_Barang + Nama\_Barang +  
Tarif + Berat + Total + Diskon + Keterangan

d. Nama Arus Data : Cetak Laporan Barang Yang Belum Diambil

Alias : CLBYBD

Bentuk Data : Dokumen

Arus Data : Operator laundry-Proses 5.4-Pemilik

Penjelasan : Sebagai laporan barang yang belum diambil

Periode : Setiap pemilik meminta laporan barang yang  
belum diambil

Volume : Setiap Hari

Struktur Data : No\_Transaksi + Tanggal + ID\_Pelanggan +  
Nama\_Pelanggan + ID\_Karyawan +  
Nama\_Karyawan + ID\_Barang + Nama\_Barang +  
Tarif + Berat + Total + Diskon + Keterangan

- e. Nama Arus Data : Cetak Laporan Transaksi
- Alias : CLT
- Bentuk Data : Dokumen
- Arus Data : Operator laundry-Proses 5.5-Pemilik
- Penjelasan : Sebagai laporan transaksi
- Periode : Setiap Pemilik meminta laporan data transaksi
- Volume : Setiap Hari
- Struktur Data : No\_Transaksi + Tanggal + ID\_Pelanggan +  
Nama\_Pelanggan + ID\_Karyawan +  
Nama\_Karyawan + ID\_Barang + Nama\_Barang +  
Tarif + Berat + Total + Diskon + Keterangan

## **P. Spesifikasi Proses Sistem yang Diusulkan**

Spesifikasi proses yang diusulkan pada Rama Laundry menjelaskan spesifikasi dari setiap proses pada diagram rinci :

### **1. Proses 1.0**

Nama Proses : Masuk  
Masukan : Melakukan masuk  
Keluaran : Mengakses sistem aplikasi manajemen barang  
Uraian : Bagian *operator laundry* dapat menginput data barang

### **2. Proses 2.1**

Nama Proses : Pendataan Pelanggan  
Masukan : Data Pelanggan  
Keluaran : *File* Pelanggan  
Uraian : Bagian operator laundry menginput data pelanggan

### **3. Proses 2.2**

Nama Proses : Pembuatan Kartu Anggota  
Masukan : Data Pelanggan  
Keluaran : Kartu Anggota  
Uraian : Bagian operator laundry menginput data pelanggan

### **4. Proses 2.3**

Nama Proses : Pendataan Karyawan  
Masukan : Data Karyawan  
Keluaran : Data Karyawan, *File* Karyawan  
Uraian : Karyawan memberikan data ke bagian *operator laundry*

### 5. Proses 3.1

Nama Proses : Input Barang  
Masukan : Barang Kotor  
Keluaran : Barang Kotor  
Uraian : Operator laundry Menerima barang kotor dari pelanggan

### 6. Proses 3.2

Nama Proses : Tanda Terima  
Masukan : Kwitansi  
Keluaran : Kwitansi  
Uraian : Operator laundry memberikan kwitansi ke pelanggan

### 7. Proses 3.3

Nama Proses : Pencucian  
Masukan : Nota Barang Cucian  
Keluaran : Nota Barang Cucian

Uraian : Operator laundry memberikan nota barang cucian ke karyawan lalu karyawan mencari barang cucian untuk di laundry

#### **8. Proses 4.1**

Nama Proses : Pencarian Barang  
Masukan : Kwitansi Pengambilan  
Keluaran : Kwitansi Pengambilan  
Uraian : Pelanggan memberikan kwitansi pengambilan ke operator laundry untuk mencari barang yang sudah selesai di laundry

#### **9. Proses 4.2**

Nama Proses : Pengambilan Barang  
Masukan : Cucian Bersih  
Keluaran : Cucian Bersih  
Uraian : Bagian Operator laundry memberikan barang cucian bersih kepada pelanggan

#### **10. Proses 5.1**

Nama Proses : Cetak Laporan Data Pelanggan  
Masukan : *File* Data Pelanggan, Laporan Data Pelanggan  
Keluaran : *File* Data Pelanggan, Laporan Data Pelanggan  
Uraian : Bagian *operator laundry* mencetak laporan untuk pemilik

### **11. Proses 5.2**

Nama Proses : Cetak Laporan Data Karyawan  
Masukan : *File* Data Karyawan, Laporan Data Karyawan  
Keluaran : *File* Data Karyawan, Laporan Data Karyawan  
Uraian : Bagian *operator laundry* mencetak laporan untuk  
Pemilik

### **12. Proses 5.3**

Nama Proses : Cetak Laporan Barang Yang Sudah Diambil  
Masukan : *File* Data Transaksi, Laporan Barang Yang Sudah  
Diambil  
Keluaran : *File* Data Transaksi, Laporan Barang Yang Sudah  
Diambil  
Uraian : Bagian *operator laundry* mencetak laporan untuk  
pemilik

### **13. Proses 5.4**

Nama Proses : Cetak Laporan Barang Yang Belum Diambil  
Masukan : File Data Transaksi, Laporan Barang Yang Belum  
Diambil  
Keluaran : File Data Transaksi, Laporan Barang Yang Belum  
Diambil

Uraian : Bagian operator laundry mencetak laporan untuk pemilik

#### **14. Proses 5.5**

Nama Proses : Cetak Laporan Transaksi

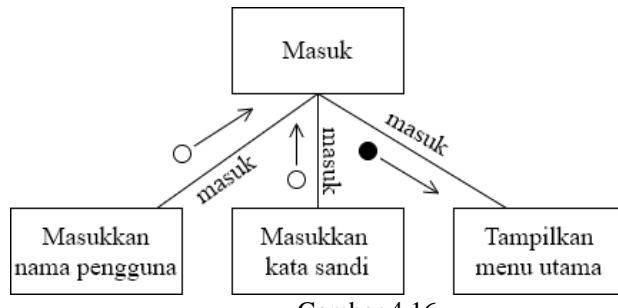
Masukan : File Data Transaksi, Laporan Transaksi

Keluaran : File Data Transaksi, Laporan Transaksi

Uraian : Bagian operator laundry mencetak laporan untuk pemilik

## Q. Bagan Terstruktur yang Diusulkan

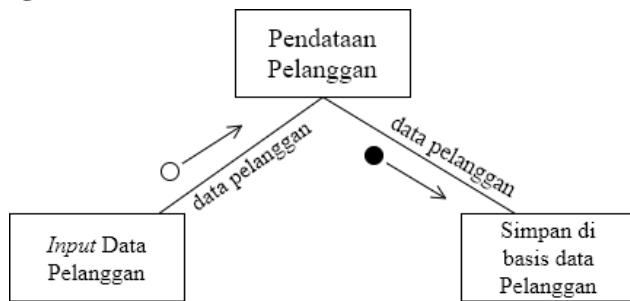
### 1. Bagan Terstruktur Proses 1.0



Gambar 4.16.

Bagan Terstruktur Proses 1.0 yang Diusulkan  
Sumber : Penulis (2021)

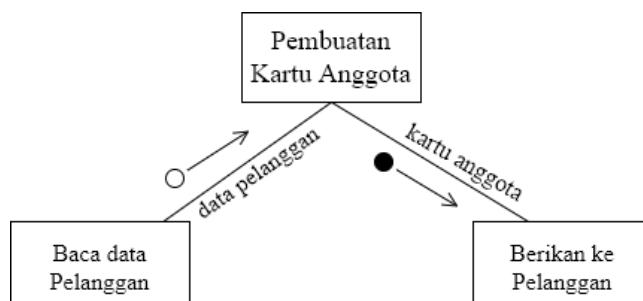
### 2. Bagan Terstruktur Proses 2.1



Gambar 4.17.

Bagan Terstruktur Proses 2.1 yang Diusulkan  
Sumber : Penulis (2021)

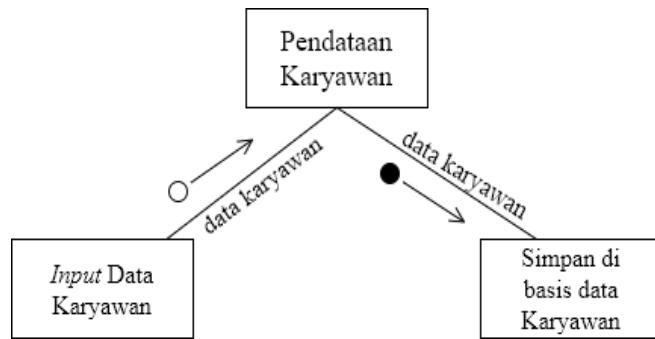
### 3. Bagan Terstruktur Proses 2.2



Gambar 4.18.

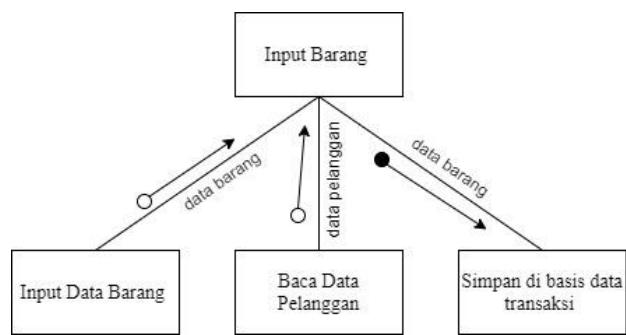
Bagan Terstruktur Proses 2.2 yang Diusulkan  
Sumber : Penulis (2021)

#### 4. Bagan Terstruktur Proses 2.3



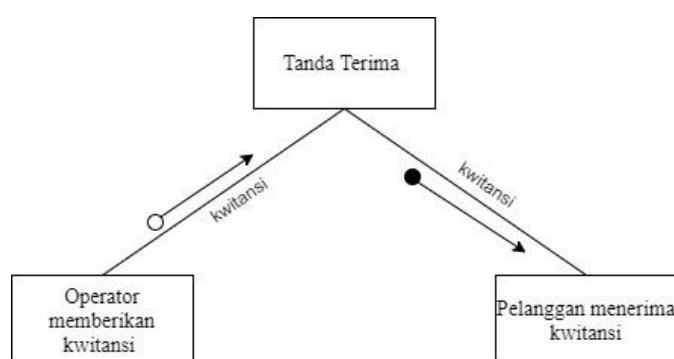
Gambar 4.19.  
Bagan Terstruktur Proses 2.3 yang Diusulkan  
Sumber : Penulis (2021)

#### 5. Bagan Terstruktur Proses 3.1



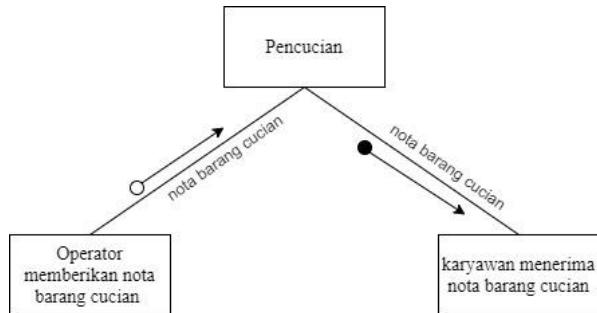
Gambar 4.20.  
Bagan Terstruktur Proses 3.1 yang Diusulkan  
Sumber : Penulis (2021)

#### 6. Bagan Terstruktural Proses 3.2



Gambar 4.21.  
Bagan Terstruktural Proses 3.2 yang Diusulkan  
Sumber : Penulis (2021)

## 7. Bagan Terstruktur Proses 3.3

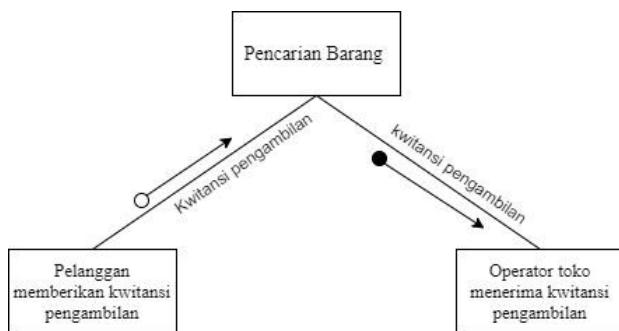


Gambar 4.22.

Bagan Terstruktur Proses 3.3 yang Diusulkan

Sumber : Penulis (2021)

## 8. Bagan Terstruktur Proses 4.1

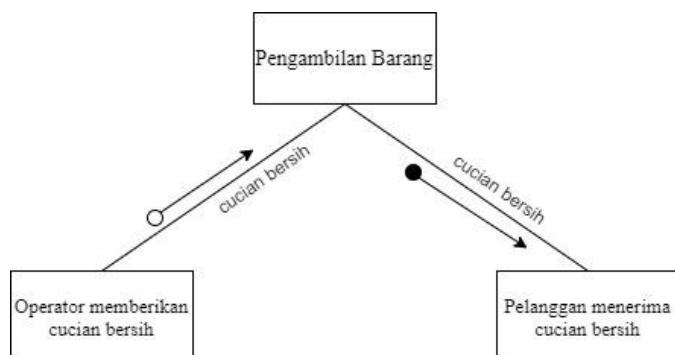


Gambar 4.23.

Bagan Terstruktur Proses 4.1 yang Diusulkan

Sumber : Penulis (2021)

## 9. Bagan Terstruktur Proses 4.2

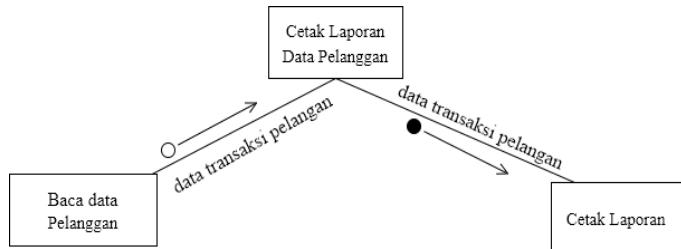


Gambar 4.24.

Bagan Terstruktur Proses 4.2 yang Diusulkan

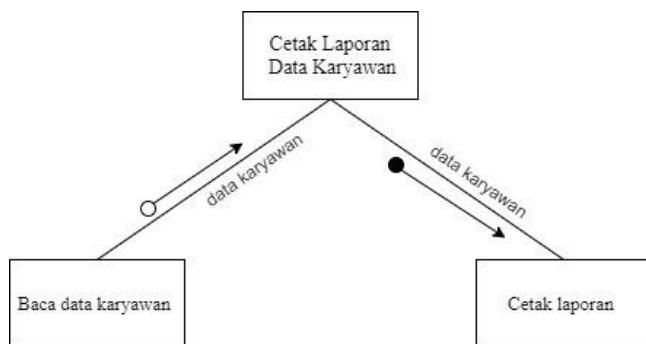
Sumber : Penulis (2021)

## 10. Bagan Terstruktur Proses 5.1



Gambar 4.25.  
Bagan Terstruktur Proses 5.1 yang Diusulkan  
Sumber : Penulis (2021)

## 11. Bagan Terstruktur Proses 5.2



Gambar 4.26.  
Bagan Terstruktur Proses 5.2 yang Diusulkan  
Sumber : Penulis (2021)

## 12. Bagan Terstruktur Proses 5.3



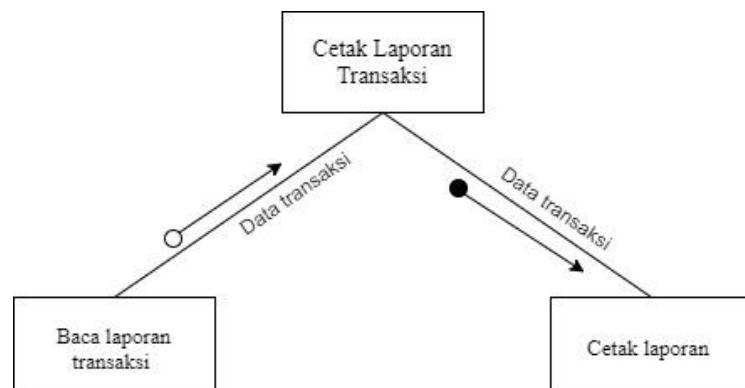
Gambar 4.27.  
Bagan Terstruktur Proses 5.3 yang Diusulkan  
Sumber : Penulis (2021)

### 13. Bagan Terstruktur Proses 5.4



Gambar 4.28.  
Bagan Terstruktur Proses 5.4 yang Diusulkan  
Sumber : Penulis (2021)

### 14. Bagan Terstruktur Proses 5.5



Gambar 4.29.  
Bagan Terstruktur Proses 5.5 yang Diusulkan  
Sumber : Penulis (2021)

## R. Spesifikasi Modul Sistem yang Diusulkan

### 1. Modul Login

Mulai

Masukkan Nama Pengguna

Masukan Kata Sandi

Selesai

### 2. Modul Menu Utama

Mulai

Pilih Menu Karyawan

Pilih Menu Pelanggan

Pilih Menu Jenis Barang

Pilih Menu Transaksi

Pilih Menu Barang

Pilih Menu Laporan

Pilih Menu Keluar

Selesai

### 3. Modul *Input* Karyawan

Mulai

Kerjakan modul *input* karyawan

Baca *file* data karyawan

*Input* data karyawan

Simpan ke *file* karyawan

Selesai

#### **4. Modul *Input* Pelanggan**

Mulai

Kerjakan modul *input* pelanggan

Baca *file* data pelanggan

*Input* data pelanggan

Simpan ke *file* pelanggan

Selesai

#### **5. Modul *Input* Jenis Barang**

Mulai

Kerjakan modul *input* jenis barang

Baca *file* jenis barang

Baca *file* jenis barang

Simpan ke *file* jenis barang

Selesai

#### **6. Modul Input Barang**

Mulai

Baca file barang

Update file barang

Selesai

#### **8. Modul Transaksi**

Mulai

Kerjakan modul *input* transaksi

*Input* no transaksi

*Input* tanggal transaksi

Baca *file* pelanggan

Baca *file* karyawan

Baca *file* jenis barang

Simpan ke *file* transaksi

Selesai

## **9. Modul Laporan**

Mulai

Cetak Laporan Pelanggan

Cetak Laporan Karyawan

Cetak Laporan Barang Yang Sudah Diambil

Cetak Laporan Barang Yang Belum Diambil

Cetak Laporan Transaksi

Selesai

## **S. Rancangan Basis Data Sistem yang Diusulkan**

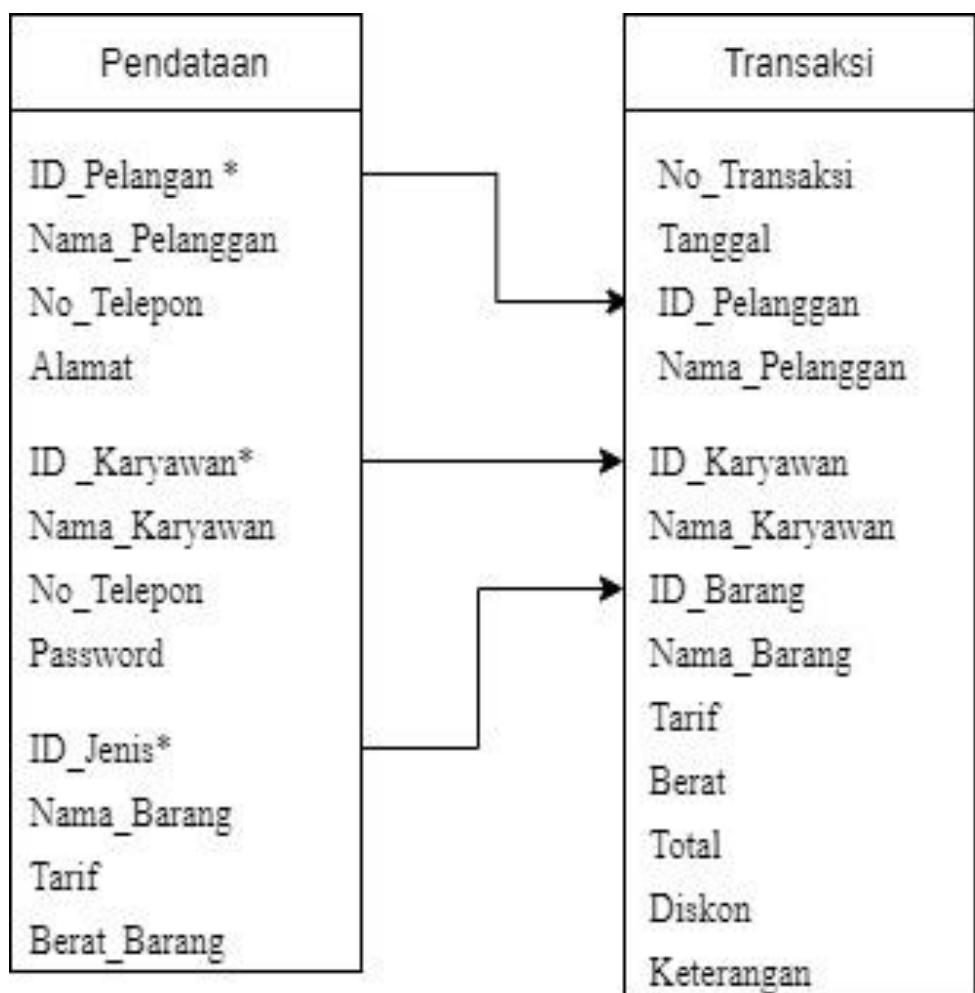
### **1. Normalisasi**

#### **a. Bentuk UnNormalisasi**

ID_Pelangan *	No_Transaksi
Nama_Pelanggan	Tanggal
No_Telepon	ID_Pelanggan
Alamat	Nama_Pelanggan
ID_Karyawan*	ID_Karyawan
Nama_Karyawan	Nama_Karyawan
No_Telepon	ID_Barang
Password	Nama_Barang
ID_Jenis*	Tarif
Nama_Barang	Berat
Tarif	Total
Berat_Barang	Diskon
	Keterangan

Gambar 4.30.  
Bentuk UnNormalisasi  
Sumber : Penulis (2021)

**b. Bentuk Normalisasi Pertama (NF1)**



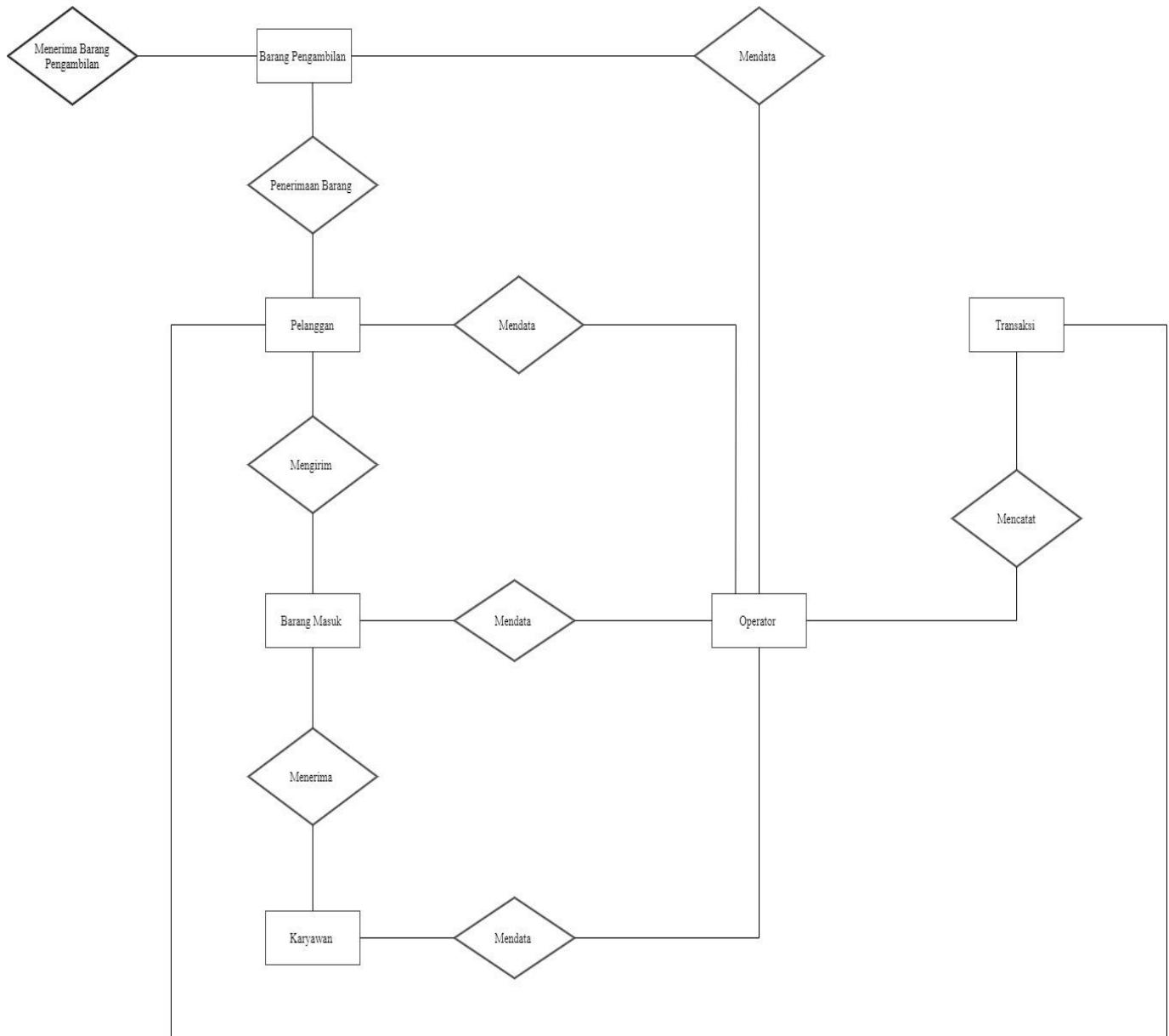
Gambar 4.31.  
Bentuk Normalisasi Pertama  
Sumber : Penulis (2021)

c. Bentuk Normalisasi Kedua (NF2)



Gambar 4.32  
Bentuk Normalisasi Kedua  
Sumber : Penulis (2021)

## 2. Entity Relationship Diagram (ERD)



Gambar 4.33.  
ERD yang Disusulkan  
Sumber : Penulis (2021)

### 3. Spesifikasi Proses

- a. Nama File : Pelanggan  
Media : Harddisk  
Organisasi : *Index Sequential*  
*Primary Key* : idplgn  
Panjang Record :  $20+50+35+50 = 155$  Byte  
Jumlah Record :  $155 \text{ byte} \times 30 \text{ hari} \times 12 \text{ bulan} \times 1 \text{ tahun} = 55.800 \text{ record}$

**Tabel 4.1.**  
**Data Pelanggan**

NO	Nama Field	Type	Size	Keterangan
1	idplgn	Varchar	20	ID Pelanggan
2	nmplgn	Varchar	50	Nama Pelanggan
3	notlp	Varchar	35	No Telepon Pelanggan
4	Alamat	Varchar	50	Alamat Pelanggan

Sumber : Penulis (2021)

- b. Nama File : Karyawan  
Media : Harddisk  
Organisasi : *Index Sequential*  
*Primary Key* : idkrywn  
Panjang Record :  $30+50+50+20 = 150$  Byte

