

**BUKU PANDUAN MAPEL PEMROGRAMAN BERORIENTASI OBJEK
TENTANG PEMBUATAN APLIKASI KASIR DAN ADMIN**

“MAKMUR JAYA”



GURU PEMBIMBING:

VEBRIYANTI DWI ANGGRAINI, M.Pd

DISUSUN OLEH:

ADHAM PRASETYO SURYO NUGROHO (XII RPL 1 / 01)

CABANG DINAS PENDIDIKAN PROVINSI JAWA TIMUR

SEKOLAH MENENGAH KEJURUAN NEGERI 2

KOTA MOJOKERTO

2022 – 2023

Kata Pengantar

Segala puji kehadirat Allah SWT. Tuhan Yang Maha Esa atas segala rahmat-Nya, sehingga buku paduan untuk mata pelajaran pemrograman berorientasi objek tentang aplikasi kasir dan admin dengan nama “Makmur Jaya” dapat tersusun hingga selesai.

Penyusunan buku panduan untuk aplikasi “Makmur Jaya” ini, bertujuan agar siswa bisa membuat aplikasi ini dari awal sampai akhir dengan baik. Dan buku ini bisa menjadi bahan ajar untuk guru memberikan materi pada saat proses pembelajaran.

Penulis menyadari bahwa tanpa bantuan dan dukungan berbagai pihak, sangatlah sulit bagi penulis untuk menyelesaikan buku panduan ini. Oleh karena itu, penulis mengucapkan banyak terima kasih kepada:

1. VEBRIYANTI DWI ANGGRAENI, M.Pd., selaku guru pembimbing mata pelajaran Pemrograman Berorientasi Objek (PBO).

Di buku panduan ini kami menjelaskan dengan semaksimal mungkin. Dengan segala keterbatasan, kami mengharapkan adanya kritik dan saran yang membangun, agar kami dapat memperbaiki buku paduan ini menjadi lebih baik lagi untuk kedepannya.

Akhir kata, semoga buku paduan pembuatan aplikasi ini dapat memberikan manfaat bagi kita semua yang membaca, khususnya untuk siswa dan guru yang membuat dan menggunakan aplikasi ini.

Mojokerto, 10 Maret 2023

Penulis

ADHAM PRASETYO SURYO NUGROHO

Daftar Isi

Isi

Kata Pengantar.....	i
Daftar Isi.....	ii
Daftar Gambar	iii
Bab 1 - Hal – Hal yang Dibutuhkan.....	1
A. Perangkat Lunak	1
B. Perangkat Keras	1
Bab 2 - Pembuatan Aplikasi	2
A. Pembuatan Database.....	2
B. Pembuatan Project Pada Netbeans	2
C. Import Library yang dibutuhkan	4
D. Instalasi Plugin Ireport Pada Netbeans.....	6
E. Pembuatan Package.....	7
F. Pembuatan Halaman Utama User Kasir	7
G. Pembuatan Halaman Menu User Admin.....	14
H. Pembuatan Sistem Koneksi Database Pada Project Ini	17
I. Pembuatan Halaman Login	18
J. Pembuatan Halaman Transaksi	21
K. Pembuatan report.....	58
L. Pembuatan Halaman Utama Admin.....	69
M. Pembuatan Halaman Kategori.....	73
N. Pembuatan Halaman Entry Barang	89
O. Pembuatan Halaman Laporan Transaksi.....	105
P. Pembuatan Exe File.....	117