Jumlah Record : 150 byte x 30 hari x 12 bulan x 1 tahun =  
54.000 record

**Tabel 4.2.**  
**Data Karyawan**

NO	Nama Field	Type	Size	Keterangan
1	idkrywn	Varchar	30	ID Karyawan
2	nmkrywn	Varchar	50	Nama Karyawan
3	notlp	Varchar	50	No Telepon
4	password	Varchar	20	Password

Sumber : Penulis (2021)

c. Nama File : Jenis Barang

Media : *Harddisk*

Organisasi : *Index Sequential*

*Primary Key* : idjenis

Panjang Record :  $50+50+50+20= 170$  Byte

Jumlah Record : 170 byte x 30 hari x 12 bulan x 1 tahun =  
61.200 record

**Tabel 4.3.**  
**Data Jenis Barang**

NO	Nama Field	Type	Size	Keterangan
1	idjenis	Varchar	25	ID Jenis Barang
2	nmbrg	Varchar	30	Nama Barang
3	tarif	Varchar	30	Tarif
4	brtbrg	Int	11	Berat Barang

Sumber : Penulis (2021)

d. Nama File : Transaksi

Media : *Database*

Organisasi : *Index Sequential*

*Index Key* : no\_transaksi

$$\begin{aligned} \text{Panjang Record} &: 20 + 20 + 20 + 25 + 50 + 20 + 30 + 15 + 100 \\ &+ 15 + 15 + 25 = 300 \text{ byte} \end{aligned}$$

$$\begin{aligned} \text{Jumlah Record} &: 300 \text{ byte} \times 30 \text{ hari} \times 12 \text{ bulan} \times 1 \text{ tahun} = \\ &108.000 \text{ record} \end{aligned}$$

**Tabel 4.4.**  
**Data Transaksi**

NO	Nama Field	Type	Size	Keterangan
1	no_transaksi	Varchar	20	No Transaksi
2	tanggal	Date	-	Tanggal Transaksi
3	id_pelanggan	int	20	Id Pelanggan
4	nama_pelanggan	Varchar	20	Nama Pelanggan

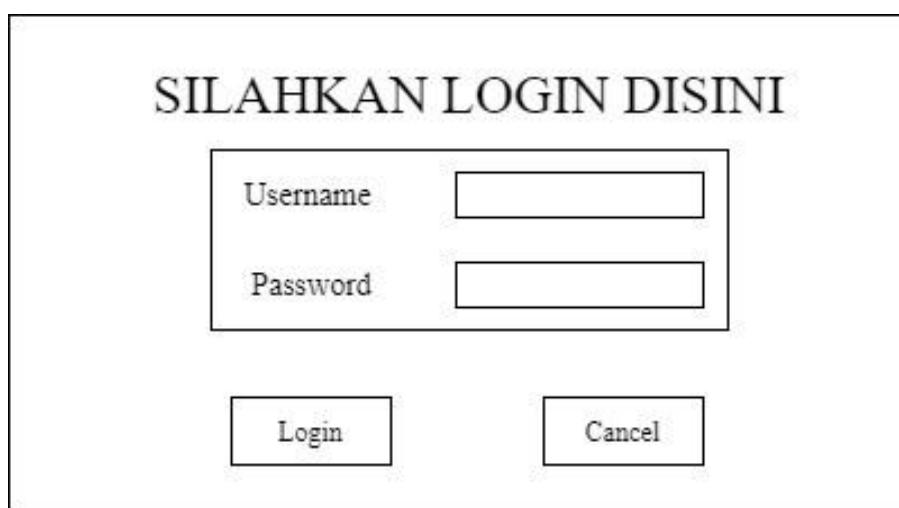
5	id_karyawan	Varchar	25	Id Karyawan
6	nama_karyawan	Varchar	50	Nama Karyawan
7	id_barang	int	20	Id Barang
8	nama_barang	Varchar	30	Nama Barang
9	tarif	int	15	Tarif
10	berat	int	100	Berat
11	total	int	15	Total
12	diskon	varchar	15	Diskon
13	keterangan	varchar	25	Keterangan

Sumber : Penulis (2021)

## **T. Rancangan Layar, Rancangan Form Masukan Data, dan Rancangan Keluaran**

### **1. Rancangan Layar**

#### **a. Login**



The diagram illustrates a login screen titled "SILAHKAN LOGIN DISINI". It features two input fields: "Username" and "Password", each accompanied by a rectangular input box. Below these fields are two buttons: "Login" on the left and "Cancel" on the right.

Gambar 4.34.  
Rancangan Layar Login  
Sumber : Penulis (2021)

Rancangan *Form Login* ini terdapat pada awal program. *Form login* digunakan sebagai kata kunci sebelum kita memasuki program utama. Agar tidak sembarang orang dapat mengakses program ini. Sehingga dalam *Form menu kerahasiaanya* dapat terjaga dengan baik. Apabila pengguna dapat memasukkan nama pengguna dan kata sandi pengguna dengan tepat, maka menu utama akan tampil dan program siap untuk disajikan.

**b. Menu Utama**



Gambar 4.35.  
Rancangan Layar Menu Utama  
Sumber : Penulis (2021)

Rancangan *form* menu utama adalah tampilan dasar dari program Pelayanan Jasa Laundry dimana terdapat beberapa menu/*Form* didalamnya, seperti Data Pelanggan, Data Karyawan, Data Jenis Barang, Transaksi, Barang, Laporan dan Keluar.

## 2. Rancangan Form Masukan Data

### a. Data Pelanggan

Data Karyawan																														
Id Karyawan	<input type="text"/>	<input type="button" value="Simpan"/>	<input type="button" value="Keluar"/>																											
Nama Karyawan	<input type="text"/>	<input type="button" value="Ubah"/>																												
No Telepon	<input type="text"/>	<input type="button" value="Hapus"/>																												
Password	<input type="text"/>	<input type="button" value="Batal"/>																												
<b>Cari Karyawan</b>																														
<input type="text"/>	<input type="button" value="Cari"/>																													
<table border="1"><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>																														

Gambar 4.36.  
Rancangan Form Data Pelanggan  
Sumber : Penulis (2021)

Rancangan *form* tampilan data pelanggan ini digunakan untuk meng-

*input* data pelanggan yang ingin dimasukkan ke dalam *database* pelanggan.

*Form* ini digunakan untuk pemanggilan pada *form* pelanggan.

### b. Data Karyawan

Data Karyawan									
Id Karyawan		<input type="text"/>		Simpan		<input type="button" value="Keluar"/>			
Nama Karyawan		<input type="text"/>		Ubah					
No Telepon		<input type="text"/>		Hapus					
Password		<input type="text"/>		Batal					
Cari Karyawan				<input type="button" value="Cetak"/>					
<input type="text"/>	<input type="button" value="Cari"/>								

Gambar 4.37.  
Rancangan Form Data Karyawan  
Sumber : Penulis (2021)

Rancangan *form* tampilan data karyawan ini digunakan untuk meng data karyawan yang ingin dimasukkan ke dalam *database* karyawan.  
*Form* ini digunakan untuk pemanggilan pada *form* karyawan.

### c. Data Jenis Barang

Jenis Barang								
Id Jenis		<input type="text"/>		Simpan		<input type="button" value="Keluar"/>		
Nama Barang		<input type="text"/>		Batal				
Tarif		<input type="text"/>		Ubah				
Berat Barang		<input type="text"/>		Hapus				
<b>Cari Jenis Barang</b> <input type="text"/> <input type="button" value="Cari"/>								

Gambar 4.38.  
Rancangan Form Jenis Barang  
Sumber : Penulis (2021)

Rancangan *form* tampilan data jenis barang ini digunakan untuk meng-  
*input* data jenis barang yang ingin dimasukkan ke dalam *database* jenis  
barang. *Form* ini digunakan untuk pemanggilan pada *form* jenis barang.

#### d. Transaksi

TRANSAKSI					
No Transaksi	<input type="text"/>				
Tanggal Transaksi	<input type="text"/>				
Id Pelanggan	<input type="text"/>	<input type="button" value="Cari"/>			
Nama Pelanggan	<input type="text"/>				
Id Karyawan	<input type="text"/>	<input type="button" value="Cari"/>	Total	<input type="text"/>	
Nama Karyawan	<input type="text"/>		Diskon	<input type="text"/>	
Id Jenis	<input type="text"/>	<input type="button" value="Cari"/>	Bayar	<input type="text"/>	
Nama Barang	<input type="text"/>		Uang Kembalian	<input type="text"/>	<input type="button" value="Keluar"/>
Tarif	<input type="text"/>				
Berat Barang (kg)	<input type="text"/>				
Subtotal	<input type="text"/>				
		<input type="button" value="Tambah"/>	<input type="button" value="Batal"/>	<input type="button" value="Kwitansi"/>	

Gambar 4.39.  
Rancangan Form Menu Transaksi  
Sumber : Penulis (2021)

Rancangan *form* menu transaksi ini digunakan untuk menginput data transaksi laundry yang ingin dimasukkan ke dalam *database* data transaksi. *Form* ini digunakan untuk pemanggilan pada *form* data transaksi.

**e. Menu Barang**

The diagram shows a form layout for managing laundry items. At the top left is a search input field with a 'Cari' (Search) button to its right. To the far right is a 'Kembali' (Return) button. Below these are three rows of four columns each, representing a grid for listing items. At the bottom right of the form area is a large rectangular button labeled 'AMBIL BARANG' (Take Item).

Gambar 4. 40.  
Rancangan Form Menu Barang  
Sumber : Penulis (2021)

Rancangan *form* tampilan data barang ini digunakan untuk meng-*input* data barang laundry baik yang ingin diambil maupun belum diambil yang akan dikembalikan kembali ke pelanggan. *Form* ini digunakan untuk pemanggilan pada *form* data transaksi.

## 2. Rancangan Keluaran

### a. Laporan Pelanggan

<b>RAMA LAUNDRY</b>			
Jl. Kenanga Bojong No. 22c Rt. 007 Rw. 019 Depok 16417			
<b>LAPORAN PELANGGAN</b>			
ID Pelanggan	Nama Pelanggan	No Telepon	Alamat

Depok 28 July 2021  
Pemilik

Dani Subardja

Gambar 4.41.  
Rancangan Keluaran Laporan Pelanggan  
Sumber : Penulis (2021)

Rancangan keluaran laporan pelanggan ini berisikan data dari pelanggan yang dipanggil sesuai dengan *database* pelanggan. Laporan ini akan dilihat oleh pemilik laundry, dan nantinya laporan ini akan diperhatikan oleh pemilik laundry untuk mengetahui pelanggan yang melaundry di laundry ini.

**b. Laporan Karyawan**

<p style="text-align: center;"><b>RAMA LAUNDRY</b></p> <p>Jl. Kenanga Bojong No. 22c Rt. 007 Rw. 019 Depok 16417</p>			
<p style="text-align: center;">LAPORAN KARYAWAN</p>			
ID Karyawan	Nama Karyawan	No Telepon	Password
<p style="text-align: right;">Depok 28 July 2021 Pemilik</p>			
<p style="text-align: right;">Dani Subardja</p>			

Gambar 4.42.  
Rancangan Keluaran Laporan Karyawan  
Sumber : Penulis (2021)

Rancangan keluaran laporan karyawan ini berisikan data dari karyawan yang dipanggil sesuai dengan *database* karyawan. Laporan ini akan dilihat oleh pemilik Laundry, dan nantinya laporan ini akan diperhatikan oleh pemilik Laundry untuk mengetahui karyawan yang bekerja di laundry ini.

**c. Laporan Barang Yang Sudah Diambil**

<b>RAMA LAUNDRY</b>											
Jl. Kenanga Bojong No. 22c Rt. 007 Rw. 019 Depok 16417											
LAPORAN TRANSAKSI BARANG YANG SUDAH DIAMBIL											
No. Transaksi	Tanggal	ID Pelanggan	Nama Pelanggan	ID Karyawan	Nama Karyawan	ID Barang	Nama Barang	Tarif	Berat	Total	Diskon

Depok 27 July 2021  
Pemilik  
  
Dani Subardja

Gambar 4.43.  
Rancangan Keluaran Laporan Barang Yang Sudah Diambil  
Sumber : Penulis (2021)

Rancangan keluaran laporan barang yang sudah diambil ini berisikan data dari transaksi yang dipanggil sesuai dengan barang masuk. Laporan ini akan dipelajari oleh pemilik Laundry, dan nantinya laporan ini akan dipelajari oleh pemilik Laundry untuk melihat perkembangan transaksi barang cucian yang sudah diambil setiap bulan.

**d. Laporan Barang Yang Belum Diambil**

<b>RAMA LAUNDRY</b>												
Jl. Kenanga Bojong No. 22c Rt. 007 Rw. 019 Depok 16417												
<b>LAPORAN TRANSAKSI BARANG YANG BELUM DIAMBIL</b>												
No. Transaksi	Tanggal	ID Pelanggan	Nama Pelanggan	Id Karyawan	Nama Karyawan	Id Barang	Nama Barang	Tarif	Berat	Total	Diskon	

Depok 28 July 2021  
Pemilik

Dani Subardja

Gambar 4.44.  
Rancangan Keluaran Laporan Barang Yang Belum Diambil  
Sumber : Penulis (2021)

Rancangan keluar tampilan laporan barang yang belum diambil ini berisikan data dari barang yang dipanggil sesuai dengan barang yang belum diambil. Laporan ini akan dipelajari oleh pemilik laundry, dan nantinya laporan ini akan dipelajari oleh pemilik laundry untuk melihat perkembangan barang yang belum diambil setiap bulan.

e. Laporan Transaksi

<b>RAMA LAUNDRY</b>												
Jl. Kenanga Bojong No. 22c Rt. 007 Rw. 019 Depok 16417												
<b>LAPORAN TRANSAKSI</b>												
No. Transaksi	Tanggal	ID Pelanggan	Nama Pelanggan	Id Karyawan	Nama Karyawan	Id Barang	Nama Barang	Tarif	Berat	Total	Diskon	Keterangan

Depok 28 July 2021  
Pemilik

Dani Subardja

Gambar 4.45.  
Rancangan Keluaran Laporan Transaksi  
Sumber : Penulis (2021)

Rancangan keluaran laporan transaksi ini berisikan data dari pelanggan, karyawan yang dipanggil sesuai dengan transaksi dari pelanggan. Laporan ini akan dipelajari oleh pemilik laundry, dan nantinya laporan ini akan dipelajari oleh pemilik laundry untuk melihat perkembangan transaksi setiap bulan.

**f. Bukti Pembayaran**

<p style="text-align: center;"><b>RAMA LAUNDRY</b></p> <p>Jl. Kenanga Bojong No. 22c Rt. 007 Rw. 019 Depok 16417</p>				
<p style="text-align: center;"><b>BUKTI PEMBAYARAN</b></p>				
ID Transaksi : Tanggal Transaksi :	ID Karyawan : Nama Karyawan :			
ID Pelanggan : Nama Pelanggan :				
ID Barang	Nama Barang	Tarif	Berat	Subtotal
Bayar :				
Kembalian :				

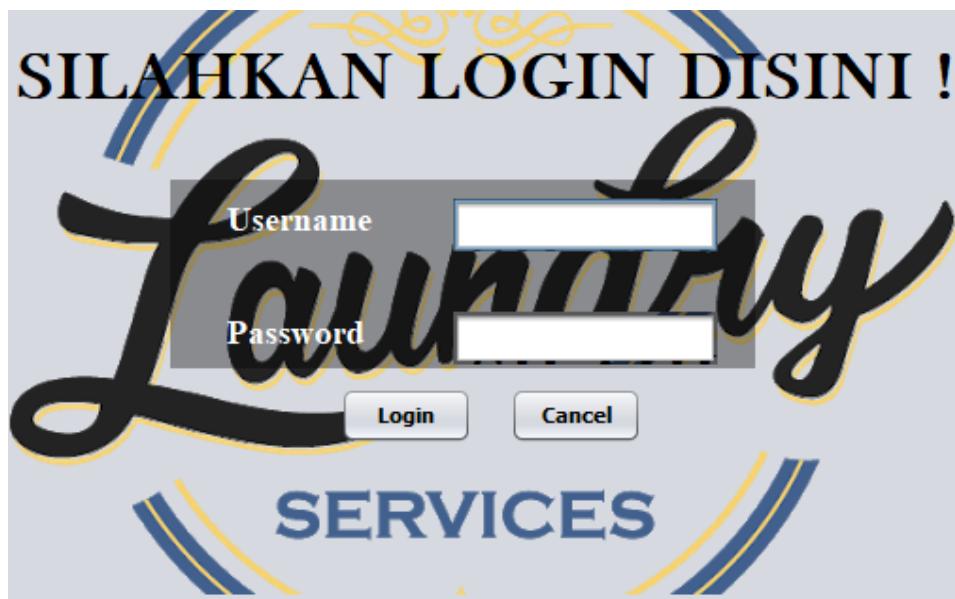
Gambar 4.46.  
Rancangan Keluaran Bukti Pembayaran  
Sumber : Penulis (2021)

Rancangan keluaran bukti pembayaran ini berisikan data dari operator yang dipanggil sesuai dengan transaksi yang telah dilakukan pelanggan dari laundry. Laporan ini akan diberikan dari laundry ke pelanggan setelah selesai melakukan transaksi, dan nantinya bukti pembayaran ini akan dipelajari oleh pelanggan jika ada ketidaksesuaian barang cucian yang dibayar dengan tanda terima.

## **U. Tampilan dan Penjelasan Layar, Tampilan Format Masukan, dan Tampilan Keluaran**

### **1. Tampilan Layar**

#### **a. Tampilan Menu Login**



Gambar 4.47.  
Tampilan Menu *Login*  
Sumber : Penulis (2021)

Tampilan menu *login* ini muncul di awal saat pengoperasian program sistem informasi pelayanan untuk diisi oleh operator laundry sebagai *user*. Masukan nama pengguna dan kata sandi yang sesuai dengan hak akses agar bisa mengoperasikan pengolahan data pelayanan jasa laundry. Jika nama pengguna dan kata sandi sesuai maka akan masuk kedalam tampilan menu utama.

### b. Tampilan Menu Utama



Gambar 4.48  
Tampilan Menu Utama  
Sumber : Penulis (2021)

Tampilan menu utama ini terdapat *form-form* yaitu data pelanggan, karyawan, jenis barang, transaksi, barang, laporan dan keluar.

Pada tampilan *form* menu utama ini terdapat banyak pilihan, jika pilih pelanggan maka akan masuk ke menu pelanggan, jika pilih karyawan maka akan masuk ke menu karyawan, jika pilih jenis barang maka akan masuk ke menu jenis barang, jika pilih transaksi maka akan masuk ke menu transaksi, jika pilih barang maka akan masuk ke menu barang, dan jika pilih menu laporan maka terdapat banyak pilihan laporan yang dapat kita cetak dengan format file PDF atau Word sesuai kebutuhan. Dan jika memilih menu keluar maka otomatis kita akan keluar dari aplikasi.

## 2. Tampilan Format Masukan

### a. Tampilan Data Pelanggan

The screenshot shows a Windows application window titled "PELANGGAN". The interface is designed for inputting customer data. At the top right is a "Keluar" button. The main area contains four input fields: "Id Pelanggan" (with value "001"), "Nama Pelanggan" (with value "Anita"), "No Telepon" (with value "0899778654"), and "Alamat" (with value "Jalan Hj. Arab"). To the right of each field are buttons: "Simpan", "Hapus", "Ubah", and "Batal". Below these fields is a "Cari Data Pelanggan" search bar with a "Cari" button. To the right of the search bar is a "Cetak" button. At the bottom of the window is a table with the following data:

Id Pelanggan	Nama Pelanggan	No Telepon	Alamat
001	Anita	0899778654	Jalan Hj. Arab

Gambar 4.49.  
Tampilan Data Pelanggan  
Sumber : Penulis (2021)

Tampilan data pelanggan ini digunakan untuk meng-*input* data pelanggan yang ingin dimasukkan ke dalam *database* pelanggan. *Form* ini digunakan untuk pemanggilan pada *form* pelanggan.

b. Tampilan Data Karyawan

The screenshot shows a Java Swing application window titled "Data Karyawan". The window has a decorative background of a sunset over water with a bridge. On the left, there is a vertical stack of four text input fields labeled "Id Karyawan", "Nama Karyawan", "No Telepon", and "Password". To the right of these fields are four buttons: "simpan", "ubah", "hapus", and "batal". At the top right is a "Kembali" button. Below the input fields is a search section with a "Cari Karyawan" label, a text input field, and a "Cari" button. A table below the search section displays one row of data:

ID Karyawan	Nama Karyawan	No.Telepon	Password
1	Devi	081266654321	admin

Gambar 4.50.  
Tampilan Data Karyawan  
Sumber : Penulis (2021)

Tampilan data karyawan ini digunakan untuk meng-*input* data karyawan yang ingin dimasukkan ke dalam *database* karyawan. *Form* ini digunakan untuk pemanggilan pada *form* karyawan.

c. Tampilan Data Jenis Barang

The screenshot shows a Windows application window titled "JENIS BARANG". The interface includes a form for inputting data with fields for "ID Jenis", "Nama Barang", "Tarif", and "Berat Barang / KG". There are also buttons for "Simpan", "Keluar", "Batal", "Ubah", and "Hapus". Below the form is a search bar labeled "Cari Jenis Barang" with a "Cari" button. A table displays existing data with columns "Id Jenis", "Nama Barang", "Tarif", and "Berat Barang/Qty".

Id Jenis	Nama Barang	Tarif	Berat Barang/Qty
001	Boneka	7000	1 kg
002	Selimut	5000	1 kg
003	Sarung	5000	1 kg
004	Jaket	10000	1 kg

Gambar 4.51.  
Tampilan Data Jenis Barang  
Sumber : Penulis (2021)

Tampilan data jenis barang ini digunakan untuk meng-*input* data jenis barang yang ingin dimasukkan ke dalam *database* jenis barang. *Form* ini digunakan untuk pemanggilan pada *form* jenis barang.

#### d. Tampilan Transaksi

TRANSAKSI						
No. Transaksi	TR-008	ID Jenis	Nama Barang	Tarif Perkilo	Berat	Sub Total
Tanggal Transaksi	28 July 2021					
ID Pelanggan	<input type="text"/>	<b>Cari</b>				
Nama Pelanggan	<input type="text"/>					
ID Karyawan	<input type="text"/>	<b>Cari</b>	Total	<input type="text"/>		
Nama Karyawan	<input type="text"/>		Diskon	<input type="text"/>		
ID Jenis	<input type="text"/>	<b>Cari</b>	Bayar	<input type="text"/>		
Nama Barang	<input type="text"/>		Uang Kembalian	<input type="text"/>		
Tarif	<input type="text"/> /KG					
Berat Barang (kg)	<input type="text"/>			<b>Tambah</b>	<b>Batal</b>	<b>Kwitansi</b>
Subtotal	<input type="text"/>					<b>Keluar</b>

Gambar 4.52.  
Tampilan Menu Transaksi  
Sumber : Penulis (2021)

Tampilan data transaksi ini digunakan untuk meng-*input* data transaksi jika ada pelanggan yang melakukan pembayaran ke laundry yang akan dimasukkan ke dalam *database* transaksi. *Form* ini digunakan untuk pemanggilan pada *form* transaksi.

### e. Tampilan Menu Barang

No Transaksi	Tanggal	ID Pelanggan	Nama Pelanggan	Keterangan
TR-001	2021-07-18	1	Salsha	Sudah Diambil
TR-002	2021-07-18	2	Devi	Sudah Diambil
TR-003	2021-07-18	1	Salsha	Sudah Diambil
TR-004	2021-07-19	2	Devi	Sudah Diambil
TR-005	2021-07-19	1	Salsha	Belum Diambil
TR-006	2021-07-20	1	Salsha	Belum Diambil
TR-007	2021-07-28	1	Anita	Belum Diambil

Gambar 4.53.  
Tampilan Barang  
Sumber : Penulis (2021)

Tampilan menu barang ini digunakan untuk menampilkan data barang cucian yang sudah diambil dan yang belum diambil. dan membuat data yang ingin dimasukkan ke dalam *database* transaksi.. *Form* ini digunakan untuk pemanggilan pada *form* data transaksi.

**3. Tampilan Keluaran**  
**a. Laporan Pelanggan**

## **Rama Laundry**

Jl. Kenanga Bojong No.22c Rt. 007 Rw.019 Depok 16417

---

### **LAPORAN PELANGGAN**

---

ID Pelanggan	Nama Pelanggan	No. Telp	Alamat
001	Anita	0899 778 654	Jalan Hj. Amb

Depok 28 July 2021  
Penulis

Deni Subardja

Gambar 4.54.  
Laporan Pelanggan  
Sumber : Penulis (2021)

Tampilan laporan pelanggan ini berisikan data dari pelanggan yang dipanggil sesuai dengan *database* pelanggan. Laporan ini akan dilihat oleh pemilik laundry, dan nantinya laporan ini akan diperhatikan oleh pemilik laundry untuk mengetahui pelanggan yang ada di Rama Laundry ini.

**b. Laporan Karyawan**

## **Rama Laundry**

Jl. Kenanga Bojong No.22c Rt. 007 Rw.019 Depok 16417

---

### LAPORAN KARYAWAN

---

ID Karyawan	Nama Karyawan	No. Telp	Password
1	Deni	081266654321	admin

Depok 28 July 2021

Pemilik

Dani Sulendja

---

Gambar 4.55.  
Laporan Karyawan  
Sumber : Penulis (2021)

Tampilan laporan karyawan ini berisikan data dari karyawan yang dipanggil sesuai dengan *database* karyawan. Laporan ini akan dilihat oleh pemilik laundry, dan nantinya laporan ini akan diperhatikan oleh pemilik laundry untuk mengetahui karyawan yang bekerja di Rama Laundry ini.

c. Laporan Barang Yang Sudah Diambil

## Rama Laundry

Jl. Kenanga Bojong No.22c Rt. 007 Rw.019 Depok 16417

### LAPORAN TRANSAKSI BARANG YANG SUDAH DIAMBIL

No Transaksi	Tanggal	ID Pelanggan	Nama Pelanggan	ID Karyawan	Nama Karyawan	ID Barang	Nama Barang	Tarif	Berat	Total	Diskon
TR-001	18 July 2021	1	Salsha	003	Salsha	2	Cuci Pakaian	5000	2 Kg	10000	10%

Depok 28 July 2021  
Pemilik

Dani Subardja

Gambar 4.56.  
Laporan Barang Yang Sudah Diambil  
Sumber : Penulis (2021)

Tampilan laporan barang yang sudah diambil ini berisikan data yang dipanggil sesuai dengan barang transaksi. Laporan ini akan dipelajari oleh pemilik laundry untuk melihat perkembangan barang setiap bulan.

**d. Laporan Barang Yang Belum Diambil**

## **Rama Laundry**

Jl. Kenanga Bojong No.22c Rt. 007 Rw.019 Depok 16417

### **LAPORAN TRANSAKSI BARANG YANG BELUM DIAMBIL**

No Transaksi	Tanggal	ID Pelanggan	Nama Pelanggan	ID Karyawan	Nama Karyawan	ID Barang	Nama Barang	Tarif	Berat	Total	Diskon
TR-005	19 July 2021	1	Salsha	005	Baginda Ratu Monica Regita	4	Jaket	10000	2 Kg	20000	10%
TR-005	19 July 2021	1	Salsha	005	Baginda Ratu Monica Regita	2	Cuci Pakaian	5000	2 Kg	10000	10%

Depok 28 July 2021  
Pemilik

Dani Subardja

**Gambar 4.57.**  
**Laporan Barng Yang Belum Diambil**  
**Sumber : Penulis (2021)**

Tampilan laporan barang yang belum diambil ini berisikan data yang dipanggil sesuai dengan barang transaksi. Laporan ini akan dipelajari oleh pemilik laundry untuk melihat perkembangan barang setiap bulan.

e. Laporan Transaksi

## Rama Laundry

Jl. Kenanga Bojong No.22c Rt. 007 Rw.019 Depok 16417

### LAPORAN TRANSAKSI

No Transaksi	Tanggal	ID Pelanggan	Nama Pelanggan	ID Karyawan	Nama Karyawan	ID Karyawan	Nama Barang	Tarif	Berat	Total	Diskon	Keterangan
TR-008	28 July 2021	1	Anita	1	Devi	3	Sarung	5000	3	15000	10%	Belum Diambil
TR-007	28 July 2021	1	Anita	1	Devi	1	Boneka	7000	2	14000	10%	Belum Diambil
TR-006	20 July 2021	1	Salsha	1	admin	1	Boneka	7000	2	14000	10%	Belum Diambil

Depok 28 July 2021

Pemilik

Dani Subardja

Gambar 4.58.  
Laporan Transaksi  
Sumber : Penulis (2021)

Tampilan laporan transaksi ini berisikan data dari operator yang dipanggil sesuai dengan transaksi dari pelanggan. Laporan ini akan dipelajari oleh pemilik laundry, dan laporan ini akan dipelajari untuk melihat perkembangan transaksi setiap bulan ke Rama Laundry.

**f. Bukti Pembayaran**

## **Rama Laundry**

Jl. Kenanga Bojong No.22c Rt. 007 Rw.019 Depok 16417

### **BUKTI PEMBAYARAN**

ID Transaksi : TR-008	ID Karyawan : 1
Tgl Transaksi : 28 July 2021	Nama Karyawan: Devi
ID Pelanggan : 1	
Nama : Anita	

ID Barang	Nama Barang	Tarif	Berat	Sub Total
3	Singg	5000	3	15.000

Bayar : Rp. 15.000  
Kembalian : Rp. 0

Gambar 4.59.  
Bukti Pembayaran  
Sumber : Penulis (2021)

Tampilan bukti pembayaran ini berisikan data dari laporan transaksi yang dipanggil sesuai dengan *database* transaksi. Laporan ini akan dilihat oleh pelanggan laundry dan laporan ini akan diperhatikan oleh pelanggan untuk mengetahui kesesuaian data yang dibeli dengan bukti pembayaran yang diberikan oleh Rama Laundry.

## **BAB V PENUTUP**

### **A. Simpulan**

Berdasarkan pembahasan yang telah dilakukan, maka dapat diambil beberapa kesimpulan, diantaranya sebagai berikut:

1. Sistem informasi pelayanan jasa laundry yang sedang berjalan Rama Laundry Depok masih dilakukan pengelahan data yang dilakukan diantaranya pendataan barang masuk, data jenis barang, transaksi, dan pembuatan laporan-laporan. Sistem yang digunakan saat ini masih dengan cara manual dengan menggunakan buku besar dan ditulis tangan.
2. Pengembangan sistem aplikasi jasa laundry pada Rama Laundry Depok dilakukan dengan menggunakan aplikasi berbasis dekstop menggunakan bahasa pemograman *java* dan penyimpanan data pada media *database MySQL*. Dengan adanya aplikasi yang dibuat ini diharapkan dapat mempermudah dalam pendataan data barang laundry dan data jenis barang. Proses transaksi yang ditunjang dengan adanya aplikasi laundry dapat dilakukan dengan lebih efektif dan efisien, sehingga proses tersebut tidak akan memakan waktu yang lama dan memperkecil kemungkinan hilangnya data transaksi yang disebabkan oleh rusak atau hilangnya dokumen transaksi. Informasi buku yang tersedia menjadi mudah dicari dengan cepat, karena petugas dapat mencari transaksi dengan otomatis tanpa harus mencari langsung

mencari di buku besar. Penyajian berbagi bentuk laporan dari bagian laundry yang sudah dilakukan secara otomatis tentu akan memudahkan dalam menyampaikan informasi kepada pemilik laundry

## B. Saran

Sistem informasi pelayanan jasa laundry yang telah penulis buat ini terbuka untuk dapat dikembangkan lagi oleh peneliti lain pada tahap penelitian selanjutnya.

Terdapat beberapa saran bagi peneliti selanjutnya berkaitan dengan pengembangan sistem informasi pelayanan jasa laundry ini, antara lain:

1. Aplikasi jasa laundry yang penulisan dibuat masih dalam bentuk aplikasi berbentuk dekstop, untuk tahap pengembangan selanjutnya diharapkan penelitian lain dapat mengembangkan sistem informasi pelayanan jasa laundry dalam aplikasi yang berbentuk web.
2. Perlunya pengadaan sarana dan prasarana yang menunjang dalam pemakaian aplikasi jasa laundry ini, seperti adanya fasilitas *hardware* dan *software* yang akan menunjang pemakaian aplikasi laundry ini serta pengarahan kepada staff laundry dalam penggunaan aplikasi ini, supaya penggunaan aplikasi laundry akan disarankan lebih optimal.

## **DAFTAR PUSTAKA**

- Chan. (2017). *Membuat Aplikasi Database dengan PowerBuileder 12.6 dan MySQL*. Jakarta: PT Alex Media Komputindo.
- Hastuti, Dwi dan Arief, Yusril. (2018). *Belajar Pemrograman Java Menggunakan Java FX dan Database H2*. Yogyakarta: Aswaja Pressindo.
- Hutahaean. (2014). *Konsep Sistem Informasi*. Yogyakarta: CV Budi Utama.
- Jogiyanto. (2005). *Analisis dan Desain Sistem Informasi*. Yogyakarta: Andi.
- Kristanto, Andri. (2018). *Perancangan Sistem Informasi dan Aplikasinya*. Yogyakarta: Gava Media.
- Maniah, dan Hamidin, Dini. (2017). *Analisis dan Perancangan Sistem Informasi Pembahasan Secara Praktis Dengan Contoh Kasus*. Yogyakarta: CV B1udi Utama.
- Muslihudin, Muhamad & Oktafianto. (2016). *Analisis Perancangan Sistem Informasi Menggunakan Model Terstruktur dan UML*.Yogyakarta: Cv Andi Offset.
- Nofriadi. (2015). *Java Fundamental Dengan Netbeans 8.0.2*. Yogyakarta: CV Budi Utama.
- Supardi, Ir Yuniar. (2010). *Semua Bisa Menjadi: Programmer Java Basic Programming*. Jakarta: PT Elex Media Komputindo.
- Yanto, Robi. (2016). *Manajemen Basis Data Menggunakan MySQL*. Yogyakarta: Deepublish.

## **DAFTAR RIWAYAT HIDUP PENULIS**

**Devianti Nur Safitri** , lahir di Jakarta beserta seluruh isinya adalah benar-benar karya saya sendiri. Saya tidak melakukan penjiplakan atau pengutipan dengan cara-cara yang tidak sesuai dengan etika ilmu yang berlaku dalam masyarakat keilmuan. Atas pernyataan ini, saya siap menanggung risiko/sanksi apabila di kemudian hari ditemukan adanya pelanggaran etika keilmuan atau ada klaim dari pihak lain terhadap keaslian karya saya ini sesuai dengan Undang-Undang Republik Indonesia Nomor 20 Tahun 2003 tentang Sistem Pendidikan Nasional Bab V Pasal 25 ayat 2.

**PERMOHONAN DOSEN PEMBIMBING TUGAS AKHIR/SKRIPSI  
SEMESTER GENJAL/GENAP TAHUN AKADEMIK 2021/2022  
PRODI : INFORMATIKA, T. INDUSTRI, ARSITEKTUR**

Yang bertanda tangan di bawah ini :

Nama : DEVIANTI NUR SAFITRI  
NPM : 201943501222  
Tempat/tgl Lahir : JAYARTA, 22 DESEMBER 1999  
Alamat : JL. KEBEMBEM PAYA RT.05 RW.02 NO.08  
KEL. ABADI JAYA, KEC. SUKMADJAYA KOTA DEPOK  
Telp/HP : 0812 1554 7020

Dalam rangka melengkapi sebagian dari tugas dan persyaratan untuk memperoleh gelar sarjana dari Universitas Indraprasta PGRI, bersama ini saya mengajukan judul skripsi/tugas akhir :

1. APLIKASI SISTEM INFORMASI PELAYANAN JASA LAUNDRTY  
PADA RAMA LAUNDRTY DEPOK BERBASIS JAVA NETBEANS.
2. ....

Mohon kiranya dapat ditetapkan dosen pembimbing skripsi/tugas akhir.

Sebagai kelengkapan pengajuan ini terlampir kami sampaikan :

1. KRS yang memuat mata Kuliah Tugas Akhir/Skripsi dari PA/Ketua Program Studi
2. Proposal Tugas Akhir yang telah diseminarkan
3. Rekap KHS/Transkrip nilai dengan akumulasi perolehan sks  $\geq$  130 sks dengan IPK  $\geq$  2.0 dan ditandatangani oleh ketua Program Studi (khusus daftar sidang)
4. Bukti telah lulus mata kuliah Seminar (Minimal C) (Kecuali T. Informatika)
5. Bukti telah lulus mata kuliah kerja Praktek atau mata kuliah yang berkaitan dengan materi penelitian (Minimal C)

Demikian kiranya mendapatkan persetujuan dan atas perhatian untuk ini diucapkan terima kasih.

Mengetahui  
Ketua Program Studi

(...Mei Lectari, M.Kom.)

Pembimbing Materi

(Dewi Anjani, M.Kom.)

Jakarta, 03 MARET 2021

Mahasiswa yang bersangkutan

(...DEVIANTI, N.S....)  
201943501222

Pembimbing Teknik

(...ENAYAT, SEPTIANA....)

**FAKULTAS TEKNIK DAN ILMU KOMPUTER**

Jl. Nangka No. 58C, Tanjung Barat, Jagakarsa - Jakarta Selatan 12530,  
 Telp. (021) 78835283, 7818718 Fax. (021) 29121071, 29121070  
 Jl. Raya Tengah No. 80 Kelurahan Gedong, Pasar Rebo - Jakarta Timur 13670,  
 Telp. (021) 87797409

Website : [www.unindra.ac.id](http://www.unindra.ac.id)  
 Email : [kampus@unindra.ac.id](mailto:kampus@unindra.ac.id)

### **KARTU ASISTENSI BIMBINGAN SKRIPSI/TUGAS AKHIR**

Nama : Devianti Nur Safitri  
 NPM : 201743501222  
 No. Telepon : 0812 1554 7920  
 Fakultas/Program Studi : FTIK / Teknik Informatika  
 Judul Skripsi : Aplikasi Sistem Informasi Pelayanan Jasa Laundry pada Rama Laundry Depok Berbasis Java Netbeans  
 Dosen Pembimbing Materi : *Dewi Anjani, M.Kom*  
 Dosen Pembimbing Teknik : *Eddy Saputra, M.Pd.I*

Jadwal Bimbingan :

NO	Tanggal	Uraian	Paraf Pembimbing
1	18/05/2021	ACC + Judul	R
2	20/05/2021	konsul bab 1 - 3	R
3	27/05/2021	revisi bab 1 - 3	R
4	01/06/2021	konsul bab 4 - 5	R
5	03/06/2021	revisi bab 4	R
6	05/06/2021	konsul bab 4 lanjut	R
7	15/06/2021	Revisi bab 4 - 5	R
8	06/07/2021	Acc Sidang	R
9			
10			
11			
12			
13			
14			

## FAKULTAS TEKNIK DAN ILMU KOMPUTER

Jl. Nangka No. 58C, Tanjung Barat, Jagakarsa - Jakarta Selatan 12530,  
Telp. (021) 78835283, 7818718 Fax. (021) 29121071, 29121070

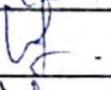
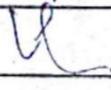
Jl. Raya Tengah No. 80 Keturahan Gedong, Pasar Rebo - Jakarta Timur 13670,  
Telp. (021) 87797409

Website : [www.unindra.ac.id](http://www.unindra.ac.id)  
Email : kampus@unindra.ac.id

**KARTU ASISTENSI BIMBINGAN SKRIPSI/TUGAS AKHIR**

Nama : Devianti Nur Safitri  
NPM : 201743501222  
No. Telepon : 0812 1554 7920  
Fakultas/Program Studi : FTIK / Teknik Informatika  
Judul Skripsi : Aplikasi Sistem Informasi Pelayanan Jasa Laundry pada Rama Laundry Depok Berbasis Java Netbeans  
Dosen Pembimbing Materi : *Dewi Anjani, M.Kom*  
Dosen Pembimbing Teknik : *Eddy Saputra, M.Pd.I*

**Jadwal Bimbingan :**

NO	Tanggal	Uraian	Paraf Pembimbing
1	20/5/2021	Acc Judul	
2	18/6/2021	Review bab 1-3 (skripsi)	
3	17/6/2021	u. bab 4	
4	15/7/2021	u. bab 4-5	
5	19/8/2021	Acc Skripsi	
6			
7			
8			
9			
10			
11			
12			
13			
14			



**FAKULTAS TEKNIK DAN ILMU KOMPUTER**

Jl. Nangka No. 58C, Tanjung Barat, Jagakarsa - Jakarta Selatan 12530,  
Telp. (021) 78835283, 7818718 Fax. (021) 29121071, 29121070

Jl. Raya Tengah No. 80 Kelurahan Gedong, Pasar Rebo - Jakarta Timur 13670,  
Telp. (021) 87797409

Website : [www.unindra.ac.id](http://www.unindra.ac.id)  
Email : kampus@unindra.ac.id

Nomor : 106 / FTIK / UNINDRA / IV / 2021

Jakarta, 18 April 2021

Lampiran : -

Hal : Permohonan untuk  
Mengadakan Penelitian

Kepada Yth.

**Pimpinan Rama Laundry Depok**  
**di Tempat**

Dengan hormat,

Salam sejahtera untuk kita semua, semoga selalu mendapat rahmat dan hidayah-Nya, Amin. Dalam rangka persiapan penulisan karya ilmiah/skripsi/tugas akhir yang merupakan salah satu syarat untuk menyelesaikan studi akhir pada Universitas Indraprasta PGRI diperlukan data primer sebagai bahan penulisan.

Sehubungan dengan hal tersebut pimpinan Fakultas Teknik dan Ilmu Komputer Universitas Indraprasta PGRI dengan ini menerangkan bahwa :

Nama	: Devianti Nur Safitri
NPM	: 201743501222
Fakultas	: FTIK
Program Studi	: Teknik Informatika
Jenjang	: Strata Satu (S1)

Adalah benar mahasiswa Universitas Indraprasta PGRI pada Fakultas Teknik dan Ilmu Komputer. Oleh karena itu kami mohon izin atas nama mahasiswa tersebut untuk mengadakan penelitian pada lembaga/instansi/perusahaan yang Bapak/Ibu pimpin guna mendapat data yang diperlukan, sebagai bahan dalam penulisan skripsi/tugas akhir yang berjudul :

*Aplikasi Sistem Informasi Pelayanan Jasa Laundry pada Rama Laundry Depok Berbasis Java Netbeans*

Demikian permohonan ini disampaikan agar dapat dipertimbangkan dan disetujui. Atas perhatian dan kerjasama yang Bapak/Ibu berikan, kami mengucapkan terima kasih.

Universitas Indraprasta PGRI

Wakil Dekan



# **RAMA LAUNDRY**

Jl. Kenanga Bojong No. 22c Rt. 007 Rw. 019 Abadijaya, Kec. Sukmajaya, Kota Depok, Jawa Barat 16417  
Telepon (08998089742)

---

Nomor : 020/RAMA/IV/2021

Lampiran : -

Hal : Balasan Permohonan Untuk Mengadakan Penelitian

Kepada Yth.

Kepala Fakultas Teknik dan Ilmu Komputer

Universitas Indraprasta PGRI

Di

Tempat

Dengan Hormat

Berdasarkan Surat Nomor : 106 / FTIK / UNINDRA / IV / 2021 Tanggal 18 April 2021 Perihal Permohonan Untuk Mengadakan Penelitian kepada Mahasiswa :

Nama : Devianti Nur Safitri  
NPM : 201743501222  
Fakultas : Fakultas Teknik dan Ilmu Komputer  
Program Studi : Teknik Informatika  
Jenjang : Strata Satu (S1)

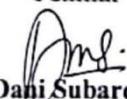
Telah kami setujui untuk melaksanakan penelitian pada laundry kami sebagai syarat untuk menyelesaikan studi akhir / Skripsi dengan judul :

**“Aplikasi Sistem Informasi Pelayanan Jasa Laundry Pada Rama Laundry Berbasis Java Netbeans”**

Demikian pemberitahuan kami, atas perhatian dan kerjasamanya kami mengucapkan terima kasih.

Depok, 20 April 2021

Pemilik

  
Dani Subardja

## Listing Program

### A. Koneksi.java

```
1. package Koneksi;
2. import java.sql.*;
3. /**
4. *
     • @author USER
5. */
6. public class koneksi {
7. private Connection koneksi;
8. public Connection connect(){
9. try{
10. Class.forName("com.mysql.jdbc.Driver");
11. System.out.println("Berhasil Konek");
12. }
13. catch(ClassNotFoundException ex){
14. System.out.println("gagal koneksi"+ex);
15. }
16. String url = "jdbc:mysql://localhost/laundry";
17. try{
18. koneksi = DriverManager.getConnection(url,"root","");
19. System.out.println("Berhasil Konek Database");
20. }
21. catch (SQLException ex){
22. System.out.println("Gagal Koneksi Database"+ex);
23. }
24. return koneksi;
25. }
26. }
```

### B. Login.java

```
package Form;
1. import java.sql.*;
2. import javax.swing.JOptionPane;
3. import Koneksi.koneksi;
4. import java.awt.Color;
5. import java.awt.event.ActionEvent;
6. import java.awt.event.KeyEvent;
7. import java.util.Arrays;

8. /**
9. *
     @author USER
10. */
11. public class login extends javax.swing.JFrame {
12. private final Connection conn = new koneksi().connect();

13. /**
     Creates new form login
14. */
15. public login() {
16. initComponents();
17. jPanel1.setBackground(new Color(0, 0, 0, 92));
18. }
```

```

19. /**
   This method is called from within the constructor to initialize the form.
   WARNING: Do NOT modify this code. The content of this method is always
   regenerated by the Form Editor.
20. */
21. @SuppressWarnings("checked")
22. // <editor-fold defaultstate="collapsed" desc="Generated Code">
23. private void initComponents() {

24.     jPanel1 = new javax.swing.JPanel();
25.     txtuser = new javax.swing.JTextField();
26.     fpass = new javax.swing.JPasswordField();
27.     jLabel3 = new javax.swing.JLabel();
28.     jLabel2 = new javax.swing.JLabel();
29.     jLabel1 = new javax.swing.JLabel();
30.     masuk = new javax.swing.JButton();
31.     bcancel = new javax.swing.JButton();
32.     jLabel6 = new javax.swing.JLabel();

33.     setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
34.     getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

35.     jPanel1.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());
36.     jPanel1.add(txtuser, new org.netbeans.lib.awtextra.AbsoluteConstraints(150, 10, 140, -1));

37.     fpass.addKeyListener(new java.awt.event.KeyAdapter() {
38.         public void keyPressed(java.awt.event.KeyEvent evt) {
39.             fpassKeyPressed(evt);
40.         }
41.     });
41.     jPanel1.add(fpass, new org.netbeans.lib.awtextra.AbsoluteConstraints(150, 70, 140, -1));

42.     jLabel3.setFont(new java.awt.Font("Times New Roman", 1, 18)); // NOI18N
43.     jLabel3.setForeground(new java.awt.Color(255, 255, 255));
44.     jLabel3.setText("Password");
45.     jPanel1.add(jLabel3, new org.netbeans.lib.awtextra.AbsoluteConstraints(30, 70, -1, -1));

46.     jLabel2.setFont(new java.awt.Font("Times New Roman", 1, 18)); // NOI18N
47.     jLabel2.setForeground(new java.awt.Color(255, 255, 255));
48.     jLabel2.setText("Username");
49.     jPanel1.add(jLabel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(30, 10, -1, -1));

50.     getContentPane().add(jPanel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(90, 90, 310, 100));

51.     jLabel1.setFont(new java.awt.Font("Baskerville Old Face", 1, 36)); // NOI18N
52.     jLabel1.setText("SILAHKAN LOGIN DISINI !");
53.     getContentPane().add(jLabel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(8, 20, 500, -1));

54.     masuk.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N
55.     masuk.setText("Login");
56.     masuk.addActionListener(new java.awt.event.ActionListener() {

```

```

57. public void actionPerformed(java.awt.event.ActionEvent evt) {
58.     masukActionPerformed(evt);
59. });
60. getContentPane().add(masuk, new org.netbeans.lib.awtextra.AbsoluteConstraints(180,
61.     200, 70, 30));

61. bcancel.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N
62. bcancel.setText("Cancel");
63. bcancel.addActionListener(new java.awt.event.ActionListener() {
64.     public void actionPerformed(java.awt.event.ActionEvent evt) {
65.         bcancelActionPerformed(evt);
66.     }
67. });
68. getContentPane().add(bcancel, new org.netbeans.lib.awtextra.AbsoluteConstraints(270,
69.     200, 70, 30));

70. jLabel6.setIcon(new
71.     javax.swing.ImageIcon(getClass().getResource("/img/logo_laundry.png"))); // NOI18N
72. getContentPane().add(jLabel6, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0,
73.     510, 310));

74. setSize(new java.awt.Dimension(527, 359));
75. setLocationRelativeTo(null);
76. } // </editor-fold>

77. private void bcancelActionPerformed(java.awt.event.ActionEvent evt) {
78.     dispose();
79. }

80. private void masukActionPerformed(java.awt.event.ActionEvent evt) {
81.     try {
82.         String sql = "select * from karyawan where idkrywn='"+txtuser.getText()+"'AND
83.             password='"+String.valueOf(fpass.getPassword())+"'";
84.         Statement stat = conn.createStatement();
85.         ResultSet hasil = stat.executeQuery(sql);
86.         if(hasil.next()){
87.             UserID.setUserLogin(hasil.getString("idkrywn"));
88.             JOptionPane.showMessageDialog(null, "Berhasil Login");
89.             this.setVisible(false);
90.             new menu().setVisible(true);
91.         } else{
92.             JOptionPane.showMessageDialog(null, "Username atau Password salah");
93.         }
94.     } catch (SQLException e){
95.         JOptionPane.showMessageDialog(null, "Gagal Login !!!" +e);
96.     }
97. }

98. private void fpassKeyPressed(java.awt.event.KeyEvent evt) {
99.     if (evt.getKeyCode() == KeyEvent.VK_ENTER) {

```

```

92. masukActionPerformed(new ActionEvent(evt.getSource(), evt.getID(), "Key Press
   proses"));
93. }
94. }

95. /**
   @param args the command line arguments
96. */
97. public static void main(String args[]) {
98. /* Set the Nimbus look and feel */
99. //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
100. /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and
   feel.
   For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
101. */
102. try {
103. for (javax.swing.UIManager.LookAndFeelInfo info :
   javax.swing.UIManager.getInstalledLookAndFeels()) {
   if ("Nimbus".equals(info.getName())) {
   javax.swing.UIManager.setLookAndFeel(info.getClassName());
   break;
   }
104. }
105. } catch (ClassNotFoundException | InstantiationException | IllegalAccessException |
   javax.swing.UnsupportedLookAndFeelException ex) {
106. java.util.logging.Logger.getLogger(login.class.getName()).log(java.util.logging.Level.SEVERE,
   null, ex);
107. }
108. //</editor-fold>

109. //</editor-fold>

110. /* Create and display the form */
111. java.awt.EventQueue.invokeLater(() -> {
112. new login().setVisible(true);
113. });
114. }

115. // Variables declaration - do not modify
116. private javax.swing.JButton bcancel;
117. private javax.swing.JPasswordField fpass;
118. private javax.swing.JLabel jLabel1;
119. private javax.swing.JLabel jLabel2;
120. private javax.swing.JLabel jLabel3;
121. private javax.swing.JLabel jLabel6;
122. private javax.swing.JPanel jPanel1;
123. private javax.swing.JButton masuk;
124. private javax.swing.JTextField txtuser;
125. // End of variables declaration
126. }