Daftar Gambar

Gambar 2.1: Class Diagram untuk spesifikasi tabel yang harus dibuat	2
Gambar 2.2 : Tata letak tombol new project pada tab file netbeans.....	3
Gambar 2.3 : Pemilihan jenis project di netbeans.....	3
Gambar 2.4 : Form pengisian nama project, dan lokasi penyimpanan projectnya.....	4
Gambar 2.5 : Library yang diimport pada project Halaman_Login.....	5
Gambar 2.6 : File yang dibutuhkan intalasi plugin ireport di netbeans	6
Gambar 2.7 : Tab Design terbuka	8
Gambar 2.8 : Tab pallete.....	8
Gambar 2.9 : Halaman utama user kasir dengan tab user di klik.....	9
Gambar 2.10 : Halaman utama user kasir dengan tab transaksi terbuka	10
Gambar 2.11 : Library yang harus dimport pada halaman utama user kasir	11
Gambar 2.12 : Beberapa variabel scope global yang harus diibuat pada halaman menu utama kasir.....	11
Gambar 2.13 : Isi dari contructor halaman utama user kasir	11
Gambar 2.14 : Fungsi sudahLogin() pada halaman Utama User Kasir	12
Gambar 2.15 : Fungsi SudahLogout() pada halaman utama user kasir.	12
Gambar 2.16 : Fungsi jikaLoginUser() pada halaman utama user kasir.....	13
Gambar 2.17 : Fungsi Kosong() pada halaman utama user kasir	13
Gambar 2.18 : Fungsi KeyPressed() pada halaman utama user kasir.....	13
Gambar 2.19 : Fungsi Event dan isinya pada halaman utama user kasir (1)	14
Gambar 2.20 : Fungsi Event dan isinya pada halaman utama user kasir (2)	14
Gambar 2.21 : Halaman Menu User Admin	15
Gambar 2.22 : Library yang harus dimport pada halaman menu user admin.....	15
Gambar 2.23 : Isi dari contructor halaman menu user admin	16
Gambar 2.24 : Fungsi keyPressed() pada halaman menu user admin	16
Gambar 2.25 : Fungsi Event dan isinya pada halaman menu user admin(1).....	16
Gambar 2.26 : Library yang ada di class Jaringan.....	17
Gambar 2.27 : Variabel yang menggunakan class ‘Connection’ pada class jaringan	17
Gambar 2.28 : Konfigurasi koneksi database	17
Gambar 2.29 : Halaman Login	18
Gambar 2.30 : Library yang harus diimport pada halaman login	19
Gambar 2.31 : Beberapa variabel scope global yang ada di halaman login	19
Gambar 2.32 : Fungsi Login() pada halaman login (1)	19
Gambar 2.33 : Fungsi Login() pada halaman login (2)	20
Gambar 2.34 : Fungsi getId() pada halaman login	20
Gambar 2.35 : Fungsi jikaAktif() pada halaman login	20
Gambar 2.36 : fungsi keyPressed() pada halaman login.....	20
Gambar 2.37 : Contructor pada halaman login	21
Gambar 2.38 : Fungsi Event dan isinya pada halaman login (1)	21
Gambar 2.39 : Halaman Form Transaksi (1)	22
Gambar 2.40 : Halaman Form Transaksi (2)	23
Gambar 2.41 : Library yang harus diimport pada halaman transaksi (1)	24
Gambar 2.42 : Library yang harus diimport pada halaman transaksi (2)	24