```

### C. Menu.java

```
1. package Form;
2. import java.sql.*;
3. import java.util.HashMap;
4. import javax.swing.JOptionPane;
5. import net.sf.jasperreports.engine.JasperFillManager;
6. import net.sf.jasperreports.engine.JasperPrint;
7. import net.sf.jasperreports.view.JasperViewer;
8. import Koneksi.koneksi;
9. import java.io.File;
10. import javax.swing.ImageIcon;
11. import net.sf.jasperreports.engine.JRException;
12. import net.sf.jasperreports.engine.JasperCompileManager;
13. import net.sf.jasperreports.engine.JasperReport;
14. import net.sf.jasperreports.engine.design.JasperDesign;
15. import net.sf.jasperreports.engine.xml.JRXmlLoader;

16. /**
17. *
18. * @author USER
19. */
20. public class menu extends javax.swing.JFrame {
21.     private final Connection conn = new koneksi().connect();

22.     /**
23.      * Creates new form menu
24.     */
25.     public menu() {
26.         initComponents();
27.         ImageIcon img = new ImageIcon("src/img/red.jpg");
28.         setIconImage(img.getImage());
29.     }

30.     /**
31.      * This method is called from within the constructor to initialize the form.
32.      * WARNING: Do NOT modify this code. The content of this method is always
33.      * regenerated by the Form Editor.
34.     */
35.     @SuppressWarnings("unchecked")
36.     // <editor-fold defaultstate="collapsed" desc="Generated Code">
37.     private void initComponents() {

38.         jLabel3 = new javax.swing.JLabel();
39.         jLabel2 = new javax.swing.JLabel();
40.         jLabel4 = new javax.swing.JLabel();
41.         jMenuBar2 = new javax.swing.JMenuBar();
42.         jMenu7 = new javax.swing.JMenu();
43.         jMenuItem1 = new javax.swing.JMenuItem();
44.         jMenuItem2 = new javax.swing.JMenuItem();
45.         jMenuItem5 = new javax.swing.JMenuItem();
46.         jMenuItem8 = new javax.swing.JMenu();
47.         jMenuItem4 = new javax.swing.JMenuItem();
48.         jMenuItem6 = new javax.swing.JMenuItem();
```

```
44. jMenu1 = new javax.swing.JMenu();
45. jMenuItem9 = new javax.swing.JMenuItem();
46. jMenuItem3 = new javax.swing.JMenuItem();
47. jMenu3 = new javax.swing.JMenu();
48. jMenuItem10 = new javax.swing.JMenuItem();
49. jMenuItem11 = new javax.swing.JMenuItem();
50. jMenuItem12 = new javax.swing.JMenuItem();
51. jMenu2 = new javax.swing.JMenu();
52. jMenuItem7 = new javax.swing.JMenuItem();
53. jMenuItem8 = new javax.swing.JMenuItem();

54. setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
55. setBackground(new java.awt.Color(204, 204, 204));
56. getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

57. jLabel3.setBackground(new java.awt.Color(204, 204, 255));
58. jLabel3.setFont(new java.awt.Font("Berlin Sans FB Demi", 1, 36)); // NOI18N
59. jLabel3.setForeground(new java.awt.Color(255, 255, 255));
60. jLabel3.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
61. jLabel3.setText("SISTEM PELAYANAN JASA LAUNDRY ");
62. getContentPane().add(jLabel3, new org.netbeans.lib.awtextra.AbsoluteConstraints(10, 90, 600, 60));

63. jLabel2.setFont(new java.awt.Font("Times New Roman", 2, 14)); // NOI18N
64. jLabel2.setForeground(new java.awt.Color(255, 255, 255));
65. jLabel2.setText("Selamat Datang, ");
66. getContentPane().add(jLabel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(10, 20, 114, -1));

67. jLabel4.setIcon(new javax.swing.ImageIcon(getClass().getResource("/img/bg1.png"))); // NOI18N
68. getContentPane().add(jLabel4, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, 730, 290));

69. jMenu7.setText("Master");

70. jMenuItem1.setText("Data Karyawan");
71. jMenuItem1.addActionListener(new java.awt.event.ActionListener() {
72.     public void actionPerformed(java.awt.event.ActionEvent evt) {
73.         jMenuItem1ActionPerformed(evt);
74.     }
75. });
75. jMenu7.add(jMenuItem1);

76. jMenuItem2.setText("Data Pelanggan");
77. jMenuItem2.addActionListener(new java.awt.event.ActionListener() {
78.     public void actionPerformed(java.awt.event.ActionEvent evt) {
79.         jMenuItem2ActionPerformed(evt);
80.     }
80. });
81. jMenu7.add(jMenuItem2);

82. jMenuItem5.setText("Data Jenis Barang");
83. jMenuItem5.addActionListener(new java.awt.event.ActionListener() {
84.     public void actionPerformed(java.awt.event.ActionEvent evt) {
```

```
        jMenuItem5ActionPerformed(evt);
85.    }
86. });
87. jMenuItem7.add(jMenuItem5);

88. jMenuBar2.add(jMenu7);

89. jMenu8.setText("Transaksi");

90. jMenuItem4.setText("Data Transaksi");
91. jMenuItem4.addActionListener(new java.awt.event.ActionListener() {
92.     public void actionPerformed(java.awt.event.ActionEvent evt) {
93.         jMenuItem4ActionPerformed(evt);
93.     }
94. });
95. jMenuItem8.add(jMenuItem4);

96. jMenuItem6.setText("Barang");
97. jMenuItem6.addActionListener(new java.awt.event.ActionListener() {
98.     public void actionPerformed(java.awt.event.ActionEvent evt) {
99.         jMenuItem6ActionPerformed(evt);
99.     }
100. });
101. jMenuItem8.add(jMenuItem6);

102. jMenuBar2.add(jMenu8);

103. jMenu1.setText("Laporan");

104. jMenuItem9.setText("Laporan Karyawan");
105. jMenuItem9.addActionListener(new java.awt.event.ActionListener() {
106.     public void actionPerformed(java.awt.event.ActionEvent evt) {
107.         jMenuItem9ActionPerformed(evt);
107.     }
108. });
109. jMenu1.add(jMenuItem9);

110. jMenuItem3.setText("Laporan Pelanggan");
111. jMenuItem3.addActionListener(new java.awt.event.ActionListener() {
112.     public void actionPerformed(java.awt.event.ActionEvent evt) {
113.         jMenuItem3ActionPerformed(evt);
113.     }
114. });
115. jMenu1.add(jMenuItem3);

116. jMenuItem3.setText("Laporan Transaksi");

117. jMenuItem10.setText("Barang Sudah Diambil");
118. jMenuItem10.addActionListener(new java.awt.event.ActionListener() {
119.     public void actionPerformed(java.awt.event.ActionEvent evt) {
120.         jMenuItem10ActionPerformed(evt);
120.     }
121. });
122. jMenuItem3.add(jMenuItem10);
```

```
123. jMenuItem11.setText("Barang Belum Diambil");
124. jMenuItem11.addActionListener(new java.awt.event.ActionListener() {
125.     public void actionPerformed(java.awt.event.ActionEvent evt) {
126.         jMenuItem11ActionPerformed(evt);
127.     });
128. jMenuItem3.add(jMenuItem11);

129. jMenuItem12.setText("Rincian Transaksi");
130. jMenuItem12.addActionListener(new java.awt.event.ActionListener() {
131.     public void actionPerformed(java.awt.event.ActionEvent evt) {
132.         jMenuItem12ActionPerformed(evt);
133.     });
134. jMenuItem3.add(jMenuItem12);

135. jMenu1.add(jMenu3);

136. jMenuBar2.add(jMenu1);

137. jMenu2.setText("Menu");

138. jMenuItem7.setText("LogOut");
139. jMenuItem7.addActionListener(new java.awt.event.ActionListener() {
140.     public void actionPerformed(java.awt.event.ActionEvent evt) {
141.         jMenuItem7ActionPerformed(evt);
142.     });
143. jMenu2.add(jMenuItem7);

144. jMenuItem8.setText("Exit");
145. jMenuItem8.addActionListener(new java.awt.event.ActionListener() {
146.     public void actionPerformed(java.awt.event.ActionEvent evt) {
147.         jMenuItem8ActionPerformed(evt);
148.     });
149. jMenu2.add(jMenuItem8);

150. jMenuBar2.add(jMenu2);

151. setJMenuBar(jMenuBar2);

152. setSize(new java.awt.Dimension(635, 357));
153. setLocationRelativeTo(null);
154. } // </editor-fold>

155. private void jMenuItem1ActionPerformed(java.awt.event.ActionEvent evt) {
156.     karyawan frmK = new karyawan();
157.     frmK.setVisible(true);
158.     this.dispose();
159. }

160. private void jMenuItem2ActionPerformed(java.awt.event.ActionEvent evt) {
161.     Pelanggan frmP = new Pelanggan();
162.     frmP.setVisible(true);
```

```

163. this.dispose();
164. }

165. private void jMenuItem5ActionPerformed(java.awt.event.ActionEvent evt) {
166. JenisBarang frmJB = new JenisBarang();
167. frmJB.setVisible(true);
168. this.dispose();
169. }

170. private void jMenuItem4ActionPerformed(java.awt.event.ActionEvent evt) {
171. Transaksi frmT = new Transaksi();
172. frmT.setVisible(true);
173. this.dispose();
174. }

175. private void jMenuItem3ActionPerformed(java.awt.event.ActionEvent evt) {
176. try{
177. File file = new File ("src/Laporan/LapPelanggan.jrxml");
178. Class.forName("com.mysql.jdbc.Driver");
179. HashMap parameter = new HashMap();
180. JasperDesign jasperDesign = JRXMLLoader.load(file);
181. JasperReport jasperReport = JasperCompileManager.compileReport(jasperDesign);
182. JasperPrint jasperPrint = JasperFillManager.fillReport(jasperReport, parameter, conn);
183. JasperViewer.viewReport(jasperPrint,false);
184. }catch (ClassNotFoundException | JRException e){
185. JOptionPane.showMessageDialog(null,"Data Tidak Ada!"+e.getMessage(),"Cetak
Data",JOptionPane.ERROR_MESSAGE);
186. }
187. }

188. private void jMenuItem7ActionPerformed(java.awt.event.ActionEvent evt) {
189. int pesan=JOptionPane.showConfirmDialog(null, "Yakin ingin keluar
?";
"Konfirmasi",JOptionPane.YES_NO_OPTION,JOptionPane.QUESTION_MESSAGE
);
190. if(pesan==JOptionPane.YES_OPTION){
191. this.setVisible(false);
192. new login().setVisible(true);
193. }
194. }

195. private void jMenuItem8ActionPerformed(java.awt.event.ActionEvent evt) {
196. dispose();
197. }

198. private void jMenuItem6ActionPerformed(java.awt.event.ActionEvent evt) {
199. Barang frmT = new Barang();
200. frmT.setVisible(true);
201. this.dispose();
202. }

203. private void jMenuItem9ActionPerformed(java.awt.event.ActionEvent evt) {
204. try{
File file = new File ("C:/Users/Rama/Documents/NetBeansProjects/Pelayanan Jasa
Laundry/src/Laporan/LapKaryawan.jrxml");

```

```
Class.forName("com.mysql.jdbc.Driver");
HashMap parameter = new HashMap();
JasperDesign jasperDesign = JRXmlLoader.load(file);
JasperReport jasperReport = JasperCompileManager.compileReport(jasperDesign);
JasperPrint jasperPrint = JasperFillManager.fillReport(jasperReport, parameter, conn);
JasperViewer.viewReport(jasperPrint,false);
}catch (ClassNotFoundException | JRException e){
JOptionPane.showMessageDialog(null,"Data Tidak Ada!"+e.getMessage(),"Cetak
Data",JOptionPane.ERROR_MESSAGE);
}
205. }

206. private void jMenuItem10ActionPerformed(java.awt.event.ActionEvent evt) {
207. try{
208. File file = new File ("src/Laporan/LapTransaksiSdh.jrxml");
209. Class.forName("com.mysql.jdbc.Driver");
210. HashMap parameter = new HashMap();
211. JasperDesign jasperDesign = JRXmlLoader.load(file);
212. JasperReport jasperReport = JasperCompileManager.compileReport(jasperDesign);
213. JasperPrint jasperPrint = JasperFillManager.fillReport(jasperReport, parameter, conn);
214. JasperViewer.viewReport(jasperPrint,false);
215. }catch (ClassNotFoundException | JRException e){
216. JOptionPane.showMessageDialog(null,"Data Tidak Ada!"+e.getMessage(),"Cetak
Data",JOptionPane.ERROR_MESSAGE);
217. }
218. }

219. private void jMenuItem11ActionPerformed(java.awt.event.ActionEvent evt) {
220. try{
221. File file = new File ("src/Laporan/LapTransaksiBlm.jrxml");
222. Class.forName("com.mysql.jdbc.Driver");
223. HashMap parameter = new HashMap();
224. JasperDesign jasperDesign = JRXmlLoader.load(file);
225. JasperReport jasperReport = JasperCompileManager.compileReport(jasperDesign);
226. JasperPrint jasperPrint = JasperFillManager.fillReport(jasperReport, parameter, conn);
227. JasperViewer.viewReport(jasperPrint,false);
228. }catch (ClassNotFoundException | JRException e){
229. JOptionPane.showMessageDialog(null,"Data Tidak Ada!"+e.getMessage(),"Cetak
Data",JOptionPane.ERROR_MESSAGE);
230. }
231. }

232. private void jMenuItem12ActionPerformed(java.awt.event.ActionEvent evt) {
233. try{
234. File file = new File ("src/Laporan/LapTransaksiRinci.jrxml");
235. Class.forName("com.mysql.jdbc.Driver");
236. HashMap parameter = new HashMap();
237. JasperDesign jasperDesign = JRXmlLoader.load(file);
238. JasperReport jasperReport = JasperCompileManager.compileReport(jasperDesign);
239. JasperPrint jasperPrint = JasperFillManager.fillReport(jasperReport, parameter, conn);
240. JasperViewer.viewReport(jasperPrint,false);
241. }catch (ClassNotFoundException | JRException e){
242. JOptionPane.showMessageDialog(null,"Data Tidak Ada!"+e.getMessage(),"Cetak
Data",JOptionPane.ERROR_MESSAGE);
243. }
```

```

244. }

245. /**
   @param args the command line arguments
246. */
247. public static void main(String args[]) {
248. /* Set the Nimbus look and feel */
249. //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
250. /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and
   feel.
   For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
251. */
252. try {
253. for (javax.swing.UIManager.LookAndFeelInfo info :
   javax.swing.UIManager.getInstalledLookAndFeels()) {
   if ("Nimbus".equals(info.getName())) {
   javax.swing.UIManager.setLookAndFeel(info.getClassName());
   break;
   }
254. }
255. } catch (ClassNotFoundException | InstantiationException | IllegalAccessException |
   javax.swing.UnsupportedLookAndFeelException ex) {
256. java.util.logging.Logger.getLogger(menu.class.getName()).log(java.util.logging.Level.SEVERE,
   null, ex);
257. }
258. //</editor-fold>

259. //</editor-fold>

260. /* Create and display the form */
261. java.awt.EventQueue.invokeLater(() -> {
262. new menu().setVisible(true);
263. });
264. }

265. // Variables declaration - do not modify
266. private javax.swing.JLabel jLabel2;
267. private javax.swing.JLabel jLabel3;
268. private javax.swing.JLabel jLabel4;
269. private javax.swing.JMenu jMenu1;
270. private javax.swing.JMenu jMenu2;
271. private javax.swing.JMenu jMenu3;
272. private javax.swing.JMenu jMenu7;
273. private javax.swing.JMenu jMenu8;
274. private javax.swing.JMenuBar jMenuBar2;
275. private javax.swing.JMenuItem jMenuItem1;
276. private javax.swing.JMenuItem jMenuItem10;
277. private javax.swing.JMenuItem jMenuItem11;
278. private javax.swing.JMenuItem jMenuItem12;
279. private javax.swing.JMenuItem jMenuItem2;
280. private javax.swing.JMenuItem jMenuItem3;
281. private javax.swing.JMenuItem jMenuItem4;
282. private javax.swing.JMenuItem jMenuItem5;
283. private javax.swing.JMenuItem jMenuItem6;
284. private javax.swing.JMenuItem jMenuItem7;

```

```
285. private javax.swing.JMenuItem jMenuItem8;
286. private javax.swing.JMenuItem jMenuItem9;
287. // End of variables declaration
288. }
```

**D. Karyawan.java**

```
1. package Form;
2. import java.sql.*;
3. import javax.swing.JOptionPane;
4. import javax.swing.table.DefaultTableModel;
5. import java.awt.event.KeyEvent;
6. import Koneksi.koneksi;
7. import java.awt.Color;
8. /**
9. *
10. * @author USER
11. */
12. public final class karyawan extends javax.swing.JFrame {
13. private final Connection conn = new koneksi().connect();
14. private DefaultTableModel tabmode;
15. /**
16. Creates new form customer
17. */
18. public karyawan() {
19. initComponents();
20. kosong();
21. aktif();
22. datatable();
23. }
24. protected void aktif(){
25. txtid.requestFocus();
26. }

27. protected void kosong(){
28. txtid.setText("");
29. txtnm.setText("");
30. txtnotlp.setText("");
31. txtcari.setText("");
32. txtpass.setText("");
33. }

34. protected void datatable(){
35. Object[] Baris = {"ID Karyawan","Nama Karyawan","No.Telepon","Password"};
36. tabmode = new DefaultTableModel(null,Baris);
37. String cariitem=txtcari.getText();

38. try {
39. String sql = "SELECT * FROM karyawan where idkrywn like '%" +cariitem+"%' or
40. nmkrywn like '%" +cariitem+"%' order by idkrywn asc";
41. Statement stat = conn.createStatement();
42. ResultSet hasil = stat.executeQuery(sql);
42. while (hasil.next()){
43. tabmode.addRow(new Object[]{
```

```

        hasil.getString(1),
        hasil.getString(2),
        hasil.getString(3),
        hasil.getString(4)

    43. });
    44. }
    45. tbkrywn.setModel(tabmode);
    46. } catch (SQLException e){
    a. JOptionPane.showMessageDialog(null, "data gagal dipanggil"+e);
    47. }
    48. }

    49. /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
    50. */
    51. @SuppressWarnings("unchecked")
    52. // <editor-fold defaultstate="collapsed" desc="Generated Code">
    53. private void initComponents() {

    54. jPanel2 = new javax.swing.JPanel();
    55. bcari = new javax.swing.JButton();
    56. txtcari = new javax.swing.JTextField();
    57. jPanel1 = new javax.swing.JPanel();
    58. jLabel2 = new javax.swing.JLabel();
    59. jLabel3 = new javax.swing.JLabel();
    60. jLabel5 = new javax.swing.JLabel();
    61. jLabel4 = new javax.swing.JLabel();
    62. txtid = new javax.swing.JTextField();
    63. txtnm = new javax.swing.JTextField();
    64. txtnotlp = new javax.swing.JTextField();
    65. txtpass = new javax.swing.JTextField();
    66. jLabel1 = new javax.swing.JLabel();
    67. bsimpan = new javax.swing.JButton();
    68. bbatal = new javax.swing.JButton();
    69. bubah = new javax.swing.JButton();
    70. bhapus = new javax.swing.JButton();
    71. bkeluar = new javax.swing.JButton();
    72. jScrollPane1 = new javax.swing.JScrollPane();
    73. tbkrywn = new javax.swing.JTable();
    74. jLabel7 = new javax.swing.JLabel();

    75. setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
    76. getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

    77. jPanel2.setBorder(javax.swing.BorderFactory.createTitledBorder(javax.swing.BorderFactory.
        createEmptyBorder(1, 1, 1, 1), "Cari Karyawan",
        javax.swing.border.TitledBorder.DEFAULT_JUSTIFICATION,
        javax.swing.border.TitledBorder.DEFAULT_POSITION, new java.awt.Font("Times
        New Roman", 0, 13), new java.awt.Color(255, 255, 255))); // NOI18N
    78. jPanel2.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

    79. bcari.setFont(new java.awt.Font("Baskerville Old Face", 0, 14)); // NOI18N

```

```
80. bcari.setText("Cari");
81. bcari.addActionListener(new java.awt.event.ActionListener() {
82.     public void actionPerformed(java.awt.event.ActionEvent evt) {
83.         bcariActionPerformed(evt);
84.     }
85. });
86. jPanel2.add(bcari, new org.netbeans.lib.awtextra.AbsoluteConstraints(150, 20, 70, -1));

86. txtcari.addKeyListener(new java.awt.event.KeyAdapter() {
87.     public void keyPressed(java.awt.event.KeyEvent evt) {
88.         txtcariKeyPressed(evt);
89.     }
90. });
90. jPanel2.add(txtcari, new org.netbeans.lib.awtextra.AbsoluteConstraints(10, 20, 130, -1));

91. getContentPane().add(jPanel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(20,
92. 250, 240, 50));

92. jPanel1.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

93. jLabel2.setFont(new java.awt.Font("Baskerville Old Face", 0, 14)); // NOI18N
94. jLabel2.setForeground(new java.awt.Color(255, 255, 255));
95. jLabel2.setText("Id Karyawan");
96. jPanel1.add(jLabel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(10, 20, -1, 20));

97. jLabel3.setFont(new java.awt.Font("Baskerville Old Face", 0, 14)); // NOI18N
98. jLabel3.setForeground(new java.awt.Color(255, 255, 255));
99. jLabel3.setText("Nama Karyawan");
100. jPanel1.add(jLabel3, new org.netbeans.lib.awtextra.AbsoluteConstraints(10, 60, 100, -1));

101. jLabel5.setFont(new java.awt.Font("Baskerville Old Face", 0, 14)); // NOI18N
102. jLabel5.setForeground(new java.awt.Color(255, 255, 255));
103. jLabel5.setText("No Telepon");
104. jPanel1.add(jLabel5, new org.netbeans.lib.awtextra.AbsoluteConstraints(10, 100, -1, -1));

105. jLabel4.setFont(new java.awt.Font("Baskerville Old Face", 0, 14)); // NOI18N
106. jLabel4.setForeground(new java.awt.Color(255, 255, 255));
107. jLabel4.setText("Password");
108. jPanel1.add(jLabel4, new org.netbeans.lib.awtextra.AbsoluteConstraints(10, 140, 93,
109. 20));
109. jPanel1.add(txtid, new org.netbeans.lib.awtextra.AbsoluteConstraints(130, 10, 170, 30));
110. jPanel1.add(txtnm, new org.netbeans.lib.awtextra.AbsoluteConstraints(130, 50, 170, 30));
111. jPanel1.add(txtnotlp, new org.netbeans.lib.awtextra.AbsoluteConstraints(130, 90, 170,
30));
112. jPanel1.add(txtpass, new org.netbeans.lib.awtextra.AbsoluteConstraints(130, 130, 170,
30));

113. getContentPane().add(jPanel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(20, 70,
310, 170));

114. jLabel1.setFont(new java.awt.Font("Sitka Banner", 1, 36)); // NOI18N
115. jLabel1.setText("Data Karyawan");
116. getContentPane().add(jLabel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(220, 0,
-1, -1));
```

```
117. bsimpan.setFont(new java.awt.Font("Baskerville Old Face", 0, 14)); // NOI18N
118. bsimpan.setText("simpan");
119. bsimpan.addActionListener(new java.awt.event.ActionListener() {
120.     public void actionPerformed(java.awt.event.ActionEvent evt) {
121.         bsimpanActionPerformed(evt);
122.     }
123.     getContentPane().add(bsimpan, new org.netbeans.lib.awtextra.AbsoluteConstraints(370,
124.         70, 80, -1));

125. bbatal.setFont(new java.awt.Font("Baskerville Old Face", 0, 14)); // NOI18N
126. bbatal.setText("batal");
127. bbatal.addActionListener(new java.awt.event.ActionListener() {
128.     public void actionPerformed(java.awt.event.ActionEvent evt) {
129.         bbatalActionPerformed(evt);
130.     }
131.     getContentPane().add(bbatal, new org.netbeans.lib.awtextra.AbsoluteConstraints(370,
132.         190, 80, -1));

133. bubah.setFont(new java.awt.Font("Baskerville Old Face", 0, 14)); // NOI18N
134. bubah.setText("ubah");
135. bubah.addActionListener(new java.awt.event.ActionListener() {
136.     public void actionPerformed(java.awt.event.ActionEvent evt) {
137.         bubahActionPerformed(evt);
138.     }
139.     getContentPane().add(bubah, new org.netbeans.lib.awtextra.AbsoluteConstraints(370,
140.         110, 80, -1));

141. bhapus.setFont(new java.awt.Font("Baskerville Old Face", 0, 14)); // NOI18N
142. bhapus.setText("hapus");
143. bhapus.addActionListener(new java.awt.event.ActionListener() {
144.     public void actionPerformed(java.awt.event.ActionEvent evt) {
145.         bhapusActionPerformed(evt);
146.     }
147.     getContentPane().add(bhapus, new org.netbeans.lib.awtextra.AbsoluteConstraints(370,
148.         150, 80, -1));

149. bkeluar.setFont(new java.awt.Font("Baskerville Old Face", 0, 14)); // NOI18N
150. bkeluar.setText("Kembali");
151. bkeluar.addActionListener(new java.awt.event.ActionListener() {
152.     public void actionPerformed(java.awt.event.ActionEvent evt) {
153.         bkeluarActionPerformed(evt);
154.     }
155.     getContentPane().add(bkeluar, new org.netbeans.lib.awtextra.AbsoluteConstraints(560,
156.         70, 80, -1));

157. tbkrywn.setModel(new javax.swing.table.DefaultTableModel(
158.     new Object [][] {
159.         {null, null, null, null},
160.         {null, null, null, null},
161.         {null, null, null, null},
162.         {null, null, null, null}
163.     },
164.     new String [] {
165.         "No", "Jenis", "Merk", "Harga"
166.     }
167. ));
```

```

    {null, null, null, null}
154. },
155. new String [] {
    "Title 1", "Title 2", "Title 3", "Title 4"
156. }
157. )));
158. tbkrywn.addMouseListener(new java.awt.event.MouseAdapter() {
159. public void mouseClicked(java.awt.event.MouseEvent evt) {
    tbkrywnMouseClicked(evt);
160. }
161. });
162. jScrollPane1.setViewportView(tbkrywn);

163. getContentPane().add(jScrollPane1, new
    org.netbeans.lib.awtextra.AbsoluteConstraints(20, 310, 620, 180));

164. jLabel7.setIcon(new javax.swing.ImageIcon(getClass().getResource("/img/bg3.jpg"))); // NOI18N
165. getContentPane().add(jLabel7, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, 670, 520));

166. setSize(new java.awt.Dimension(685, 565));
167. setLocationRelativeTo(null);
168. } // </editor-fold>

169. private void bhapusActionPerformed(java.awt.event.ActionEvent evt) {
170. int ok = JOptionPane.showConfirmDialog(null,"hapus","konfirmasi
    dialog",JOptionPane.YES_NO_OPTION);
171. if (ok==0){
    String sql = "delete from karyawan where idkrywn = '"+txtid.getText()+"'";
    try{
        PreparedStatement stat = conn.prepareStatement(sql);
        stat.executeUpdate();
        JOptionPane.showMessageDialog(null, "data berhasil dihapus");
        kosong();
        txtid.requestFocus();
    }
    catch (SQLException e){
        JOptionPane.showMessageDialog(null, "data gagal dihapus"+e);
    }
    datatable();
172. }
173. }

174. private void bbatalActionPerformed(java.awt.event.ActionEvent evt) {
175. kosong();
176. datatable();
177. }

178. private void bubahActionPerformed(java.awt.event.ActionEvent evt) {
179. try{
180. String sql = "update karyawan set idkrywn=?,nmkrywn=?,notlp=?,password=? where
    idkrywn='"+txtid.getText()+"'";
    PreparedStatement stat = conn.prepareStatement(sql);
    stat.setString(1, txtid.getText());

```

```

stat.setString(2, txtnm.getText());
stat.setString(3, txtnotlp.getText());
stat.setString(4, txtpass.getText());

stat.executeUpdate();
JOptionPane.showMessageDialog(null, "Data Berhasil Diubah");
kosong();
txtid.requestFocus();
181. }
182. catch (SQLException e){
183. JOptionPane.showMessageDialog(null, "Data Gagal Diubah"+e);
184. }
185. datatable();
186. }

187. private void bsimpanActionPerformed(java.awt.event.ActionEvent evt) {
188. String sql = "insert into karyawan values (?,?,?,?,?)";
189. try{
190. PreparedStatement stat = conn.prepareStatement(sql);
    stat.setString(1, txtid.getText());
    stat.setString(2, txtnm.getText());
    stat.setString(3, txtnotlp.getText());
    stat.setString(4, txtpass.getText());

191. stat.executeUpdate();
192. JOptionPane.showMessageDialog(null, "Data Berhasil Disimpan");
193. kosong();
194. txtid.requestFocus();
195. }
196. catch (SQLException e){
197. JOptionPane.showMessageDialog(null, "Data Gagal Disimpan");
198. }
199. datatable();
200. }

201. private void txtcariKeyPressed(java.awt.event.KeyEvent evt) {
202. if (evt.getKeyCode() == KeyEvent.VK_ENTER){
203. datatable();
204. }
205. }

206. private void bcariActionPerformed(java.awt.event.ActionEvent evt) {
207. datatable();
208. }

209. private void tbkrywnMouseClicked(java.awt.event.MouseEvent evt) {
210. int bar = tbkrywn.getSelectedRow();
211. String a = tabmode.getValueAt(bar, 0).toString();
212. String b = tabmode.getValueAt(bar, 1).toString();
213. String c = tabmode.getValueAt(bar, 2).toString();
214. String d = tabmode.getValueAt(bar, 3).toString();

215. txtid.setText(a);
216. txtnm.setText(b);
217. txtnotlp.setText(c);