Gambar 2.43 : Variabel scope global yang ada di sistem form transaksi	25
Gambar 2.44 : Constructor pada halaman form transaksi.....	25
Gambar 2.45 : Fungsi jikaAktif() pada halaman form transaksi.....	25
Gambar 2.46 : Fungsi Kosong() pada halaman form transaksi.....	26
Gambar 2.47 : Fungsi bisa_isi() pada halaman form transaksi.....	26
Gambar 2.48 : Fungsi tidak_bisa_isi() pada halaman transaksi.....	27
Gambar 2.49 : Fungsi tombol_mati() pada halaman form transaksi.....	27
Gambar 2.50 : Fungsi tombol_hidup() pada halaman form transaksi.....	27
Gambar 2.51 : Fungsi placeholder() pada halaman form transaksi	28
Gambar 2.52 : Fungsi keyPressed() pada halaman form transaksi	28
Gambar 2.53 : Fungsi ambilDataTabelBarang() pada halaman form transaksi.....	29
Gambar 2.54 : Fungsi sistemTabelTransaksi() pada halaman form transaksi	29
Gambar 2.55 : Fungsi Simpan() pada halaman form transaksi (1)	30
Gambar 2.56 : Fungsi Simpan() pada halaman form transaksi (2)	30
Gambar 2.57 : Fungsi Simpan() pada halaman form transaksi (3)	31
Gambar 2.58 : Fungsi Simpan() pada halaman form transaksi (4)	31
Gambar 2.59 : Fungsi Simpan() pada halaman form transaksi (5).....	32
Gambar 2.60 : Fungsi Simpan() pada halaman form transaksi (6)	32
Gambar 2.61 : Fungsi Simpan() pada halaman form transaksi (7)	33
Gambar 2.62 : Fungsi updateJumlah() pada halaman form transaksi (1)	33
Gambar 2.63 : Fungsi updateJumlah() pada halaman form transaksi (2)	34
Gambar 2.64 : Fungsi updateJumlah() pada halaman form transaksi (3)	34
Gambar 2.65 : Fungsi hapus() pada halaman form transaksi (1)	35
Gambar 2.66 : Fungsi hapus() pada halaman form transaksi (2)	35
Gambar 2.67 : Fungsi hapus() pada halaman form transaksi (3)	36
Gambar 2.68 : Fungsi isi_tableTransaksi() pada halaman form transaksi (1)	36
Gambar 2.69 : Fungsi isi_tableTransaksi() pada halaman form transaksi (2)	37
Gambar 2.70 : Fungsi isi_tableBarang() pada halaman form transaksi (1)	37
Gambar 2.71 : Fungsi isi_tableBarang() pada halaman form transaksi (2)	38
Gambar 2.72 : Fungsi AutoIncrement() pada halaman form transaksi (1)	38
Gambar 2.73 : Fungsi AutoIncrement() pada halaman form transaksi (2)	39
Gambar 2.74 : Fungsi AutoIncrement() pada halaman form transaksi (3)	39
Gambar 2.75 : Fungsi AutoIncrement() pada halaman form transaksi (4)	40
Gambar 2.76 : Fungsi AutoIncrement() pada halaman form transaksi (5)	40
Gambar 2.77 : Fungsi AutoIncrement() pada halaman form transaksi (6)	41
Gambar 2.78 : Fungsi hitungDigitAngka() pada halaman form transaksi	41
Gambar 2.79 : Fungsi hitungKarangter() pada halaman form transaksi (1)	42
Gambar 2.80 : Fungsi hitungKarangter() pada halaman form transaksi (2)	42
Gambar 2.81 : Fungsi hitungKarangter() pada halaman form transaksi (3)	43
Gambar 2.82 : Fungsi hitungKarangter() pada halaman form transaksi (4)	43
Gambar 2.83 : Fungsi searchBarang() pada halaman form transaksi (1).....	43
Gambar 2.84 : Fungsi searchBarang() pada halaman form transaksi (2).....	44
Gambar 2.85 : Fungsi searchBarang() pada halaman form transaksi (2).....	44
Gambar 2.86 : Fungsi searchTransaksi() pada halaman form transaksi (1).....	45