```

```
218. txtpass.setText(d);
219. }

220. private void bkeluarActionPerformed(java.awt.event.ActionEvent evt) {
221.     this.setVisible(false);
222.     new menu().setVisible(true);
223. }

224. /**
225.  * @param args the command line arguments
226. */
227. public static void main(String args[]) {
228.     /* Set the Nimbus look and feel */
229.     //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
230.     /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and
231.      feel.
232.      For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
233.     */
234.     try {
235.         for (javax.swing.UIManager.LookAndFeelInfo info :
236.             javax.swing.UIManager.getInstalledLookAndFeels()) {
237.             if ("Nimbus".equals(info.getName())) {
238.                 javax.swing.UIManager.setLookAndFeel(info.getClassName());
239.                 break;
240.             }
241.         }
242.     } catch (ClassNotFoundException | InstantiationException | IllegalAccessException |
243.             javax.swing.UnsupportedLookAndFeelException ex) {
244.         java.util.logging.Logger.getLogger(karyawan.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
245.     }
246.     //
247.     //

248.     // Create and display the form */
249.     java.awt.EventQueue.invokeLater(() -> {
250.         new karyawan().setVisible(true);
251.     });
252. }

253. // Variables declaration - do not modify
254. private javax.swing.JButton bbatal;
255. private javax.swing.JButton bcaro;
256. private javax.swing.JButton bhapus;
257. private javax.swing.JButton bkeluar;
258. private javax.swing.JButton bsimpan;
259. private javax.swing.JButton bubah;
260. private javax.swing.JLabel jLabel1;
261. private javax.swing.JLabel jLabel2;
262. private javax.swing.JLabel jLabel3;
263. private javax.swing.JLabel jLabel4;
264. private javax.swing.JLabel jLabel5;
```

```
258. private javax.swing.JLabel jLabel7;
259. private javax.swing.JPanel jPanel1;
260. private javax.swing.JPanel jPanel2;
261. private javax.swing.JScrollPane jScrollPane1;
262. private javax.swing.JTable tbkrywn;
263. private javax.swing.JTextField txtcari;
264. private javax.swing.JTextField txtid;
265. private javax.swing.JTextField txtnm;
266. private javax.swing.JTextField txtnotlp;
267. private javax.swing.JTextField txtpass;
268. // End of variables declaration
269. }
```

#### E. Pelanggan.java

```
1. package Form;
2. import java.sql.*;
3. import javax.swing.JOptionPane;
4. import javax.swing.table.DefaultTableModel;
5. import java.awt.event.KeyEvent;
6. import Koneksi.koneksi;
7. import java.io.File;
8. import java.util.HashMap;
9. import net.sf.jasperreports.engine.JRException;
10. import net.sf.jasperreports.engine.JasperFillManager;
11. import net.sf.jasperreports.engine.JasperPrint;
12. import net.sf.jasperreports.engine.JasperReport;
13. import net.sf.jasperreports.engine.util.JRLoader;
14. import net.sf.jasperreports.view.JasperViewer;

15. /**
16. *
17. * @author USER
18. */
19. public final class Pelanggan extends javax.swing.JFrame {
20. private final Connection conn = new koneksi().connect();
21. private DefaultTableModel tabmode;
21. /**
22. * Creates new form Pelanggan
22. */
23. public Pelanggan() {
24. initComponents();
25. kosong();
26. aktif();
27. datatable();

28. }
29. protected void aktif(){
30. txtidplgn.requestFocus();
31. }

32. protected void kosong(){
33. txtidplgn.setText("");
34. txtnmplgn.setText("");
35. txtnotlp.setText("");
```

```

36. txtalamat.setText("");
37. txtcari.setText("");
38. }

39. protected void datatable (){
40. Object[] Baris = {"Id Pelanggan", "Nama Pelanggan", "No Telepon", "Alamat"};
41. tabmode = new DefaultTableModel(null,Baris);
42. String cariitem=txtcari.getText();

43. try {
44. String sql = "SELECT * FROM pelanggan where idplgn like '%" +cariitem+"%' or nmplgn
like '%" +cariitem+"%' order by idplgn asc";
45. Statement stat = conn.createStatement();
46. ResultSet hasil = stat.executeQuery(sql);
47. while (hasil.next()){
tabmode.addRow(new Object[]{
hasil.getString(1),
hasil.getString(2),
hasil.getString(3),
hasil.getString(4)
});
49. }
50. tbplgn.setModel(tabmode);
51. } catch (SQLException e){
JOptionPane.showMessageDialog(null, "data gagal dipanggil"+e);
52. }
53. }
54. /**
This method is called from within the constructor to initialize the form.
WARNING: Do NOT modify this code. The content of this method is always
regenerated by the Form Editor.
55. */
56. @SuppressWarnings("unchecked")
57. // <editor-fold defaultstate="collapsed" desc="Generated Code">
58. private void initComponents() {

59. jLabel6 = new javax.swing.JLabel();
60. txtcari = new javax.swing.JTextField();
61. bcari = new javax.swing.JButton();
62. jScrollPane2 = new javax.swing.JScrollPane();
63. tbplgn = new javax.swing.JTable();
64. bsimpan = new javax.swing.JButton();
65. bubah = new javax.swing.JButton();
66. bcetak = new javax.swing.JButton();
67. bkeluar = new javax.swing.JButton();
68. bhapus = new javax.swing.JButton();
69. bbatal = new javax.swing.JButton();
70. jScrollPane1 = new javax.swing.JScrollPane();
71. txtalamat = new javax.swing.JTextArea();
72. jLabel5 = new javax.swing.JLabel();
73. txtnotlp = new javax.swing.JTextField();
74. jLabel4 = new javax.swing.JLabel();
75. txtmplgn = new javax.swing.JTextField();
76. jLabel3 = new javax.swing.JLabel();
77. txtidplgn = new javax.swing.JTextField();

```

```
78. jLabel2 = new javax.swing.JLabel();
79. jLabel1 = new javax.swing.JLabel();
80. jLabel7 = new javax.swing.JLabel();

81. setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
82. getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

83. jLabel6.setFont(new java.awt.Font("Times New Roman", 1, 18)); // NOI18N
84. jLabel6.setText("Cari Data Pelanggan");
85. getContentPane().add(jLabel6, new org.netbeans.lib.awtextra.AbsoluteConstraints(10, 240, -1, -1));
86. getContentPane().add(txtcari, new org.netbeans.lib.awtextra.AbsoluteConstraints(20, 280, 70, -1));

87. bcari.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N
88. bcari.setText("Cari");
89. bcari.addActionListener(new java.awt.event.ActionListener() {
90.     public void actionPerformed(java.awt.event.ActionEvent evt) {
a.         bcariActionPerformed(evt);
91.     }
92. });
93. bcari.addKeyListener(new java.awt.event.KeyAdapter() {
94.     public void keyPressed(java.awt.event.KeyEvent evt) {
a.         bcariKeyPressed(evt);
95.     }
96. });
97. getContentPane().add(bcari, new org.netbeans.lib.awtextra.AbsoluteConstraints(100, 280, 70, -1));

98. tbplgn.setModel(new javax.swing.table.DefaultTableModel(
99.     new Object [][] {
        {null, null, null, null},
        {null, null, null, null},
        {null, null, null, null},
        {null, null, null, null}
    },
100.     new String [] {
        "Title 1", "Title 2", "Title 3", "Title 4"
    }
101. );
102. );
103. );
104. tbplgn.addMouseListener(new java.awt.event.MouseAdapter() {
105.     public void mouseClicked(java.awt.event.MouseEvent evt) {
            tbplgnMouseClicked(evt);
106.     }
107. });
108. jScrollPane2.setViewportView(tbplgn);

109. getContentPane().add(jScrollPane2, new
        org.netbeans.lib.awtextra.AbsoluteConstraints(10, 320, 740, 160));

110. bsimpan.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N
111. bsimpan.setText("Simpan");
112. bsimpan.addActionListener(new java.awt.event.ActionListener() {
113.     public void actionPerformed(java.awt.event.ActionEvent evt) {
            bsimpanActionPerformed(evt);

```

```
114. }
115. });
116. getContentPane().add(bsimpan, new org.netbeans.lib.awtextra.AbsoluteConstraints(330,
40, 80, -1));

117. bubah.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N
118. bubah.setText("Ubah");
119. bubah.addActionListener(new java.awt.event.ActionListener() {
120.     public void actionPerformed(java.awt.event.ActionEvent evt) {
121.         bubahActionPerformed(evt);
122.     });
123. getContentPane().add(bubah, new org.netbeans.lib.awtextra.AbsoluteConstraints(330,
120, 80, -1));

124. bcetak.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N
125. bcetak.setText("Cetak");
126. bcetak.addActionListener(new java.awt.event.ActionListener() {
127.     public void actionPerformed(java.awt.event.ActionEvent evt) {
128.         bcetakActionPerformed(evt);
129.     });
130. getContentPane().add(bcetak, new org.netbeans.lib.awtextra.AbsoluteConstraints(330,
240, 80, -1));

131. bkeluar.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N
132. bkeluar.setText("Keluar");
133. bkeluar.addActionListener(new java.awt.event.ActionListener() {
134.     public void actionPerformed(java.awt.event.ActionEvent evt) {
135.         bkeluarActionPerformed(evt);
136.     });
137. getContentPane().add(bkeluar, new org.netbeans.lib.awtextra.AbsoluteConstraints(630,
70, 80, -1));

138. bhapus.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N
139. bhapus.setText("Hapus");
140. bhapus.addActionListener(new java.awt.event.ActionListener() {
141.     public void actionPerformed(java.awt.event.ActionEvent evt) {
142.         bhapusActionPerformed(evt);
143.     });
144. getContentPane().add(bhapus, new org.netbeans.lib.awtextra.AbsoluteConstraints(330,
80, 80, -1));

145. bbatal.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N
146. bbatal.setText("Batal");
147. bbatal.addActionListener(new java.awt.event.ActionListener() {
148.     public void actionPerformed(java.awt.event.ActionEvent evt) {
149.         bbatalActionPerformed(evt);
150.     });
151. getContentPane().add(bbatal, new org.netbeans.lib.awtextra.AbsoluteConstraints(330,
160, 80, -1));
```

```

152. txtalamat.setColumns(20);
153. txtalamat.setRows(5);
154. jScrollPane1.setViewportView(txtalamat);

155. getContentPane().add(jScrollPane1, new
org.netbeans.lib.awtextra.AbsoluteConstraints(120, 150, 180, 60));

156. jLabel5.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N
157. jLabel5.setText("Alamat");
158. getContentPane().add(jLabel5, new org.netbeans.lib.awtextra.AbsoluteConstraints(10,
180, -1, -1));
159. getContentPane().add(txtnotlp, new org.netbeans.lib.awtextra.AbsoluteConstraints(120,
110, 180, 30));

160. jLabel4.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N
161. jLabel4.setText("No Telepon");
162. getContentPane().add(jLabel4, new org.netbeans.lib.awtextra.AbsoluteConstraints(10,
120, -1, -1));
163. getContentPane().add(txtnmplgn, new org.netbeans.lib.awtextra.AbsoluteConstraints(120,
70, 180, 30));

164. jLabel3.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N
165. jLabel3.setText("Nama Pelanggan");
166. getContentPane().add(jLabel3, new org.netbeans.lib.awtextra.AbsoluteConstraints(10, 80,
-1, -1));
167. getContentPane().add(txtidplgn, new org.netbeans.lib.awtextra.AbsoluteConstraints(120,
30, 180, 30));

168. jLabel2.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N
169. jLabel2.setText("Id Pelanggan");
170. getContentPane().add(jLabel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(10, 40,
-1, -1));

171. jLabel1.setFont(new java.awt.Font("Baskerville Old Face", 1, 36)); // NOI18N
172. jLabel1.setText("PELANGGAN");
173. getContentPane().add(jLabel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(490,
20, -1, 40));

174. jLabel7.setIcon(new javax.swing.ImageIcon(getClass().getResource("/img/bg2.jpg"))); // NOI18N
175. getContentPane().add(jLabel7, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0,
760, 490));

176. setSize(new java.awt.Dimension(776, 536));
177. setLocationRelativeTo(null);
178. } // </editor-fold>

179. private void bbatalActionPerformed(java.awt.event.ActionEvent evt) {
180. kosong();
181. datatable();
182. }

183. private void bsimpanActionPerformed(java.awt.event.ActionEvent evt) {
184. String sql = "insert into pelanggan values (?,?,?,?,?)";
185. try{

```

```

186. PreparedStatement stat = conn.prepareStatement(sql);
187. stat.setString(1, txtidplgn.getText());
188. stat.setString(2, txtnmplgn.getText());
189. stat.setString(3, txtnotlp.getText());
190. stat.setString(4, txtalamat.getText());

191. stat.executeUpdate();
192. JOptionPane.showMessageDialog(null, "Data Berhasil Disimpan");
193. kosong();
194. txtidplgn.requestFocus();
195. }
196. catch (SQLException e){
197. JOptionPane.showMessageDialog(null, "Data Gagal Disimpan"+e);
198. }
199. datatable();
200. }

201. private void bubahActionPerformed(java.awt.event.ActionEvent evt) {
202. try{
203. String sql = "update pelanggan set idplgn=? ,nmplgn=? ,notlp=? ,alamat=? where
    idplgn='"+txtidplgn.getText()+"'";
    PreparedStatement stat = conn.prepareStatement(sql);
    stat.setString(1, txtidplgn.getText());
    stat.setString(2, txtnmplgn.getText());
    stat.setString(3, txtnotlp.getText());
    stat.setString(4, txtalamat.getText());

    stat.executeUpdate();
    JOptionPane.showMessageDialog(null, "Data Berhasil Diubah");
    kosong();
    txtidplgn.requestFocus();
204. }
205. catch (SQLException e){
206. JOptionPane.showMessageDialog(null, "Data Gagal Diubah"+e);
207. }
208. datatable();
209. }

210. private void bhapusActionPerformed(java.awt.event.ActionEvent evt) {
211. int ok = JOptionPane.showConfirmDialog(null,"hapus","konfirmasi
    dialog",JOptionPane.YES_NO_OPTION);
212. if (ok==0){
    String sql = "delete from pelanggan where idplgn = '"+txtidplgn.getText()+"'";
    try{
    PreparedStatement stat = conn.prepareStatement(sql);
    stat.executeUpdate();
    JOptionPane.showMessageDialog(null, "data berhasil dihapus");
    kosong();
    txtidplgn.requestFocus();
    }
    catch (SQLException e){
    JOptionPane.showMessageDialog(null, "data gagal dihapus"+e);
    }
    datatable();
213. }

```

```

214. }

215. private void bcariKeyPressed(java.awt.event.KeyEvent evt) {
216. if (evt.getKeyCode() == KeyEvent.VK_ENTER){
217. datatable();
218. }
219. }

220. private void bcariActionPerformed(java.awt.event.ActionEvent evt) {
221. datatable();
222. }

223. private void tbplgnMouseClicked(java.awt.event.MouseEvent evt) {
224. int bar = tbplgn.getSelectedRow();
225. String a = tabmode.getValueAt(bar, 0).toString();
226. String b = tabmode.getValueAt(bar, 1).toString();
227. String c = tabmode.getValueAt(bar, 2).toString();
228. String d = tabmode.getValueAt(bar, 3).toString();

229. txtidplgn.setText(a);
230. txtmplgn.setText(b);
231. txtnotlp.setText(c);
232. txtalamat.setText(d);
233. }

234. private void bkeluarActionPerformed(java.awt.event.ActionEvent evt) {
235. this.setVisible(false);
236. new menu().setVisible(true);
237. }

238. private void bcetakActionPerformed(java.awt.event.ActionEvent evt) {
239. try {
240. HashMap parameter = new HashMap();
241. Class.forName("com.mysql.jdbc.Driver");
242. Connection connect =
        DriverManager.getConnection("jdbc:mysql://localhost/laundry","root","");
243. File file = new File("src/Laporan/reportpelanggan.jasper");
244. JasperReport jr = (JasperReport) JRLoader.loadObject(file);
245. JasperPrint jp = JasperFillManager.fillReport(jr, parameter, connect);
246. JasperViewer.viewReport(jp, false);
247. JasperViewer.setLookAndFeelDecorated(true);
248. }catch(ClassNotFoundException | SQLException | JRException e) {
249. javax.swing.JOptionPane.showMessageDialog(null, "Data tidak dapat di cetak
        !!!"+'\n'+e.getMessage(), "Cetak data", javax.swing.JOptionPane.ERROR_MESSAGE);
250. }
251. }

252. /**
     * @param args the command line arguments
253. */
254. public static void main(String args[]) {
255. /* Set the Nimbus look and feel */
256. //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

```

```

257. /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and
feel.
       For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
258. */
259. try {
260. for (javax.swing.UIManager.LookAndFeelInfo info :
261.         javax.swing.UIManager.getInstalledLookAndFeels()) {
262.     if ("Nimbus".equals(info.getName())) {
263.         javax.swing.UIManager.setLookAndFeel(info.getClassName());
264.         break;
265.     }
266. } catch (ClassNotFoundException | InstantiationException | IllegalAccessException |
267.         javax.swing.UnsupportedLookAndFeelException ex) {
268.     java.util.logging.Logger.getLogger(Pelanggan.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
269. }
270. });
271. }

272. // Variables declaration - do not modify
273. private javax.swing.JButton bbatal;
274. private javax.swing.JButton bcari;
275. private javax.swing.JButton bcetak;
276. private javax.swing.JButton bhapus;
277. private javax.swing.JButton bkeluar;
278. private javax.swing.JButton bsimpan;
279. private javax.swing.JButton bubah;
280. private javax.swing.JLabel jLabel1;
281. private javax.swing.JLabel jLabel2;
282. private javax.swing.JLabel jLabel3;
283. private javax.swing.JLabel jLabel4;
284. private javax.swing.JLabel jLabel5;
285. private javax.swing.JLabel jLabel6;
286. private javax.swing.JLabel jLabel7;
287. private javax.swing.JScrollPane jScrollPane1;
288. private javax.swing.JScrollPane jScrollPane2;
289. private javax.swing.JTable tbplgn;
290. private javax.swing.JTextArea txtalamat;
291. private javax.swing.JTextField txtcari;
292. private javax.swing.JTextField txtidplgn;
293. private javax.swing.JTextField txtmplgn;
294. private javax.swing.JTextField txtnotlp;
295. // End of variables declaration
296. }

```

#### **F. Transaksi.java**

```
1. package Form;
2. import Koneksi.koneksi;
3. import java.awt.HeadlessException;
4. import java.awt.event.KeyEvent;
5. import java.io.File;
6. import java.sql.Connection;
7. import java.sql.PreparedStatement;
8. import java.sql.ResultSet;
9. import java.sql.SQLException;
10. import java.sql.Statement;
11. import java.text.NumberFormat;
12. import java.text.SimpleDateFormat;
13. import java.util.Date;
14. import java.util.HashMap;
15. import java.util.Locale;
16. import javax.swing.JOptionPane;
17. import javax.swing.table.DefaultTableModel;
18. import net.sf.jasperreports.engine.JRException;
19. import net.sf.jasperreports.engine.JasperCompileManager;
20. import net.sf.jasperreports.engine.JasperFillManager;
21. import net.sf.jasperreports.engine.JasperPrint;
22. import net.sf.jasperreports.engine.JasperReport;
23. import net.sf.jasperreports.engine.design.JasperDesign;
24. import net.sf.jasperreports.engine.xml.JRXmlLoader;
25. import net.sf.jasperreports.view.JasperViewer;

26. /**
27. *
28. * @author USER
29. */
29. public final class Transaksi extends javax.swing.JFrame {
30. private final Connection conn = new koneksi().connect();
31. private final DefaultTableModel model;
32. public String tanggal;
33. public String id_karyawan,nama_karyawan, id_pelanggan, nama_pelanggan, id_barang,
nama_barang, tarif;
34. java.util.Date utilDate = new java.util.Date();
35. java.sql.Date sqlDate = new java.sql.Date(utilDate.getTime());
36. String keterangan="Belum Diambil";
37. /**
38. * Creates new form Transaksi
39. */
39. public Transaksi() {
40. initComponents();
41. kosong();
42. aktif();
43. autonumber();
44. Date skrg=new Date();
45. SimpleDateFormat frm=new SimpleDateFormat("dd MMMM yyyy");
46. String t=frm.format(skrg);
47. tgltrans.setText(t);
48. model=new DefaultTableModel();
49. jTable1.setModel(model);
50. model.addColumn("ID Jenis");
```

```
51. model.addColumn("Nama Barang");
52. model.addColumn("Tarif Perkilo");
53. model.addColumn("Berat");
54. model.addColumn("Sub Total");
55. notrans.setEditable(false);
56. tgltrans.setEditable(false);
57. }
58. public String id_pelanggan() {
59. return id_pelanggan;
60. }
61. public String nama_pelanggan() {
62. return nama_pelanggan;
63. }
64. public String id_karyawan() {
65. return id_karyawan;
66. }
67. public String nama_karyawan() {
68. return nama_karyawan;
69. }
70. public String id_barang() {
71. return id_barang;
72. }
73. public String nama_barang() {
74. return nama_barang;
75. }
76. public String tarif() {
77. return tarif;
78. }
79. public void itemTerpilih(){
80. popuppelanggan pop = new popuppelanggan();
81. pop.FN = this;
82. idpel.setText(id_pelanggan);
83. namapel.setText(nama_pelanggan);
84. idkar.setText(id_karyawan);
85. namakar.setText(nama_karyawan);
86. idjen.setText(id_barang);
87. namabar.setText(nama_barang);
88. tarifbar.setText(tarif);

89. }
90. protected void aktif(){
91. idpel.requestFocus();
92. }
93. protected void kosong(){
94. idjen.setText("");
95. namabar.setText("");
96. tarifbar.setText("");
97. beratbar.setText("");
98. subtot.setText("");
99. }
100. public void TambahData() {
101. String kode=notrans.getText();
102. String KodeBar=idjen.getText();
103. String namaBar=namabar.getText();
104. String Harga=tarifbar.getText();
```

```

105. String berat=beratbar.getText();
106. String subtotal=subtot.getText();
107. String digits = subtotal.replaceAll("[^0-9]", "");

108. try {
109. String sql1="Select * from transaksi where no_transaksi='"+ kode + "' AND
    id_barang='"+ KodeBar + "'";
110. java.sql.Statement stat1 = conn.createStatement();
111. ResultSet rs = stat1.executeQuery(sql1);
112. if (rs.next()){
    try{
        String sql ="update transaksi set berat_barang=?, subtotal=subtotal+? where
        id_barang='"+KodeBar+"' and id_transaksi='"+kode+"'";
        PreparedStatement stat = conn.prepareStatement(sql);
        stat.setInt (1, Integer.valueOf(berat));
        stat.setString (2, digits);

        stat.executeUpdate();

    }catch(SQLException e){
        JOptionPane.showMessageDialog(null, "Data Gagal Disimpan"+e);
    }
113. }else{
    try{
        String sql ="insert into transaksi values (?,?,?,?,?,?,?,?,?,?,?,?,?,?)";
        PreparedStatement stat = conn.prepareStatement(sql);
        stat.setString (1, notrans.getText());
        stat.setDate(2, sqlDate);
        stat.setInt (3, Integer.valueOf(idpel.getText()));
        stat.setString (4, namapel.getText());
        stat.setString (5, idkar.getText());
        stat.setString (6, namakar.getText());
        stat.setInt (7, Integer.valueOf(idjen.getText()));
        stat.setString (8, namabar.getText());
        stat.setInt (9, Integer.valueOf(tarifbar.getText()));
        stat.setInt (10, Integer.valueOf(beratbar.getText()));
        stat.setString (11, digits);
        stat.setString (12, diskon.getText());
        stat.setString (13, keterangan);

        stat.executeUpdate();

    }catch(SQLException e){
        JOptionPane.showMessageDialog(null, "Data Gagal Disimpan"+e);
    }
114. }
115. }catch(SQLException e){
116. System.out.println(e);
117. }finally{
118. loadData();
119. }
120. }
121. public void loadData() {
122. model.getDataVector().removeAllElements();
123. model.fireTableDataChanged();

```

```

124. try {
125.     String sql="Select * from transaksi where no_transaksi='"+notrans.getText() + "'";
126.     java.sql.Statement stat = conn.createStatement();
127.     try (ResultSet hasil = stat.executeQuery(sql)) {
128.         while (hasil.next()) {
129.             Object[] o=new Object[5];
130.             o[0]=hasil.getString("id_barang");
131.             o[1]=hasil.getString("nama_barang");
132.             o[2]=hasil.getString("tarif");
133.             o[3]=hasil.getString("berat");
134.             o[4]=hasil.getString("total");

135.             model.addRow(o);
136.         }
137.     }
138. }

139. }catch(SQLException e) {
140.     System.out.println("Terjadi kesalahan"+e);
141. }
142. }

143. protected void autonumber(){
144.     try {
145.         String sql = "SELECT no_transaksi from transaksi order by no_transaksi asc";
146.         Statement st = conn.createStatement();
147.         ResultSet rs = st.executeQuery(sql);
148.         notrans.setText("TR-001");
149.         while (rs.next()) {
150.             String notransaksi = rs.getString("no_transaksi").substring(3);
151.             int AN = Integer.parseInt(notransaksi) + 1;
152.             String Nol = "";
153.

154.             if(AN<10)
155.                 {Nol = "00";}
156.             else if(AN<100)
157.                 {Nol = "0";}
158.             else if(AN<1000)
159.                 {Nol = "0";}
160.             else if(AN<10000)
161.                 {Nol = "";}
162.

163.             notrans.setText("TR-" + Nol + AN);
164.         }
165.     }catch(NumberFormatException | SQLException e){
166.         JOptionPane.showMessageDialog(null, "Auto Number Gagal"+e);
167.     }
168. }
169. }

170. /**
171. This method is called from within the constructor to initialize the form.
172. WARNING: Do NOT modify this code. The content of this method is always
173. regenerated by the Form Editor.
174. */
175. @SuppressWarnings("unchecked")
176. // <editor-fold defaultstate="collapsed" desc="Generated Code">
177. private void initComponents() {

```

```
150. jPanel1 = new javax.swing.JPanel();
151. jLabel1 = new javax.swing.JLabel();
152. jLabel2 = new javax.swing.JLabel();
153. jLabel5 = new javax.swing.JLabel();
154. notrans = new javax.swing.JTextField();
155. idpel = new javax.swing.JTextField();
156. jLabel6 = new javax.swing.JLabel();
157. btTambah = new javax.swing.JButton();
158. btn cetak = new javax.swing.JButton();
159. btn keluar = new javax.swing.JButton();
160. jScrollPane1 = new javax.swing.JScrollPane();
161. jTable1 = new javax.swing.JTable();
162. btn batal = new javax.swing.JButton();
163. jLabel7 = new javax.swing.JLabel();
164. jLabel8 = new javax.swing.JLabel();
165. namapel = new javax.swing.JTextField();
166. idjen = new javax.swing.JTextField();
167. jLabel9 = new javax.swing.JLabel();
168. jLabel10 = new javax.swing.JLabel();
169. jLabel11 = new javax.swing.JLabel();
170. jLabel12 = new javax.swing.JLabel();
171. namabar = new javax.swing.JTextField();
172. tarifbar = new javax.swing.JTextField();
173. beratbar = new javax.swing.JTextField();
174. subtot = new javax.swing.JTextField();
175. jLabel13 = new javax.swing.JLabel();
176. btncaripelanggan = new javax.swing.JButton();
177. btncaribarang = new javax.swing.JButton();
178. tgltrans = new javax.swing.JTextField();
179. jLabel15 = new javax.swing.JLabel();
180. jLabel16 = new javax.swing.JLabel();
181. txttotal = new javax.swing.JTextField();
182. txtbayar = new javax.swing.JTextField();
183. txtkembalian = new javax.swing.JTextField();
184. jLabel17 = new javax.swing.JLabel();
185. diskon = new javax.swing.JTextField();
186. jLabel18 = new javax.swing.JLabel();
187. jLabel14 = new javax.swing.JLabel();
188. namakar = new javax.swing.JTextField();
189. btncaripelanggan1 = new javax.swing.JButton();
190. idkar = new javax.swing.JTextField();
191. jLabel19 = new javax.swing.JLabel();

192. setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

193. jPanel1.setBackground(new java.awt.Color(102, 102, 102));
194. jPanel1.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(0, 0, 0), 2));

195. jLabel1.setBackground(new java.awt.Color(204, 0, 102));
196. jLabel1.setFont(new java.awt.Font("Times New Roman", 1, 24)); // NOI18N
197. jLabel1.setForeground(new java.awt.Color(255, 255, 255));
198. jLabel1.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
199. jLabel1.setText("TRANSAKSI");
```

```
200. jLabel2.setFont(new java.awt.Font("Times New Roman", 0, 13)); // NOI18N
201. jLabel2.setForeground(new java.awt.Color(255, 255, 255));
202. jLabel2.setText("No. Transaksi");

203. jLabel5.setFont(new java.awt.Font("Times New Roman", 0, 13)); // NOI18N
204. jLabel5.setForeground(new java.awt.Color(255, 255, 255));
205. jLabel5.setText("Tanggal Transaksi");

206. notrans.setFont(new java.awt.Font("Times New Roman", 0, 13)); // NOI18N

207. idpel.setFont(new java.awt.Font("Times New Roman", 0, 13)); // NOI18N
208. idpel.addActionListener(new java.awt.event.ActionListener() {
209.     public void actionPerformed(java.awt.event.ActionEvent evt) {
210.         idpelActionPerformed(evt);
211.     });
212. });

213. jLabel6.setFont(new java.awt.Font("Times New Roman", 0, 13)); // NOI18N
214. jLabel6.setForeground(new java.awt.Color(255, 255, 255));
215. jLabel6.setText("ID Pelanggan");

216. btTambah.setFont(new java.awt.Font("Times New Roman", 1, 13)); // NOI18N
217. btTambah.setText("Tambah");
218. btTambah.addActionListener(new java.awt.event.ActionListener() {
219.     public void actionPerformed(java.awt.event.ActionEvent evt) {
220.         btTambahActionPerformed(evt);
221.     });
222. });

223. btncetak.setFont(new java.awt.Font("Times New Roman", 1, 13)); // NOI18N
224. btncetak.setText("Kwitansi");
225. btncetak.addActionListener(new java.awt.event.ActionListener() {
226.     public void actionPerformed(java.awt.event.ActionEvent evt) {
227.         btncetakActionPerformed(evt);
228.     });
229. });

230. btnkeluar.setFont(new java.awt.Font("Times New Roman", 1, 13)); // NOI18N
231. btnkeluar.setText("Keluar");
232. btnkeluar.addActionListener(new java.awt.event.ActionListener() {
233.     public void actionPerformed(java.awt.event.ActionEvent evt) {
234.         btnkeluarActionPerformed(evt);
235.     });
236. });

237. jTable1.setFont(new java.awt.Font("Times New Roman", 0, 13)); // NOI18N
238. jTable1.setModel(new javax.swing.table.DefaultTableModel(
239.     new Object [][] {
240.         {null, null, null, null},
241.         {null, null, null, null},
242.         {null, null, null, null},
243.         {null, null, null, null}
244.     },
245.     new String [] {
246.         ""
247.     }
248. ));
```

```
    "Title 1", "Title 2", "Title 3", "Title 4"
238. }
239. ));
240. jTable1.addMouseListener(new java.awt.event.MouseAdapter() {
241.     public void mouseClicked(java.awt.event.MouseEvent evt) {
242.         jTable1MouseClicked(evt);
243.     });
244. jScrollPane1.setViewportView(jTable1);

245. btnbatal.setFont(new java.awt.Font("Times New Roman", 1, 13)); // NOI18N
246. btnbatal.setText("Batal");
247. btnbatal.addActionListener(new java.awt.event.ActionListener() {
248.     public void actionPerformed(java.awt.event.ActionEvent evt) {
249.         btnbatalActionPerformed(evt);
250.     });
251. jLabel7.setFont(new java.awt.Font("Times New Roman", 0, 13)); // NOI18N
252. jLabel7.setForeground(new java.awt.Color(255, 255, 255));
253. jLabel7.setText("Nama Pelanggan");

254. jLabel8.setFont(new java.awt.Font("Times New Roman", 0, 13)); // NOI18N
255. jLabel8.setForeground(new java.awt.Color(255, 255, 255));
256. jLabel8.setText("ID Jenis");

257. namapel.setFont(new java.awt.Font("Times New Roman", 0, 13)); // NOI18N
258. namapel.addActionListener(new java.awt.event.ActionListener() {
259.     public void actionPerformed(java.awt.event.ActionEvent evt) {
260.         namapelActionPerformed(evt);
261.     });
262. idjen.setFont(new java.awt.Font("Times New Roman", 0, 13)); // NOI18N
263. idjen.addActionListener(new java.awt.event.ActionListener() {
264.     public void actionPerformed(java.awt.event.ActionEvent evt) {
265.         idjenActionPerformed(evt);
266.     });
267. jLabel9.setFont(new java.awt.Font("Times New Roman", 0, 13)); // NOI18N
268. jLabel9.setForeground(new java.awt.Color(255, 255, 255));
269. jLabel9.setText("Nama Barang");

270. jLabel10.setFont(new java.awt.Font("Times New Roman", 0, 13)); // NOI18N
271. jLabel10.setForeground(new java.awt.Color(255, 255, 255));
272. jLabel10.setText("Tarif");

273. jLabel11.setFont(new java.awt.Font("Times New Roman", 0, 13)); // NOI18N
274. jLabel11.setForeground(new java.awt.Color(255, 255, 255));
275. jLabel11.setText("Berat Barang (kg)");

276. jLabel12.setFont(new java.awt.Font("Times New Roman", 0, 13)); // NOI18N
277. jLabel12.setForeground(new java.awt.Color(255, 255, 255));
278. jLabel12.setText("Subtotal");
```

```
279. namabar.setFont(new java.awt.Font("Times New Roman", 0, 13)); // NOI18N
280. namabar.addActionListener(new java.awt.event.ActionListener() {
281.     public void actionPerformed(java.awt.event.ActionEvent evt) {
282.         namabarActionPerformed(evt);
283.     });
284.
285. tarifbar.setFont(new java.awt.Font("Times New Roman", 0, 13)); // NOI18N
286. tarifbar.addActionListener(new java.awt.event.ActionListener() {
287.     public void actionPerformed(java.awt.event.ActionEvent evt) {
288.         tarifbarActionPerformed(evt);
289.
290. beratbar.setFont(new java.awt.Font("Times New Roman", 0, 13)); // NOI18N
291. beratbar.addActionListener(new java.awt.event.ActionListener() {
292.     public void actionPerformed(java.awt.event.ActionEvent evt) {
293.         beratbarActionPerformed(evt);
294.         beratbar.addKeyListener(new java.awt.event.KeyAdapter() {
295.             public void keyReleased(java.awt.event.KeyEvent evt) {
296.                 beratbarKeyReleased(evt);
297.             });
298.
299. subtot.setFont(new java.awt.Font("Times New Roman", 0, 13)); // NOI18N
300. subtot.addActionListener(new java.awt.event.ActionListener() {
301.     public void actionPerformed(java.awt.event.ActionEvent evt) {
302.         subtotActionPerformed(evt);
303.
304. jLabel13.setForeground(new java.awt.Color(255, 255, 255));
305. btncaripelanggan.setFont(new java.awt.Font("Times New Roman", 1, 13)); // NOI18N
306. btncaripelanggan.setText("Cari");
307. btncaripelanggan.addActionListener(new java.awt.event.ActionListener() {
308.     public void actionPerformed(java.awt.event.ActionEvent evt) {
309.         btncaripelangganActionPerformed(evt);
310.     });
311.
312. btncaribarang.setFont(new java.awt.Font("Times New Roman", 1, 13)); // NOI18N
313. btncaribarang.setText("Cari");
314. btncaribarang.addActionListener(new java.awt.event.ActionListener() {
315.     public void actionPerformed(java.awt.event.ActionEvent evt) {
316.         btncaribarangActionPerformed(evt);
317.
318. jLabel15.setForeground(new java.awt.Color(255, 255, 255));
319. jLabel15.setText("Total ");
```

```
319. jLabel16.setForeground(new java.awt.Color(255, 255, 255));
320. jLabel16.setText("Uang Kembalian");

321. txttotal.addActionListener(new java.awt.event.ActionListener() {
322.     public void actionPerformed(java.awt.event.ActionEvent evt) {
            txttotalActionPerformed(evt);
323. }
324. });

325. txtbayar.addKeyListener(new java.awt.event.KeyAdapter() {
326.     public void keyPressed(java.awt.event.KeyEvent evt) {
            txtbayarKeyPressed(evt);
327. }
328. });

329. txtkembalian.addActionListener(new java.awt.event.ActionListener() {
330.     public void actionPerformed(java.awt.event.ActionEvent evt) {
            txtkembalianActionPerformed(evt);
331. }
332. });

333. jLabel17.setForeground(new java.awt.Color(255, 255, 255));
334. jLabel17.setText("Bayar");

335. diskon.addActionListener(new java.awt.event.ActionListener() {
336.     public void actionPerformed(java.awt.event.ActionEvent evt) {
            diskonActionPerformed(evt);
337. }
338. });

339. jLabel18.setForeground(new java.awt.Color(255, 255, 255));
340. jLabel18.setText("Diskon");

341. jLabel14.setFont(new java.awt.Font("Times New Roman", 0, 13)); // NOI18N
342. jLabel14.setForeground(new java.awt.Color(255, 255, 255));
343. jLabel14.setText("Nama Karyawan");

344. namakar.setFont(new java.awt.Font("Times New Roman", 0, 13)); // NOI18N
345. namakar.addActionListener(new java.awt.event.ActionListener() {
346.     public void actionPerformed(java.awt.event.ActionEvent evt) {
            namakarActionPerformed(evt);
347. }
348. });

349. btncaripelanggan1.setFont(new java.awt.Font("Times New Roman", 1, 13)); // NOI18N
350. btncaripelanggan1.setText("Cari");
351. btncaripelanggan1.addActionListener(new java.awt.event.ActionListener() {
352.     public void actionPerformed(java.awt.event.ActionEvent evt) {
            btncaripelanggan1ActionPerformed(evt);
353. }
354. });

355. idkar.setFont(new java.awt.Font("Times New Roman", 0, 13)); // NOI18N
356. idkar.addActionListener(new java.awt.event.ActionListener() {
```



```

.addComponent(jLabel7)
.addComponent(jLabel6)
.addGroup(jPanel1Layout.createSequentialGroup())
374. .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
ADING)
.addComponent(jLabel2)
.addComponent(jLabel5))
.addGap(27, 27, 27)
375. .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING,
false)
.addComponent(notrans, javax.swing.GroupLayout.DEFAULT_SIZE, 145,
Short.MAX_VALUE)
.addComponent(tgltrans)))
.addComponent(jLabel14)
.addComponent(jLabel19)
.addGroup(jPanel1Layout.createSequentialGroup()
.addGap(124, 124, 124)
376. .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addComponent(jPanel1Layout.createSequentialGroup())
377. .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING,
false)
.addComponent(idpel, javax.swing.GroupLayout.Alignment.LEADING,
javax.swing.GroupLayout.DEFAULT_SIZE, 146, Short.MAX_VALUE)
.addComponent(namapel, javax.swing.GroupLayout.Alignment.LEADING))
.addGap(18, 18, 18)
.addComponent(btncaripelanggan))
.addGroup(jPanel1Layout.createSequentialGroup())
378. .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING,
false)
.addComponent(idkar, javax.swing.GroupLayout.Alignment.LEADING)
.addComponent(namakar, javax.swing.GroupLayout.Alignment.LEADING,
javax.swing.GroupLayout.PREFERRED_SIZE, 146,
javax.swing.GroupLayout.PREFERRED_SIZE))
.addGap(18, 18, 18)
.addComponent(btncaripelanggan1))))
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 61,
Short.MAX_VALUE)
379. .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addComponent(jPanel1Layout.createSequentialGroup())
380. .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addComponent(jLabel18)
.addComponent(jLabel15))
.addGap(73, 73, 73)
381. .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addComponent(diskon, javax.swing.GroupLayout.PREFERRED_SIZE, 170,
javax.swing.GroupLayout.PREFERRED_SIZE)
.addComponent(txttotal, javax.swing.GroupLayout.PREFERRED_SIZE, 170,
javax.swing.GroupLayout.PREFERRED_SIZE)))
382. .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING,
false)
.addComponent(jPanel1Layout.createSequentialGroup())

```

```
.addComponent(btTambah)
.addComponent(btnbatal)
383. .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
.addComponent(btnkeluar))
.addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 577,
javax.swing.GroupLayout.PREFERRED_SIZE))
384. .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
.addComponent(btncetak)
.addGroup(jPanel1Layout.createSequentialGroup())
385. .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addComponent(jPanel1Layout.createSequentialGroup()
.addComponent(jLabel16)
.addGap(18, 18, 18))
.addComponent(jPanel1Layout.createSequentialGroup()
.addComponent(jLabel17)
.addGap(78, 78, 78)))
386. .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING,
false)
.addComponent(txtkembalian)
.addComponent(txtbayar, javax.swing.GroupLayout.PREFERRED_SIZE, 170,
javax.swing.GroupLayout.PREFERRED_SIZE))))))
.addContainerGap()))
387. );
388. jPanel1Layout.setVerticalGroup(
389. jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
390. .addGroup(jPanel1Layout.createSequentialGroup()
.addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 52,
javax.swing.GroupLayout.PREFERRED_SIZE)
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
391. .addGroup(jPanel1Layout.createSequentialGroup()
.addComponent(jPanel1Layout.createSequentialGroup()
.addComponent(txttotal, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
.addGap(21, 21, 21)
392. .addGroup(jPanel1Layout.createSequentialGroup()
.addComponent(jPanel1Layout.createSequentialGroup()
.addComponent(jLabel18)
.addComponent(diskon, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
.addGap(21, 21, 21)
393. .addGroup(jPanel1Layout.createSequentialGroup()
.addComponent(jPanel1Layout.createSequentialGroup()
.addComponent(jLabel17)
```

```
.addComponent(txtbayar, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
.addGap(21, 21, 21)
394. .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addComponent(jLabel16)
.addComponent(txtkembalian, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
.addGap(39, 39, 39)
395. .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
.addComponent(btTambah)
.addComponent(btnbatal)
.addComponent(btnkeluar)
.addComponent(btncetak)))
.addGroup(jPanel1Layout.createSequentialGroup()
.addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 138,
javax.swing.GroupLayout.PREFERRED_SIZE)
.addGap(18, 18, 18)
.addComponent(jLabel15))
.addGroup(jPanel1Layout.createSequentialGroup()
396. .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
.addComponent(jLabel2)
.addComponent(notrans, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
397. .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
398. .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
.addComponent(jLabel5)
.addComponent(tgltrans, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 20,
Short.MAX_VALUE)
399. .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addComponent(jLabel13, javax.swing.GroupLayout.Alignment.TRAILING)
.addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel1Layout.createSequentialGroup())
400. .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
.addComponent(jLabel6)
.addComponent(idpel, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
.addComponent(btncaripelanggan))
401. .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
402. .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
.addComponent(jLabel7)
```

```
.addComponent(namapel, javax.swing.GroupLayout.PREFERRED_SIZE,
    javax.swing.GroupLayout.DEFAULT_SIZE,
    javax.swing.GroupLayout.PREFERRED_SIZE))
403. .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
404. .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(jLabel19)
    .addComponent(idkar, javax.swing.GroupLayout.PREFERRED_SIZE,
        javax.swing.GroupLayout.DEFAULT_SIZE,
        javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(btncaripelanggan1))
405. .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
406. .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(jLabel14)
    .addComponent(namakar, javax.swing.GroupLayout.PREFERRED_SIZE,
        javax.swing.GroupLayout.DEFAULT_SIZE,
        javax.swing.GroupLayout.PREFERRED_SIZE))
407. .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
408. .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(idjen, javax.swing.GroupLayout.PREFERRED_SIZE,
        javax.swing.GroupLayout.DEFAULT_SIZE,
        javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(jLabel8)
    .addComponent(btncaribarang))
409. .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
410. .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(namabar, javax.swing.GroupLayout.PREFERRED_SIZE,
        javax.swing.GroupLayout.DEFAULT_SIZE,
        javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(jLabel9))
411. .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
412. .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(tarifbar, javax.swing.GroupLayout.PREFERRED_SIZE,
        javax.swing.GroupLayout.DEFAULT_SIZE,
        javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(jLabel10)))
413. .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
414. .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(beratbar, javax.swing.GroupLayout.PREFERRED_SIZE,
        javax.swing.GroupLayout.DEFAULT_SIZE,
        javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(jLabel11))
415. .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
416. .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(subtot, javax.swing.GroupLayout.PREFERRED_SIZE,
        javax.swing.GroupLayout.DEFAULT_SIZE,
        javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(jLabel12)))
    .addContainerGap(130, Short.MAX_VALUE))
```

```
417. );
418. javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
419. getContentPane().setLayout(layout);
420. layout.setHorizontalGroup(
421. layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
422. .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE,
423. javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
423. );
424. layout.setVerticalGroup(
425. layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
426. .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
427. javax.swing.GroupLayout.DEFAULT_SIZE,
427. javax.swing.GroupLayout.PREFERRED_SIZE)
427. );
428. setSize(new java.awt.Dimension(1030, 636));
429. setLocationRelativeTo(null);
430. } // </editor-fold>
431. private void idpelActionPerformed(java.awt.event.ActionEvent evt) {
432. // TODO add your handling code here:
433. }
434. private void btTambahActionPerformed(java.awt.event.ActionEvent evt) {
435. TambahData();
436. loadData();
437. NumberFormat nf3 = NumberFormat.getInstance(new Locale("da", "DK"));
438. int b=0;
439. int total = 0;
440. for (int i =0; i< jTable1.getRowCount(); i++){
441.     int amount = Integer.parseInt((String)jTable1.getValueAt(i, 4));
442.     total += amount;
443. }
444. txttotal.setText("Rp. "+nf3.format(total));
445. kosong();
446. }
447. }
448. private void btnbatalActionPerformed(java.awt.event.ActionEvent evt) {
449. this.setVisible(false);
450. new menu().setVisible(true);
451. }
452. private void jTable1MouseClicked(java.awt.event.MouseEvent evt) {
453. int i=jTable1.getSelectedRow();
454. if(i==-1){
455. return;
456. }
457. String kode=notrans.getText();
```

```
458. String kodebar=(String) model.getValueAt(i, 0);
459. String jml=(String) model.getValueAt(i, 2);
460. String sql="Delete from transaksi where no_transaksi=? AND id_barang=?";
461. try {
462. PreparedStatement st2 = conn.prepareStatement(sql);
463. st2.setString(1, kode);
464. st2.setString(2, kodebar);
465. st2.executeUpdate();

466. }catch(SQLException e) {
467. System.out.println(e);
468. }finally{
469. loadData();
470. }
471. NumberFormat nf3 = NumberFormat.getInstance(new Locale("da", "DK"));
472. int b=0;
473. int total = 0;
474. for (int n =0; n<jTable1.getRowCount(); n++){
475. int amount = Integer.parseInt((String)jTable1.getValueAt(n, 4));
476. total += amount;
477. }
478. txttotal.setText("Rp. "+nf3.format(total));
479. }

480. private void namapelActionPerformed(java.awt.event.ActionEvent evt) {
481. // TODO add your handling code here:
482. }

483. private void idjenActionPerformed(java.awt.event.ActionEvent evt) {
484. // TODO add your handling code here:
485. }

486. private void namabarActionPerformed(java.awt.event.ActionEvent evt) {
487. // TODO add your handling code here:
488. }

489. private void tarifbarActionPerformed(java.awt.event.ActionEvent evt) {
490. // TODO add your handling code here:
491. }

492. private void beratbarActionPerformed(java.awt.event.ActionEvent evt) {
493. }

494. private void subtotActionPerformed(java.awt.event.ActionEvent evt) {
495. // TODO add your handling code here:
496. }

497. private void btncaripelangganActionPerformed(java.awt.event.ActionEvent evt) {
498. // TODO add your handling code here:
499. popuppelanggan pop = new popuppelanggan();
500. pop.FN = this;
501. pop.setVisible(true);
502. pop.setResizable(false);
503. diskon.setText("10%");
```

```

504. }

505. private void btncaribarangActionPerformed(java.awt.event.ActionEvent evt) {
506. // TODO add your handling code here:
507. popupjenisbarang pop = new popupjenisbarang();
508. pop.FN = this;
509. pop.setVisible(true);
510. pop.setResizable(false);
511. }

512. private void btncetakActionPerformed(java.awt.event.ActionEvent evt) {
513. String notran = notrans.getText();
514. int a=Integer.valueOf(txtbayar.getText());
515. NumberFormat nf3 = NumberFormat.getInstance(new Locale("da", "DK"));
516. String val3 = nf3.format(a);
517. try{
518. String sql = "SELECT * FROM transaksi where no_transaksi='"+notran+"'";
519. Statement stat = conn.createStatement();
520. ResultSet hasil = stat.executeQuery(sql);
521. if (hasil.next()){
522. try{
523. File file = new File ("src/Laporan/BuktiPembayaran.jrxml");
524. Class.forName("com.mysql.jdbc.Driver");
525. HashMap parameter = new HashMap();
526. parameter.put("id",notran);
527. parameter.put("byr",val3);
528. parameter.put("kembali",txtkembalian.getText());
529. JasperDesign jasperDesign = JRXmlLoader.load(file);
530. JasperReport jasperReport = JasperCompileManager.compileReport(jasperDesign);
531. JasperPrint jasperPrint = JasperFillManager.fillReport(jasperReport, parameter, conn);
532. JasperViewer.viewReport(jasperPrint,false);
533. }catch (ClassNotFoundException | JRException e){
534. JOptionPane.showMessageDialog(null,"Data Tidak Ada!"+e.getMessage(),"Cetak
535. Data",JOptionPane.ERROR_MESSAGE);
536. }
537. }else{
538. try{
539. File file = new File ("src/Laporan/reporttransaksi.jrxml");
540. Class.forName("com.mysql.jdbc.Driver");
541. HashMap parameter = new HashMap();
542. JasperDesign jasperDesign = JRXmlLoader.load(file);
543. JasperReport jasperReport = JasperCompileManager.compileReport(jasperDesign);
544. JasperPrint jasperPrint = JasperFillManager.fillReport(jasperReport, parameter, conn);
545. JasperViewer.viewReport(jasperPrint,false);
546. }catch (ClassNotFoundException | JRException e){
547. JOptionPane.showMessageDialog(null,"Data Tidak Ada!"+e.getMessage(),"Cetak
548. Data",JOptionPane.ERROR_MESSAGE);
549. }
550. }
551. }catch (HeadlessException | SQLException e){
552. JOptionPane.showMessageDialog(null,"Data Tidak Ada!"+e.getMessage(),"Cetak
553. Data",JOptionPane.ERROR_MESSAGE);
554. }
555. }