Gambar 2.87 : Fungsi searchTransaksi() pada halaman form transaksi (2).....	45
Gambar 2.88 : Fungsi searchTransaksi() pada halaman form transaksi (3).....	46
Gambar 2.89 : Fungsi functionEdit() pada halaman form transaksi	46
Gambar 2.90 : Fungsi functionTotalTagihan() pada halaman form transaksi	47
Gambar 2.91 : Fungsi bayar() pada halaman form transaksi (1)	47
Gambar 2.92 : Fungsi bayar() pada halaman form transaksi (2)	48
Gambar 2.93 : Fungsi bayar() pada halaman form transaksi (3)	48
Gambar 2.94 : Fungsi bayar() pada halaman form transaksi (4)	49
Gambar 2.95 : Fungsi bayar() pada halaman form transaksi (5)	49
Gambar 2.96 : Fungsi bayar() pada halaman form transaksi (6)	50
Gambar 2.97 : Fungsi bayar() pada halaman form transaksi (7)	50
Gambar 2.98 : Fungsi subtotalHargaBarang() pada halaman form transaksi	51
Gambar 2.99 : Fungsi transaksiBaru() pada halaman form transaksi	51
Gambar 2.100 : Fungsi printData() pada halaman form transaksi (1)	52
Gambar 2.101 : Fungsi printData() pada halaman form transaksi (2)	52
Gambar 2.102 : Fungsi ambilKodeTransaksiTerakhir() pada halaman form transaksi	53
Gambar 2.103 : Fungsi ambilKodeBarangTerkait() pada halaman form transaksi	53
Gambar 2.104 : Fungsi mergePembelianBarang() pada halaman form transaksi (1)	54
Gambar 2.105 : Fungsi mergePembelianBarang() pada halaman form transaksi (2)	54
Gambar 2.106 : Fungsi Event dan isinya pada halaman Transaksi (1).....	55
Gambar 2.107 : Fungsi Event dan isinya pada halaman Transaksi (2).....	55
Gambar 2.108 : Fungsi Event dan isinya pada halaman Transaksi (3).....	56
Gambar 2.109 : Fungsi Event dan isinya pada halaman Transaksi (4).....	56
Gambar 2.110 : Fungsi Event dan isinya pada halaman Transaksi (5).....	57
Gambar 2.111 : Fungsi Event dan isinya pada halaman Transaksi (6).....	57
Gambar 2.112 : Fungsi Event dan isinya pada halaman Transaksi (7).....	58
Gambar 2.113 : Settingan koneksi JDBC Ireport	59
Gambar 2.114 : Query yang bisa diisikan pada kotak query Ireport	60
Gambar 2.115 : Tab pallete pada ireport	61
Gambar 2.116 : Tab report inspector	61
Gambar 2.117 : properties parameter id_tr	62
Gambar 2.118 : Default value parameter id_tr	62
Gambar 2.119 : Contoh ireport yang menggunakan 2 tabel (1).....	63
Gambar 2.120 : Contoh ireport yang menggunakan 2 tabel (2).....	64
Gambar 2.121 : Membuat group elemen ireport dengan group wizard	64
Gambar 2.122 : Contoh isi dari properti printWhenExpression	65
Gambar 2.123 : File ireport sudah terbuka	66
Gambar 2.124 : Compatibility configuration compiling.....	66
Gambar 2.125 : compiler ireport 3.5.1 sudah terinstal	67
Gambar 2.126 : setting compatibility file dan compiler	67
Gambar 2.127 : letak tombol compile file jrxml.....	68
Gambar 2.128 : tulisan pada bagian report state bahwa file jrxml sudah berhasil di compile.....	68
Gambar 2.129 : file null.jasper hasil dari compiling file jrxml.....	68
Gambar 2.130 : Halaman Utama Admin (1).....	69