```

```
527. private void txttotalActionPerformed(java.awt.event.ActionEvent evt) {  
528. // TODO add your handling code here:  
529. }  
  
530. private void txtbayarKeyPressed(java.awt.event.KeyEvent evt) {  
531. if(evt.getKeyCode()==KeyEvent.VK_ENTER){  
532. int total=Integer.parseInt(txttotal.getText().replaceAll("[^0-9]", ""));  
533. int bayar=Integer.parseInt(txtbayar.getText());  
534. if(bayar<total){  
      JOptionPane.showMessageDialog(null, "Jumlah bayar tidak mencukupi");  
      txtbayar.requestFocus();  
535. } else{  
      String digits = txttotal.getText().replaceAll("[^0-9]", "");  
      int a=Integer.parseInt(txtbayar.getText())-Integer.parseInt(digits);  
      NumberFormat nf3 = NumberFormat.getInstance(new Locale("da", "DK"));  
      String val3 = nf3.format(a);  
      txtkembalian.setText("Rp. "+val3);  
536. }  
537. } // TODO add your handling code here:  
538. }  
  
539. private void txtkembalianActionPerformed(java.awt.event.ActionEvent evt) {  
540. // TODO add your handling code here:  
541. }  
  
542. private void beratbarKeyReleased(java.awt.event.KeyEvent evt) {  
543. try{  
544. String sql = "select * FROM pelanggan where  
      idplgn='"+idpel.getText()+"';java.sql.Statement stat = conn.createStatement();  
545. ResultSet hasil = stat.executeQuery(sql);  
546. if(hasil.next()) {  
      String qtys=beratbar.getText();  
      String totals=tarifbar.getText();  
      double qtysd=Double.parseDouble(qtys);  
      double totalsd=Double.parseDouble(totals);  
      NumberFormat nf3 = NumberFormat.getInstance(new Locale("da", "DK"));  
      String val3 = nf3.format((qtysd * totalsd));  
      subtotal.setText(val3);  
547. }  
548. } catch (SQLException e) {  
549. }  
550. }  
  
551. private void diskonActionPerformed(java.awt.event.ActionEvent evt) {  
552. // TODO add your handling code here:  
553. }  
  
554. private void namakarActionPerformed(java.awt.event.ActionEvent evt) {  
555. // TODO add your handling code here:  
556. }  
  
557. private void btncaripelanggan1ActionPerformed(java.awt.event.ActionEvent evt) {
```

```
558. popupkaryawan pop = new popupkaryawan();
559. pop.FN = this;
560. pop.setVisible(true);
561. pop.setResizable(false);
562. }

563. private void idkarActionPerformed(java.awt.event.ActionEvent evt) {
564. // TODO add your handling code here:
565. }

566. /**
   * @param args the command line arguments
567. */
568. public static void main(String args[]) {
569. /* Set the Nimbus look and feel */
570. //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
571. /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and
   feel.
   For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
572. */
573. try {
574. for (javax.swing.UIManager.LookAndFeelInfo info :
   javax.swing.UIManager.getInstalledLookAndFeels()) {
   if ("Nimbus".equals(info.getName())) {
   javax.swing.UIManager.setLookAndFeel(info.getClassName());
   break;
   }
575. }
576. } catch (ClassNotFoundException | InstantiationException | IllegalAccessException |
   javax.swing.UnsupportedLookAndFeelException ex) {
577. java.util.logging.Logger.getLogger(Transaksi.class.getName()).log(java.util.logging.Level
   .SEVERE, null, ex);
578. }
579. //</editor-fold>

580. //</editor-fold>

581. /* Create and display the form */
582. java.awt.EventQueue.invokeLater(() -> {
583. new Transaksi().setVisible(true);
584. });
585. }

586. // Variables declaration - do not modify
587. private javax.swing.JTextField beratbar;
588. private javax.swing.JButton btTambah;
589. private javax.swing.JButton btnbatal;
590. private javax.swing.JButton btncaribarang;
591. private javax.swing.JButton btncaripelanggan;
592. private javax.swing.JButton btncaripelanggan1;
593. private javax.swing.JButton btncetak;
594. private javax.swing.JButton btnkeluar;
595. private javax.swing.JTextField diskon;
596. private javax.swing.JTextField idjen;
597. private javax.swing.JTextField idkar;
```

```
598. private javax.swing.JTextField idpel;
599. private javax.swing.JLabel jLabel1;
600. private javax.swing.JLabel jLabel10;
601. private javax.swing.JLabel jLabel11;
602. private javax.swing.JLabel jLabel12;
603. private javax.swing.JLabel jLabel13;
604. private javax.swing.JLabel jLabel14;
605. private javax.swing.JLabel jLabel15;
606. private javax.swing.JLabel jLabel16;
607. private javax.swing.JLabel jLabel17;
608. private javax.swing.JLabel jLabel18;
609. private javax.swing.JLabel jLabel19;
610. private javax.swing.JLabel jLabel2;
611. private javax.swing.JLabel jLabel5;
612. private javax.swing.JLabel jLabel6;
613. private javax.swing.JLabel jLabel7;
614. private javax.swing.JLabel jLabel8;
615. private javax.swing.JLabel jLabel9;
616. private javax.swing.JPanel jPanel1;
617. private javax.swing.JScrollPane jScrollPane1;
618. private javax.swing.JTable jTable1;
619. private javax.swing.JTextField namabar;
620. private javax.swing.JTextField namakar;
621. private javax.swing.JTextField namapel;
622. private javax.swing.JTextField notrans;
623. private javax.swing.JTextField subtot;
624. private javax.swing.JTextField tarifbar;
625. private javax.swing.JTextField tgltrans;
626. private javax.swing.JTextField txtbayar;
627. private javax.swing.JTextField txtkembalian;
628. private javax.swing.JTextField txttotal;
629. // End of variables declaration
630. }
```

#### G. PengeluaranBarang.java

```
1. package Form;
2. import Koneksi.koneksi;
3. import java.sql.Connection;
4. import java.sql.DriverManager;
5. import java.sql.PreparedStatement;
6. import java.sql.ResultSet;
7. import java.sql.SQLException;
8. import java.sql.Statement;
9. import javax.swing.JOptionPane;
10. import javax.swing.table.DefaultTableModel;

11. /**
12. *
13. * @author USER
14. public final class PengeluaranBarang extends javax.swing.JFrame {
15. private final Connection conn = new koneksi().connect();
16. private DefaultTableModel tabmode;
17. String id;
18. /**
```

```

    Creates new form PengeluaranBarang
19. */
20. public PengeluaranBarang() {
21.     initComponents();
22.     datatable();
23.     txtcari.setText("");
24. }
25. protected void datatable(){
26.     Object[] Baris = {"No Transaksi","Tanggal","ID Pelanggan","Nama
Pelanggan","Keterangan"};
27.     tabmode = new DefaultTableModel(null,Baris);
28.     String cariitem=txtcari.getText();

29.     try {
30.         String sql = "SELECT * FROM transaksi where keterangan='Belum Diambil' GROUP
BY no_transaksi ";
31.         Statement stat = conn.createStatement();
32.         ResultSet hasil = stat.executeQuery(sql);
33.         while (hasil.next()){
34.             tabmode.addRow(new Object[]{
hasil.getString(1),
hasil.getString(2),
hasil.getString(3),
hasil.getString(4),
hasil.getString(13)

35. });
36.         jTable1.setModel(tabmode);
37.     } catch (SQLException e){
JOptionPane.showMessageDialog(null, "data gagal dipanggil"+e);
38.     }
39. }
40. @SuppressWarnings("unchecked")
41. // <editor-fold defaultstate="collapsed" desc="Generated Code">
42. private void initComponents() {

43.     jPanel1 = new javax.swing.JPanel();
44.     txtcari = new javax.swing.JTextField();
45.     btncari = new javax.swing.JButton();
46.     jScrollPane1 = new javax.swing.JScrollPane();
47.     jTable1 = new javax.swing.JTable();
48.     btndiambil = new javax.swing.JButton();
49.     btndiambil1 = new javax.swing.JButton();

50.     setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

51.     jPanel1.setBackground(new java.awt.Color(204, 204, 255));

52.     btncari.setText("Cari");
53.     btncari.addActionListener(new java.awt.event.ActionListener() {
54.         public void actionPerformed(java.awt.event.ActionEvent evt) {
55.             btncariActionPerformed(evt);
56.         }
56.     });

```



```

        .addGap(328, 328, 328)
        .addComponent(btndiambil)
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
85. );
86. jPanel1Layout.setVerticalGroup(
87. jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
88. .addGroup(jPanel1Layout.createSequentialGroup()
     .addContainerGap()
89. .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
     .addComponent(txtcari, javax.swing.GroupLayout.PREFERRED_SIZE,
     javax.swing.GroupLayout.DEFAULT_SIZE,
     javax.swing.GroupLayout.PREFERRED_SIZE)
     .addComponent(btncari)
     .addComponent(btndiambil1))
     .addGap(18, 18, 18)
     .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 154,
     javax.swing.GroupLayout.PREFERRED_SIZE)
     .addGap(18, 18, 18)
     .addComponent(btndiambil)
     .addContainerGap(26, Short.MAX_VALUE))
90. );
91. javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
92. getContentPane().setLayout(layout);
93. layout.setHorizontalGroup(
94. layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
95. .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE,
     javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
96. );
97. layout.setVerticalGroup(
98. layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
99. .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE,
     javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
100. );

101. pack();
102. setLocationRelativeTo(null);
103. }// </editor-fold>

104. private void btncariActionPerformed(java.awt.event.ActionEvent evt) {
105. Object[] Baris = {"No Transaksi","Tanggal","ID Pelanggan","Nama
     Pelanggan","Keterangan"};
106. tabmode = new DefaultTableModel(null,Baris);
107. String cariitem=txtcari.getText();

108. try {
109. String sql = "SELECT * FROM transaksi where keterangan='Belum Diambil' and
     no_transaksi='"+cariitem+"' GROUP BY no_transaksi ";
110. Statement stat = conn.createStatement();
111. ResultSet hasil = stat.executeQuery(sql);
112. while (hasil.next()){
     tabmode.addRow(new Object[]{{
     hasil.getString(1),
     hasil.getString(2),

```

```

    hasil.getString(3),
    hasil.getString(4),
    hasil.getString(13)

113. });
114. }
115. jTable1.setModel(tabmode);
116. } catch (SQLException e){
    JOptionPane.showMessageDialog(null, "data gagal dipanggil"+e);
117. }
118. }

119. private void btndiambilActionPerformed(java.awt.event.ActionEvent evt) {
120. int ok = JOptionPane.showConfirmDialog(null,"Barang ingin diambil ?","konfirmasi
    dialog",JOptionPane.YES_NO_OPTION);
121. if (ok==0){
    String sql = "update transaksi set keterangan='Sudah Diambil' where no_transaksi =
    "+id+"";
    try{
        PreparedStatement stat = conn.prepareStatement(sql);
        stat.executeUpdate();
        JOptionPane.showMessageDialog(null, "Barang Sudah Diambil");
        txtcari.setText("");
        txtcari.requestFocus();

        new Barang().setEnabled(true);
        new Barang().setVisible(false);
        this.dispose();
        new Barang().setVisible(true);

    }
    catch (SQLException e){
        JOptionPane.showMessageDialog(null, "Barang gagal diambil"+e);
    }
}
122. }
123. datatable();
124. id=null;

125. }

126. private void jTable1MouseClicked(java.awt.event.MouseEvent evt) {
127. int bar = jTable1.getSelectedRow();
128. String a = tabmode.getValueAt (bar, 0).toString();

129. id=a;
130. }

131. private void btndiambil1ActionPerformed(java.awt.event.ActionEvent evt) {
132. new Barang().setEnabled(true);
    new Barang().setVisible(false);
    this.dispose();
    new Barang().setVisible(true);

133. }

```

```

134. /**
     @param args the command line arguments
135. */
136. public static void main(String args[]) {
137. /* Set the Nimbus look and feel */
138. //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
139. /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and
   feel.
   For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
140. */
141. try {
142. for (javax.swing.UIManager.LookAndFeelInfo info :
   javax.swing.UIManager.getInstalledLookAndFeels()) {
   if ("Nimbus".equals(info.getName())) {
   javax.swing.UIManager.setLookAndFeel(info.getClassName());
   break;
   }
143. }
144. } catch (ClassNotFoundException | InstantiationException | IllegalAccessException |
   javax.swing.UnsupportedLookAndFeelException ex) {
145. java.util.logging.Logger.getLogger(PengeluaranBarang.class.getName()).log(java.util.log
   ging.Level.SEVERE, null, ex);
146. }
147. //</editor-fold>

148. //</editor-fold>

149. /* Create and display the form */
150. java.awt.EventQueue.invokeLater(() -> {
151. new PengeluaranBarang().setVisible(true);
152. });
153. }

154. // Variables declaration - do not modify
155. private javax.swing.JButton btncari;
156. private javax.swing.JButton btndambil;
157. private javax.swing.JButton btndambil1;
158. private javax.swing.JPanel jPanel1;
159. private javax.swing.JScrollPane jScrollPane1;
160. private javax.swing.JTable jTable1;
161. private javax.swing.JTextField txtcari;
162. // End of variables declaration
163. }

```

## H. Barang.java

```

1.  package Form;
2.  import Koneksi.koneksi;
3.  import java.sql.Connection;
4.  import java.sql.DriverManager;
5.  import java.sql.PreparedStatement;
6.  import java.sql.ResultSet;
7.  import java.sql.SQLException;
8.  import java.sql.Statement;
9.  import java.text.SimpleDateFormat;
10. import java.util.Date;

```

```

11. import javax.swing.JOptionPane;
12. import javax.swing.table.DefaultTableModel;

13. /**
14. *
15. * @author USER
16. */
17. public class Barang extends javax.swing.JFrame {
18.     private Connection conn = new koneksi().connect();
19.     private DefaultTableModel tabmode;
20.     String id;
21.     /**
22.      Creates new form PengeluaranBarang
23.     */
24.     public Barang() {
25.         initComponents();
26.         datatable();
27.         txtcari.setText("");
28.     }
29.     protected void datatable(){
30.         Object[] Baris = {"No Transaksi","Tanggal","ID Pelanggan","Nama
31.         Pelanggan","Keterangan"};
32.         tabmode = new DefaultTableModel(null,Baris);
33.         try {
34.             String sql = "SELECT * FROM transaksi GROUP BY no_transaksi ";
35.             Statement stat = conn.createStatement();
36.             ResultSet hasil = stat.executeQuery(sql);
37.             while (hasil.next()){
38.                 tabmode.addRow(new Object[]{
39.                     hasil.getString(1),
40.                     hasil.getString(2),
41.                     hasil.getString(3),
42.                     hasil.getString(4),
43.                     hasil.getString(13)
44.                 });
45.             }
46.             jTable1.setModel(tabmode);
47.         } catch (SQLException e){
48.             JOptionPane.showMessageDialog(null, "data gagal dipanggil"+e);
49.         }
50.     }

51.     /**
52.      @SuppressWarnings("unchecked")
53.      // <editor-fold defaultstate="collapsed" desc="Generated Code">
54.      private void initComponents() {

55.         jPanel1 = new javax.swing.JPanel();
56.         txtcari = new javax.swing.JTextField();
57.         btncari = new javax.swing.JButton();
58.         jScrollPane1 = new javax.swing.JScrollPane();
59.         jTable1 = new javax.swing.JTable();
60.         btndiambil = new javax.swing.JButton();
61.         btndiambil1 = new javax.swing.JButton();

```



```

.addComponent(txtcari, javax.swing.GroupLayout.PREFERRED_SIZE, 121,
javax.swing.GroupLayout.PREFERRED_SIZE)
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
.addComponent(btncari)
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
.addComponent(btndiambil1, javax.swing.GroupLayout.PREFERRED_SIZE, 131,
javax.swing.GroupLayout.PREFERRED_SIZE))
.addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel1Layout.createSequentialGroup()
.addGap(0, 0, Short.MAX_VALUE)
.addComponent(btndiambil, javax.swing.GroupLayout.PREFERRED_SIZE, 131,
javax.swing.GroupLayout.PREFERRED_SIZE)))
.addContainerGap())
85. );
86. jPanel1Layout.setVerticalGroup(
87. jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
88. .addGroup(jPanel1Layout.createSequentialGroup()
89. .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
ADING)
.addGroup(jPanel1Layout.createSequentialGroup()
.addGap(86, 86, 86)
90. .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
SELINE)
.addComponent(txtcari, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
.addComponent(btncari)))
.addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel1Layout.createSequentialGroup()
.addContainerGap()
.addComponent(btndiambil1, javax.swing.GroupLayout.PREFERRED_SIZE, 46,
javax.swing.GroupLayout.PREFERRED_SIZE)))
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
.addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 297,
javax.swing.GroupLayout.PREFERRED_SIZE)
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
.addComponent(btndiambil, javax.swing.GroupLayout.DEFAULT_SIZE, 39,
Short.MAX_VALUE)
.addGap(23, 23, 23))
91. );
92. javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
93. getContentPane().setLayout(layout);
94. layout.setHorizontalGroup(
95. layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
96. .addGroup(layout.createSequentialGroup()
.addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
.addGap(0, 0, Short.MAX_VALUE))
97. );
98. layout.setVerticalGroup(
99. layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
100. .addGroup(layout.createSequentialGroup())

```

```

        .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
        javax.swing.GroupLayout.DEFAULT_SIZE,
        javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(0, 0, Short.MAX_VALUE))
101. );
102. pack();
103. setLocationRelativeTo(null);
104. }// </editor-fold>

105. private void btncariActionPerformed(java.awt.event.ActionEvent evt) {
106. Object[] Baris = {"No Transaksi","Tanggal","ID Pelanggan","Nama
Pelanggan","Keterangan"};
107. tabmode = new DefaultTableModel(null,Baris);
108. String cariitem=txtcari.getText();

109. try {
110. String sql = "SELECT * FROM transaksi where no_transaksi='"+cariitem+"' GROUP BY
no_transaksi ";
111. Statement stat = conn.createStatement();
112. ResultSet hasil = stat.executeQuery(sql);
113. while (hasil.next()){
114. tabmode.addRow(new Object[]{
hasil.getString(1),
hasil.getString(2),
hasil.getString(3),
hasil.getString(4),
hasil.getString(13)

115. });
116. jTable1.setModel(tabmode);
117. } catch (SQLException e){
JOptionPane.showMessageDialog(null, "data gagal dipanggil"+e);
118. }
119. }

120. private void btndambilActionPerformed(java.awt.event.ActionEvent evt) {
121. this.setVisible(false);
122. new PengeluaranBarang().setVisible(true);

123. }

124. private void jTable1MouseClicked(java.awt.event.MouseEvent evt) {
125. int bar = jTable1.getSelectedRow();
126. String a = tabmode.getValueAt (bar, 0).toString();

127. id=a;
128. }

129. private void btndambil1ActionPerformed(java.awt.event.ActionEvent evt) {
130. this.setVisible(false);
131. new menu().setVisible(true);
132. }

```

```

133. /**
134.  * @param args the command line arguments
135. */
136. public static void main(String args[]) {
137. /* Set the Nimbus look and feel */
138. //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
139. /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and
140. feel.
141. For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
142. */
143. try {
144. for (javax.swing.UIManager.LookAndFeelInfo info :
145. javax.swing.UIManager.getInstalledLookAndFeels()) {
146. if ("Nimbus".equals(info.getName())) {
147. javax.swing.UIManager.setLookAndFeel(info.getClassName());
148. break;
149. }
150. }
151. } catch (ClassNotFoundException | InstantiationException | IllegalAccessException |
152. javax.swing.UnsupportedLookAndFeelException ex) {
153. java.util.logging.Logger.getLogger(Barang.class.getName()).log(java.util.logging.Level.S
154. EVERE, null, ex);
155. }
156. //</editor-fold>
157. //</editor-fold>
158. //</editor-fold>
159. //</editor-fold>
160. //</editor-fold>
161. //</editor-fold>
162. //</editor-fold>
163. //</editor-fold>
164. //</editor-fold>
165. //</editor-fold>
166. //</editor-fold>
167. // End of variables declaration
168. }

```

### I. JenisBarang.java

```
1. package Form;
2. import java.sql.*;
3. import javax.swing.JOptionPane;
4. import javax.swing.table.DefaultTableModel;
5. import java.awt.event.KeyEvent;
6. import Koneksi.koneksi;
7. import java.awt.Color;
8. /**
9. *
10. * @author USER
11. */
12. public final class JenisBarang extends javax.swing.JFrame {
13. private final Connection conn = new koneksi().connect();
14. private DefaultTableModel tabmode;
15. /**
16. Creates new form JenisBarang
17. */
18. public JenisBarang() {
19. initComponents();
20. kosong();
21. aktif();
22. datatable();
23. }
24. protected void aktif(){
25. txtidjenis.requestFocus();
26. }

27. protected void kosong(){
28. txtidjenis.setText("");
29. txtnmbrg.setText("");
30. txttarif.setText("");
31. txtbrt.setText("");
32. txtcari.setText("");
33. }

34. protected void datatable(){
35. Object[] Baris = { "Id Jenis","Nama Barang","Tarif","Berat Barang/Qty"};
36. tabmode = new DefaultTableModel(null,Baris);
37. String cariitem=txtcari.getText();

38. try {
39. String sql = "SELECT * FROM jenisbarang where idjenis like '%" +cariitem+"%' or
40. nmbrg like '%"+cariitem+"%' order by idjenis asc";
41. Statement stat = conn.createStatement();
42. ResultSet hasil = stat.executeQuery(sql);
43. while (hasil.next()){
44. tabmode.addRow(new Object[]{
45. hasil.getString(1),
46. hasil.getString(2),
47. hasil.getString(3),
48. hasil.getString(4)
49. })
```

```

43.    });
44.    }
45.    tbjenisbrg.setModel(tabmode);
46.    } catch (SQLException e){
47.        JOptionPane.showMessageDialog(null, "data gagal dipanggil"+e);
48.    }
49.    }
50.    */
51.    @SuppressWarnings("unchecked")
52.    // <editor-fold defaultstate="collapsed" desc="Generated Code">
53.    private void initComponents() {

54.        jPanel2 = new javax.swing.JPanel();
55.        jLabel2 = new javax.swing.JLabel();
56.        txtidjenis = new javax.swing.JTextField();
57.        jLabel3 = new javax.swing.JLabel();
58.        txtnbrg = new javax.swing.JTextField();
59.        txttarif = new javax.swing.JTextField();
60.        jLabel4 = new javax.swing.JLabel();
61.        txtbrt = new javax.swing.JTextField();
62.        jLabel6 = new javax.swing.JLabel();
63.        jLabel8 = new javax.swing.JLabel();
64.        jLabel7 = new javax.swing.JLabel();
65.        jPanel1 = new javax.swing.JPanel();
66.        txtcari = new javax.swing.JTextField();
67.        bcari = new javax.swing.JButton();
68.        jScrollPane2 = new javax.swing.JScrollPane();
69.        tbjenisbrg = new javax.swing.JTable();
70.        bsimpan = new javax.swing.JButton();
71.        bbatal = new javax.swing.JButton();
72.        jButton1 = new javax.swing.JButton();
73.        bhapus = new javax.swing.JButton();
74.        bubah = new javax.swing.JButton();
75.        jLabel1 = new javax.swing.JLabel();
76.        jLabel9 = new javax.swing.JLabel();

77.        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
78.        getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

79.        jPanel2.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

80.        jLabel2.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N
81.        jLabel2.setForeground(new java.awt.Color(255, 255, 255));
82.        jLabel2.setText("ID Jenis");
83.        jPanel2.add(jLabel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(10, 10, -1, -1));

84.        txtidjenis.addActionListener(new java.awt.event.ActionListener() {
85.            public void actionPerformed(java.awt.event.ActionEvent evt) {
86.                txtidjenisActionPerformed(evt);
87.            }
88.        });

```

```

88. jPanel2.add(txtidjenis, new org.netbeans.lib.awtextra.AbsoluteConstraints(130, 10, 140,
30));

89. jLabel3.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N
90. jLabel3.setForeground(new java.awt.Color(255, 255, 255));
91. jLabel3.setText("Nama Barang");
92. jPanel2.add(jLabel3, new org.netbeans.lib.awtextra.AbsoluteConstraints(10, 50, -1, -1));

93. txtnbrg.addActionListener(new java.awt.event.ActionListener() {
94.     public void actionPerformed(java.awt.event.ActionEvent evt) {
95.         txtnbrgActionPerformed(evt);
96.     }
97. });
98. jPanel2.add(txtnbrg, new org.netbeans.lib.awtextra.AbsoluteConstraints(130, 50, 140,
30));

99. txttarif.addActionListener(new java.awt.event.ActionListener() {
100.    public void actionPerformed(java.awt.event.ActionEvent evt) {
101.        txttarifActionPerformed(evt);
102.    }
103. });
104. jPanel2.add(txttarif, new org.netbeans.lib.awtextra.AbsoluteConstraints(130, 100, 140,
30));

105. jLabel4.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N
106. jLabel4.setForeground(new java.awt.Color(255, 255, 255));
107. jLabel4.setText("Tarif");
108. jPanel2.add(jLabel4, new org.netbeans.lib.awtextra.AbsoluteConstraints(10, 100, -1, -1));
109. jPanel2.add(txtbrt, new org.netbeans.lib.awtextra.AbsoluteConstraints(130, 140, 140,
30));

110. jLabel6.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N
111. jLabel6.setForeground(new java.awt.Color(255, 255, 255));
112. jLabel6.setText("Berat Barang");
113. jPanel2.add(jLabel6, new org.netbeans.lib.awtextra.AbsoluteConstraints(10, 150, -1, -1));

114. jLabel8.setFont(new java.awt.Font("Times New Roman", 0, 14)); // NOI18N
115. jLabel8.setForeground(new java.awt.Color(255, 255, 255));
116. jLabel8.setText("/");
117. jPanel2.add(jLabel8, new org.netbeans.lib.awtextra.AbsoluteConstraints(280, 150, -1, -1));

118. jLabel7.setFont(new java.awt.Font("Times New Roman", 0, 14)); // NOI18N
119. jLabel7.setForeground(new java.awt.Color(255, 255, 255));
120. jLabel7.setText("KG");
121. jPanel2.add(jLabel7, new org.netbeans.lib.awtextra.AbsoluteConstraints(290, 150, -1, -1));

122. getContentPane().add(jPanel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(30, 70,
340, 190));

123. jPanel1.setBorder(javax.swing.BorderFactory.createTitledBorder(javax.swing.BorderFactory.createEmptyBorder(1, 1, 1, 1), "Cari Jenis Barang",
javax.swing.border.TitledBorder.DEFAULT_JUSTIFICATION,

```

```

        javax.swing.border.TitledBorder.DEFAULT_POSITION, new java.awt.Font("Tahoma",
0, 13), new java.awt.Color(255, 255, 255))); // NOI18N
122. jPanel1.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

123. txtcari.addKeyListener(new java.awt.event.KeyAdapter() {
124.     public void keyPressed(java.awt.event.KeyEvent evt) {
125.         txtcariKeyPressed(evt);
126.     });
127. jPanel1.add(txtcari, new org.netbeans.lib.awtextra.AbsoluteConstraints(20, 20, 190, -1));

128. bcari.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N
129. bcari.setText("Cari");
130. bcari.addActionListener(new java.awt.event.ActionListener() {
131.     public void actionPerformed(java.awt.event.ActionEvent evt) {
132.         bcariActionPerformed(evt);
133.     });
134. jPanel1.add(bcari, new org.netbeans.lib.awtextra.AbsoluteConstraints(240, 20, -1, -1));

135. getContentPane().add(jPanel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(30,
270, 340, 60));

136. tbjenisbrg.setModel(new javax.swing.table.DefaultTableModel(
137.     new Object [][] {
138.         {null, null, null, null},
139.         {null, null, null, null},
140.         {null, null, null, null}
141.     },
142.     new String [] {
143.         "Title 1", "Title 2", "Title 3", "Title 4"
144.     }
145. ));
146. tbjenisbrg.addMouseListener(new java.awt.event.MouseAdapter() {
147.     public void mouseClicked(java.awt.event.MouseEvent evt) {
148.         tbjenisbrgMouseClicked(evt);
149.     }
150. });
151. jScrollPane2.setViewportView(tbjenisbrg);

152. getContentPane().add(jScrollPane2, new
org.netbeans.lib.awtextra.AbsoluteConstraints(30, 340, 650, 180));

153. bsimpan.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N
154. bsimpan.setText("Simpan");
155. bsimpan.addActionListener(new java.awt.event.ActionListener() {
156.     public void actionPerformed(java.awt.event.ActionEvent evt) {
157.         bsimpanActionPerformed(evt);
158.     }
159. });
160. getContentPane().add(bsimpan, new org.netbeans.lib.awtextra.AbsoluteConstraints(440,
80, -1, -1));

161. bbatal.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N

```

```

156. bbatal.setText("Batal");
157. bbatal.addActionListener(new java.awt.event.ActionListener() {
158.     public void actionPerformed(java.awt.event.ActionEvent evt) {
159.         bbatalActionPerformed(evt);
160.     });
161.     getContentPane().add(bbatal, new org.netbeans.lib.awtextra.AbsoluteConstraints(440,
162. 120, 75, -1));

163. jButton1.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N
164. jButton1.setText("Keluar");
165. jButton1.addActionListener(new java.awt.event.ActionListener() {
166.     public void actionPerformed(java.awt.event.ActionEvent evt) {
167.         jButton1ActionPerformed(evt);
168.     });
169.     getContentPane().add(jButton1, new org.netbeans.lib.awtextra.AbsoluteConstraints(560,
170. 80, -1, -1));

171. bhapus.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N
172. bhapus.setText("Hapus");
173. bhapus.addActionListener(new java.awt.event.ActionListener() {
174.     public void actionPerformed(java.awt.event.ActionEvent evt) {
175.         bhapusActionPerformed(evt);
176.     });
177.     getContentPane().add(bhapus, new org.netbeans.lib.awtextra.AbsoluteConstraints(440,
178. 190, 75, -1));

179. bubah.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N
180. bubah.setText("Ubah");
181. bubah.addActionListener(new java.awt.event.ActionListener() {
182.     public void actionPerformed(java.awt.event.ActionEvent evt) {
183.         bubahActionPerformed(evt);
184.     });
185.     getContentPane().add(bubah, new org.netbeans.lib.awtextra.AbsoluteConstraints(440,
186. 160, 75, -1));

187. jLabel1.setFont(new java.awt.Font("Times New Roman", 1, 24)); // NOI18N
188. jLabel1.setText("JENIS BARANG");
189. getContentPane().add(jLabel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(260,
190. 20, -1, -1));

191. jLabel9.setIcon(new javax.swing.ImageIcon(getClass().getResource("/img/bg4.jpg"))); // NOI18N
192. getContentPane().add(jLabel9, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0,
700, 550));

193. setSize(new java.awt.Dimension(719, 578));
194. setLocationRelativeTo(null);
195. } // </editor-fold>

196. private void txtidjenisActionPerformed(java.awt.event.ActionEvent evt) {
197.     // TODO add your handling code here:

```

```

193.    }

194.    private void txtnbrgActionPerformed(java.awt.event.ActionEvent evt) {
195.        // TODO add your handling code here:
196.    }

197.    private void txttarifActionPerformed(java.awt.event.ActionEvent evt) {
198.        // TODO add your handling code here:
199.    }

200.    private void bhapusActionPerformed(java.awt.event.ActionEvent evt) {
201.        int ok = JOptionPane.showConfirmDialog(null,"hapus","konfirmasi
202.        dialog",JOptionPane.YES_NO_OPTION);
203.        if (ok==0){
204.            String sql = "delete from jenisbarang where idjenis = '"+txtidjenis.getText()+"'";
205.            try{
206.                PreparedStatement stat = conn.prepareStatement(sql);
207.                stat.executeUpdate();
208.                JOptionPane.showMessageDialog(null, "data berhasil dihapus");
209.                kosong();
210.                datatable();
211.            }
212.            catch (SQLException e){
213.                JOptionPane.showMessageDialog(null, "data gagal dihapus"+e);
214.            }
215.           datatable();
216.        }
217.    }

205.    private void bbatalActionPerformed(java.awt.event.ActionEvent evt) {
206.        kosong();
207.        datatable();
208.    }

209.    private void bubahActionPerformed(java.awt.event.ActionEvent evt) {
210.        try{
211.            String sql = "update jenisbarang set idjenis=?,nmbrg=?,tarif=?,brtbrg=? where
212.            idjenis='"+txtidjenis.getText()+"'";
213.            PreparedStatement stat = conn.prepareStatement(sql);
214.            stat.setString(1, txtidjenis.getText());
215.            stat.setString(2, txtnbrg.getText());
216.            stat.setString(3, txttarif.getText());
217.            stat.setString(4, txtbrt.getText());

218.            stat.executeUpdate();
219.            JOptionPane.showMessageDialog(null, "Data Berhasil Diubah");
220.            kosong();
221.            txtidjenis.requestFocus();
222.        }
223.        catch (SQLException e){
224.            JOptionPane.showMessageDialog(null, "Data Gagal Diubah"+e);
225.        }
226.        datatable();
227.    }

```

```

218.    private void bsimpanActionPerformed(java.awt.event.ActionEvent evt) {
219.        String sql = "insert into jenisbarang values (?,?,?,?,?)";
220.        try{
221.            PreparedStatement stat = conn.prepareStatement(sql);
222.            stat.setString(1, txtidjenis.getText());
223.            stat.setString(2, txtnmbrg.getText());
224.            stat.setString(3, txttarif.getText());
225.            stat.setString(4, txtbrt.getText());
226.
227.            stat.executeUpdate();
228.            JOptionPane.showMessageDialog(null, "Data Berhasil Disimpan");
229.            kosong();
230.            txtidjenis.requestFocus();
231.        }
232.        catch (SQLException e){
233.            JOptionPane.showMessageDialog(null, "Data Gagal Disimpan");
234.        }
235.        datatable();
236.    }
237.
238.    private void txtcariKeyPressed(java.awt.event.KeyEvent evt) {
239.        datatable();
240.
241.    private void tbjenisbrgMouseClicked(java.awt.event.MouseEvent evt) {
242.        int bar = tbjenisbrg.getSelectedRow();
243.        String a = tabmode.getValueAt(bar, 0).toString();
244.        String b = tabmode.getValueAt(bar, 1).toString();
245.        String c = tabmode.getValueAt(bar, 2).toString();
246.        String d = tabmode.getValueAt(bar, 3).toString();
247.
248.        txtidjenis.setText(a);
249.        txtnmbrg.setText(b);
250.        txttarif.setText(c);
251.        txtbrt.setText(d);
252.
253.    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
254.        this.setVisible(false);
255.        new menu().setVisible(true);
256.    }
257.
258.    /**
259.     * @param args the command line arguments
260.     */
261.    public static void main(String args[]) {
262.        /* Set the Nimbus look and feel */
263.        /*<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

```

```

260.    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and
261.       feel.
262.       For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
263. */
264.    try {
265.        for (javax.swing.UIManager.LookAndFeelInfo info :
266.             javax.swing.UIManager.getInstalledLookAndFeels()) {
267.            if ("Nimbus".equals(info.getName())) {
268.                javax.swing.UIManager.setLookAndFeel(info.getClassName());
269.                break;
270.            }
271.        } catch (ClassNotFoundException | InstantiationException | IllegalAccessException |
272.               javax.swing.UnsupportedLookAndFeelException ex) {
273.            java.util.logging.Logger.getLogger(JenisBarang.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
274.        }
275.    } //
```

```

276.    // Create and display the form
277.    java.awt.EventQueue.invokeLater(() -> {
278.        new JenisBarang().setVisible(true);
279.    });
280. }

281. // Variables declaration - do not modify
282. private javax.swing.JButton bbatal;
283. private javax.swing.JButton bcari;
284. private javax.swing.JButton bhapus;
285. private javax.swing.JButton bsimpan;
286. private javax.swing.JButton bubah;
287. private javax.swing.JButton jButton1;
288. private javax.swing.JLabel jLabel1;
289. private javax.swing.JLabel jLabel2;
290. private javax.swing.JLabel jLabel3;
291. private javax.swing.JLabel jLabel4;
292. private javax.swing.JLabel jLabel6;
293. private javax.swing.JLabel jLabel7;
294. private javax.swing.JLabel jLabel8;
295. private javax.swing.JLabel jLabel9;
296. private javax.swing.JPanel jPanel1;
297. private javax.swing.JPanel jPanel2;
298. private javax.swing.JScrollPane jScrollPane2;
299. private javax.swing.JTable tbjenisbrg;
300. private javax.swing.JTextField txtbrt;
301. private javax.swing.JTextField txtcari;
302. private javax.swing.JTextField txtidjenis;
303. private javax.swing.JTextField txtnmbrg;
304. private javax.swing.JTextField txttarif;
305. // End of variables declaration
306. }
```

### **J. Popupjenisbarang.java**

```
1. package Form;
2. import Koneksi.koneksi;
3. import java.sql.Connection;
4. import java.sql.ResultSet;
5. import java.sql.SQLException;
6. import java.sql.Statement;
7. import javax.swing.JOptionPane;
8. import javax.swing.table.DefaultTableModel;

9. /**
10. *
11. * @author USER
12. */
13. public final class popupjenisbarang extends javax.swing.JFrame {
14. private final Connection conn = new koneksi().connect();
15. private DefaultTableModel tabmode;
16. public Transaksi FN = null;

17. /**
18. Creates new form popupjenisbarang
19. */
20. public popupjenisbarang() {
21. initComponents();
22. datatable();
23. }
24. protected void datatable(){
25. Object[] Baris = { "Id Jenis","Nama Barang","Tarif","Berat Barang/Qty"};
26. tabmode = new DefaultTableModel(null,Baris);
27. String cariitem=txtcari.getText();

28. try {
29. String sql = "SELECT * FROM jenisbarang where idjenis like '%" +cariitem+"%' or
30. nmbrg like '%"+cariitem+"%' order by idjenis asc";
31. Statement stat = conn.createStatement();
32. ResultSet hasil = stat.executeQuery(sql);
33. while (hasil.next()){
34. tabmode.addRow(new Object[]{
35. hasil.getString(1),
36. hasil.getString(2),
37. hasil.getString(3),
38. hasil.getString(4)
39. });
30. }
31. tblbrg.setModel(tabmode);
32. } catch (SQLException e){
33. JOptionPane.showMessageDialog(null, "data gagal dipanggil"+e);
34. }
35. }

36. }

37. /**
38. This method is called from within the constructor to initialize the form.
39. WARNING: Do NOT modify this code. The content of this method is always
40. regenerated by the Form Editor.
```



```

.addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT_SIZE, 601,
Short.MAX_VALUE)
.addGroup(jPanel1Layout.createSequentialGroup())
.addComponent(txtcari, javax.swing.GroupLayout.PREFERRED_SIZE, 128,
javax.swing.GroupLayout.PREFERRED_SIZE)
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
.addComponent(jButton1)
.addGap(72, 72, 72)
.addComponent(jLabel1)
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
.addComponent(btndiamobil1)))
.addContainerGap()
);
jPanel1Layout.setVerticalGroup(
jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(jPanel1Layout.createSequentialGroup()
.a.
.addContainerGap()
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
.addComponent(txtcari, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
.addComponent(jButton1)
.addComponent(jLabel1)
.addComponent(btndiamobil1))
.addGap(18, 18, 18)
.addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 132,
javax.swing.GroupLayout.PREFERRED_SIZE)
.addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
);
81.
82. javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
83. getContentPane().setLayout(layout);
84. layout.setHorizontalGroup(
85. layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
86. .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
87. );
88. layout.setVerticalGroup(
89. layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
90. .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
91. );
92. setSize(new java.awt.Dimension(643, 248));
93. setLocationRelativeTo(null);
94. } // </editor-fold>

95. private void tblbrgMouseClicked(java.awt.event.MouseEvent evt) {
96. // TODO add your handling code here:
97. int tabelBarang = tblbrg.getSelectedRow();
98. FN.id_barang = tblbrg.getValueAt(tabelBarang, 0).toString();
99. FN.nama_barang = tblbrg.getValueAt(tabelBarang, 1).toString();
100. FN.tarif = tblbrg.getValueAt(tabelBarang, 2).toString();

```

```

101.    FN.itemTerpilih();
102.    this.dispose();
103. }
104. private void btndiambil1ActionPerformed(java.awt.event.ActionEvent evt) {
105.     this.setVisible(false);
106. }
107. /**
108.  * @param args the command line arguments
109. */
110. public static void main(String args[]) {
111.     /* Set the Nimbus look and feel */
112.     //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
113.     /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and
114.      feel.
115.      For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
116.     */
117.     try {
118.         for (javax.swing.UIManager.LookAndFeelInfo info :
119.             javax.swing.UIManager.getInstalledLookAndFeels()) {
120.             if ("Nimbus".equals(info.getName())) {
121.                 javax.swing.UIManager.setLookAndFeel(info.getClassName());
122.                 break;
123.             }
124.         }
125.     } catch (ClassNotFoundException | InstantiationException | IllegalAccessException |
126.         javax.swing.UnsupportedLookAndFeelException ex) {
127.         java.util.logging.Logger.getLogger(popupjenisbarang.class.getName()).log(java.util.loggi
128.         ng.Level.SEVERE, null, ex);
129.     }
130.     //

```

### **K. Popuppelanggan.java**

```
1. package Form;
2. import java.sql.*;
3. import javax.swing.JOptionPane;
4. import javax.swing.table.DefaultTableModel;
5. import java.awt.event.KeyEvent;
6. import Koneksi.koneksi;
7. /**
8. *
9. * @author USER
10. */
11. public final class popuppelanggan extends javax.swing.JFrame {
12.     private final Connection conn = new koneksi().connect();
13.     private DefaultTableModel tabmode;
14.     public Transaksi FN = null;
15.     /**
16.      Creates new form popuppelanggan
17.     */
18.     public popuppelanggan() {
19.         initComponents();
20.         datatable();
21.     }
22.     protected void datatable(){
23.         Object[] Baris = {"ID Pelanggan","Nama Pelanggan"};
24.         tabmode = new DefaultTableModel (null, Baris);
25.         String cariitem=txtcari.getText();
26.         try{
27.             tabmode.getDataVector().removeAllElements();
28.             String sql = "SELECT * FROM pelanggan where idplgn like '%"+cariitem+"%' or
29.             nmplgn like '%"+cariitem+"%' order by idplgn asc";
30.             Statement stat = conn.createStatement();
31.             ResultSet hasil = stat.executeQuery(sql);
32.             while (hasil.next()){
33.                 tabmode.addRow(new Object[]{
34.                     hasil.getString(1),
35.                     hasil.getString(2)
36.                 });
37.             }
38.             tblplgn.setModel(tabmode);
39.         } catch (SQLException e){
40.             JOptionPane.showMessageDialog(null, "data gagal dipanggil" +e);
41.         }
42.         /**
43.          * This method is called from within the constructor to initialize the form.
44.          * WARNING: Do NOT modify this code. The content of this method is always
45.          * regenerated by the Form Editor.
46.          */
47.         @SuppressWarnings("unchecked")
48.         // <editor-fold defaultstate="collapsed" desc="Generated Code">
49.         private void initComponents() {
50.
51.             txtcari = new javax.swing.JTextField();
52.             bcari = new javax.swing.JButton();
53.
54.             setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
55.             setTitle("Pencarian Pelanggan");
56.             setIconImage(null);
57.             setUndecorated(true);
58.
59.             txtcari.addActionListener(new java.awt.event.ActionListener() {
60.                 public void actionPerformed(java.awt.event.ActionEvent evt) {
61.                     search();
62.                 }
63.             });
64.
65.             bcari.setText("Cari");
66.             bcari.addActionListener(new java.awt.event.ActionListener() {
67.                 public void actionPerformed(java.awt.event.ActionEvent evt) {
68.                     search();
69.                 }
70.             });
71.
72.             javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
73.             getContentPane().setLayout(layout);
74.             layout.setHorizontalGroup(
75.                 layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
76.                     .addGroup(layout.createSequentialGroup()
77.                         .addComponent(txtcari, javax.swing.GroupLayout.PREFERRED_SIZE, 150, javax.swing.GroupLayout.PREFERRED_SIZE)
78.                         .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
79.                         .addComponent(bcari, javax.swing.GroupLayout.PREFERRED_SIZE, 80, javax.swing.GroupLayout.PREFERRED_SIZE)
80.                         .addContainerGap())
81.             );
82.             layout.setVerticalGroup(
83.                 layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
84.                     .addGroup(layout.createSequentialGroup()
85.                         .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
86.                             .addComponent(txtcari, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
87.                             .addComponent(bcari, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
88.                         .addContainerGap())
89.             );
90.
91.             pack();
92.         }
93.
94.         private void search() {
95.             String cariitem=txtcari.getText();
96.             try{
97.                 tabmode.getDataVector().removeAllElements();
98.                 String sql = "SELECT * FROM pelanggan where idplgn like '%"+cariitem+"%' or
99.                 nmplgn like '%"+cariitem+"%' order by idplgn asc";
100.                 Statement stat = conn.createStatement();
101.                 ResultSet hasil = stat.executeQuery(sql);
102.                 while (hasil.next()){
103.                     tabmode.addRow(new Object[]{
104.                         hasil.getString(1),
105.                         hasil.getString(2)
106.                     });
107.                 }
108.                 tblplgn.setModel(tabmode);
109.             } catch (SQLException e){
110.                 JOptionPane.showMessageDialog(null, "data gagal dipanggil" +e);
111.             }
112.         }
113.     }
114.
```

```

44. jLabel1 = new javax.swing.JLabel();
45. jScrollPane1 = new javax.swing.JScrollPane();
46. tblplgn = new javax.swing.JTable();
47. btndiambil1 = new javax.swing.JButton();

48. setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
49. setBackground(new java.awt.Color(153, 153, 153));

50. txtcari.addKeyListener(new java.awt.event.KeyAdapter() {
51.     public void keyPressed(java.awt.event.KeyEvent evt) {
52.         txtcariKeyPressed(evt);
53.     }
54. });

55. bcari.setText("Cari");
56. bcari.addActionListener(new java.awt.event.ActionListener() {
57.     public void actionPerformed(java.awt.event.ActionEvent evt) {
58.         bcariActionPerformed(evt);
59.     }
60. });

61. jLabel1.setFont(new java.awt.Font("Times New Roman", 1, 18)); // NOI18N
62. jLabel1.setText("PopUp Pelanggan");

63. tblplgn.setModel(new javax.swing.table.DefaultTableModel(
64.     new Object [][] {
65.         {null, null, null, null},
66.         {null, null, null, null},
67.         {null, null, null, null},
68.         {null, null, null, null}
69.     },
70.     new String [] {
71.         "Title 1", "Title 2", "Title 3", "Title 4"
72.     }
73. ));
74. tblplgn.addMouseListener(new java.awt.event.MouseAdapter() {
75.     public void mouseClicked(java.awt.event.MouseEvent evt) {
76.         tblplgnMouseClicked(evt);
77.     }
78. });
79. jScrollPane1.setViewportView(tblplgn);

80. btndiambil1.setText("BATAL");
81. btndiambil1.addActionListener(new java.awt.event.ActionListener() {
82.     public void actionPerformed(java.awt.event.ActionEvent evt) {
83.         btndiambil1ActionPerformed(evt);
84.     }
85. });

86. javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
87. getContentPane().setLayout(layout);
88. layout.setHorizontalGroup(
89.     layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
90.         .addGroup(layout.createSequentialGroup()
91.             .addContainerGap()
92.         )
93.     );
94. layout.setVerticalGroup(
95.     layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
96.         .addGroup(layout.createSequentialGroup()
97.             .addContainerGap()
98.         )
99.     );

```

```

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
.addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT_SIZE, 550,
Short.MAX_VALUE))
.addGroup(layout.createSequentialGroup()
.addGap(251, 251, 251)
.addComponent(jLabel1)
.addGap(0, 0, Short.MAX_VALUE))
.addGroup(layout.createSequentialGroup()
.addComponent(txtcari, javax.swing.GroupLayout.PREFERRED_SIZE, 149,
javax.swing.GroupLayout.PREFERRED_SIZE)
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
.addComponent(bcari)
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
.addComponent(btndiambil1)))
.addComponent(btndiambil1)))
.addContainerGap())
83. );
84. layout.setVerticalGroup(
85. layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
86. .addGroup(layout.createSequentialGroup()
.addComponent(jLabel1)
.addGap(4, 4, 4)
87. .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
.addComponent(txtcari, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
.addComponent(bcari)
.addComponent(btndiambil1)
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
.addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 138,
javax.swing.GroupLayout.PREFERRED_SIZE)
.addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
88. );
89. setSize(new java.awt.Dimension(592, 275));
90. setLocationRelativeTo(null);
91. } // </editor-fold>

92. private void tblplgnMouseClicked(java.awt.event.MouseEvent evt) {
93. int tabelPelanggan = tblplgn.getSelectedRow();
94. FN.id_pelanggan = tblplgn.getValueAt(tabelPelanggan, 0).toString();
95. FN.nama_pelanggan = tblplgn.getValueAt(tabelPelanggan, 1).toString();
96. FN.itemTerpilih();
97. this.dispose();
98. }

99. private void bcariActionPerformed(java.awt.event.ActionEvent evt) {
100. datatable();
101. }

102. private void txtcariKeyPressed(java.awt.event.KeyEvent evt) {

```

```

103.    if (evt.getKeyCode() == KeyEvent.VK_ENTER) {
104.        datatable();
105.    }
106.    }

107.   private void btndiambil1ActionPerformed(java.awt.event.ActionEvent evt) {
108.       this.setVisible(false);
109.   }

110.  /**
111.   * @param args the command line arguments
112.   */
113.  public static void main(String args[]) {
114.      /* Set the Nimbus look and feel */
115.      //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
116.      /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and
117.         feel.
118.         For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
119.      */
120.      try {
121.          for (javax.swing.UIManager.LookAndFeelInfo info :
122.              javax.swing.UIManager.getInstalledLookAndFeels()) {
123.              if ("Nimbus".equals(info.getName())) {
124.                  javax.swing.UIManager.setLookAndFeel(info.getClassName());
125.                  break;
126.              }
127.          }
128.      } catch (ClassNotFoundException | InstantiationException | IllegalAccessException |
129.              javax.swing.UnsupportedLookAndFeelException ex) {
130.          java.util.logging.Logger.getLogger(popuppelanggan.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
131.      }
132.      //

```

**L. Popupkaryawan.java**

```
1. package Form;
2. import java.sql.*;
3. import javax.swing.JOptionPane;
4. import javax.swing.table.DefaultTableModel;
5. import java.awt.event.KeyEvent;
6. import Koneksi.koneksi;
7. /**
8. *
9. * @author USER
10. */
11. public final class popupkaryawan extends javax.swing.JFrame {
12.     private final Connection conn = new koneksi().connect();
13.     private DefaultTableModel tabmode;
14.     public Transaksi FN = null;
15.     /**
16.      Creates new form popuppelanggan
17.     */
18.     public popupkaryawan() {
19.         initComponents();
20.         datatable();
21.     }
22.     protected void datatable(){
23.         Object[] Baris = { "ID Karyawan","Nama Karyawan"};
24.         tabmode = new DefaultTableModel (null, Baris);
25.         String cariitem=txtcari.getText();
26.         try{
27.             tabmode.getDataVector().removeAllElements();
28.             String sql = "SELECT * FROM karyawan";
29.             Statement stat = conn.createStatement();
30.             ResultSet hasil = stat.executeQuery(sql);
31.             while (hasil.next()){
32.                 tabmode.addRow(new Object[]{
33.                     hasil.getString(1),
34.                     hasil.getString(2)
35.                 });
36.             }
37.             /**
38.              This method is called from within the constructor to initialize the form.
39.              WARNING: Do NOT modify this code. The content of this method is always
40.              regenerated by the Form Editor.
41.              */
42.             txtcari = new javax.swing.JTextField();
43.             bcari = new javax.swing.JButton();
44.             jLabel1 = new javax.swing.JLabel();
```



```

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
.addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT_SIZE, 550,
Short.MAX_VALUE))
.addGroup(layout.createSequentialGroup()
.addGap(251, 251, 251)
.addComponent(jLabel1)
.addGap(0, 0, Short.MAX_VALUE))
.addGroup(layout.createSequentialGroup()
.addComponent(txtcari, javax.swing.GroupLayout.PREFERRED_SIZE, 149,
javax.swing.GroupLayout.PREFERRED_SIZE)
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
.addComponent(bcari)
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
.addComponent(btndiambil1)))
.addComponent(btndiambil1)))
.addContainerGap())
83. );
84. layout.setVerticalGroup(
85. layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
86. .addGroup(layout.createSequentialGroup()
.addComponent(jLabel1)
.addGap(4, 4, 4)
87. .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
.addComponent(txtcari, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
.addComponent(bcari)
.addComponent(btndiambil1)
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
.addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 138,
javax.swing.GroupLayout.PREFERRED_SIZE)
.addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
88. );
89. setSize(new java.awt.Dimension(592, 275));
90. setLocationRelativeTo(null);
91. } // </editor-fold>

92. private void tblplgnMouseClicked(java.awt.event.MouseEvent evt) {
93. int tabelPelanggan = tblplgn.getSelectedRow();
94. FN.id_karyawan = tblplgn.getValueAt(tabelPelanggan, 0).toString();
95. FN.nama_karyawan = tblplgn.getValueAt(tabelPelanggan, 1).toString();
96. FN.itemTerpilih();
97. this.dispose();
98. }

99. private void bcariActionPerformed(java.awt.event.ActionEvent evt) {
100. datatable();
101. }

102. private void txtcariKeyPressed(java.awt.event.KeyEvent evt) {

```

```

103.    if (evt.getKeyCode() == KeyEvent.VK_ENTER) {
104.        datatable();
105.    }
106.    }

107.   private void btndiambil1ActionPerformed(java.awt.event.ActionEvent evt) {
108.       this.setVisible(false);
109.   }

110. /**
111.  * @param args the command line arguments
112. */
113. public static void main(String args[]) {
114. /* Set the Nimbus look and feel */
115. //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
116. /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and
117. feel.
118. For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
119. */
120. try {
121.     for (javax.swing.UIManager.LookAndFeelInfo info :
122.         javax.swing.UIManager.getInstalledLookAndFeels()) {
123.         if ("Nimbus".equals(info.getName())) {
124.             javax.swing.UIManager.setLookAndFeel(info.getClassName());
125.             break;
126.         }
127.     }
128. } catch (ClassNotFoundException | InstantiationException | IllegalAccessException |
129. javax.swing.UnsupportedLookAndFeelException ex) {
130.     java.util.logging.Logger.getLogger(popupkaryawan.class.getName()).log(java.util.loggin
131. g.Level.SEVERE, null, ex);
132. }
133. //</editor-fold>
134. //</editor-fold>
135. //</editor-fold>
136. //</editor-fold>
137. //</editor-fold>
138. //</editor-fold>
139. //</editor-fold>
140. //</editor-fold>

141. /* Create and display the form */
142. java.awt.EventQueue.invokeLater(() -> {
143.     new popupkaryawan().setVisible(true);
144. });
145. }

146. // Variables declaration - do not modify
147. private javax.swing.JButton bcar;
148. private javax.swing.JButton btndiambil1;
149. private javax.swing.JLabel jLabel1;
150. private javax.swing.JScrollPane jScrollPane1;
151. private javax.swing.JTable tblplgn;
152. private javax.swing.JTextField txtcar;

```

```
143.    // End of variables declaration  
144. }
```

### **M. UserID.java**

```
1.      package Form;  
  
2.      /**  
3.       *  
4.       *@author USER  
5.      */  
6.      public class UserID {  
7.          private static String kd;  
  
8.          public static void setUserLogin(String kode){  
9.              kd=kode;  
10.         }  
11.         public static String getUserLogin(){  
12.             return kd;  
13.         }
```