Gambar 2.131 : Halaman Utama Admin (2).....	70
Gambar 2.132 : Halaman Utama Admin (3).....	71
Gambar 2.133 : Isi dari constructor pada halaman utama admin.....	72
Gambar 2.134 : Fungsi event dan isinya pada halaman utama admin (1)	72
Gambar 2.135 : Fungsi event dan isinya pada halaman utama admin (2)	72
Gambar 2.136 : Halaman Kategori	73
Gambar 2.137 : Library yang dibutuhkan untuk sistem CRUD kategori	74
Gambar 2.138 : Variabel scope global yang ada di halaman CRUD kategori.....	74
Gambar 2.139 : Constructor pada halaman CRUD kategori	74
Gambar 2.140 : Fungsi Kosong() pada halaman CRUD kategori.	75
Gambar 2.141 : Fungsi bisa_isi() pada halaman CRUD kategori.....	75
Gambar 2.142 : Fungsi tidak_bisa_isi() pada halaman CRUD kategori.....	75
Gambar 2.143 : Fungsi tombol_mati() pada halaman CRUD kategori	75
Gambar 2.144 : Fungsi tombol_hidup() pada halaman CRUD kategori	76
Gambar 2.145 : Fungsi placeholder() di halaman CRUD kategori.....	76
Gambar 2.146 : Fungsi keyPressed() pada halaman CRUD kategori.....	76
Gambar 2.147 : Fungsi Simpan() pada halaman CRUD kategori (1).....	77
Gambar 2.148 : Fungsi Simpan() pada halaman CRUD kategori (2).....	77
Gambar 2.149 : Fungsi Simpan() pada halaman CRUD kategori (3).....	78
Gambar 2.150 : Fungsi update() pada halaman CRUD kategori (1)	78
Gambar 2.151 : Fungsi update() pada halaman CRUD kategori (2)	79
Gambar 2.152 : Fungsi hapus() pada halaman CRUD kategori (1).....	79
Gambar 2.153 : Fungsi hapus() pada halaman CRUD kategori (2).....	80
Gambar 2.154 : Fungsi hapus() pada halaman CRUD kategori (3).....	80
Gambar 2.155 : Fungsi isi_table() pada halaman CRUD kategori (1).....	81
Gambar 2.156 : Fungsi isi_table() pada halaman CRUD kategori (2).....	81
Gambar 2.157 : Fungsi AutoIncrement() pada halaman CRUD kategori (1).....	82
Gambar 2.158 : Fungsi AutoIncrement() pada halaman CRUD kategori (2).....	82
Gambar 2.159 : Fungsi hitungDigitAngka() pada halaman CRUD kategori.....	83
Gambar 2.160 : Fungsi hitungKarakter() pada halaman CRUD kategori (1).....	83
Gambar 2.161 : Fungsi hitungKarakter() pada halaman CRUD kategori (2)	84
Gambar 2.162 : Fungsi hitungKarakter() pada halaman CRUD kategori (3)	84
Gambar 2.163 : Fungsi hitungKarakter() pada halaman CRUD kategori (4)	85
Gambar 2.164 : Fungsi Search() pada halaman CRUD kategori (1)	85
Gambar 2.165 : Fungsi Search() pada halaman CRUD kategori (2)	86
Gambar 2.166 : Fungsi FunctionEdit() pada halaman CRUD kategori	86
Gambar 2.167 : Fungsi Refresh Halaman pada halaman CRUD Kategori.....	86
Gambar 2.168 : Fungsi Event dan isinya pada halaman CRUD Kategori (1)	87
Gambar 2.169 : Fungsi Event dan isinya pada halaman CRUD Kategori (2)	87
Gambar 2.170 : Fungsi Event dan isinya pada halaman CRUD Kategori (3)	88
Gambar 2.171 : Fungsi Event dan isinya pada halaman CRUD Kategori (4)	88
Gambar 2.172 : Fungsi Event dan isinya pada halaman CRUD Kategori (5)	88
Gambar 2.173 : Halaman Entry Barang.....	89
Gambar 2.174 : Library yang dibutuhkan untuk sistem CRUD entry barang	90

Gambar 2.175 : Beberapa variabel scope global pada halaman entry barang	90
Gambar 2.176 : Isi dari Constructor pada halaman entry barang	90
Gambar 2.177 : Fungsi Kosong() pada halaman entry barang	91
Gambar 2.178 : Fungsi bisa_isi() pada halaman entry barang.....	91
Gambar 2.179 : Fungsi tidak_bisa_isi() di halaman entry barang	91
Gambar 2.180 : Fungsi tombol_mati() di halaman entry barang.....	91
Gambar 2.181 : Fungsi tombol_hidup() pada halaman entry barang	92
Gambar 2.182 : Fungsi placeHolder() pada halaman entry barang	92
Gambar 2.183 : Fungsi keyPressed() pada halaman entry barang.....	92
Gambar 2.184 : Fungsi Simpan() di halaman entry barang (1)	93
Gambar 2.185 : Fungsi Simpan() di halaman entry barang (2)	93
Gambar 2.186 : Fungsi Simpan() di halaman entry barang (3)	94
Gambar 2.187 : Fungsi Update() di halaman entry barang (1)	94
Gambar 2.188 : Fungsi Update() di halaman entry barang (2)	95
Gambar 2.189 : Fungsi Update() di halaman entry barang (3)	95
Gambar 2.190 : Fungsi hapus() di halaman entry barang(1)	96
Gambar 2.191 : Fungsi hapus() di halaman entry barang(2)	96
Gambar 2.192 : Fungsi isi_table() di halaman entry barang(1)	97
Gambar 2.193 : Fungsi isi_table() di halaman entry barang(2)	97
Gambar 2.194 : Fungsi isiItemKategori() pada halaman entry barang.....	98
Gambar 2.195 : Fungsi tambahStock() pada halaman entry barang(1)	98
Gambar 2.196 : Fungsi tambahStock() pada halaman entry barang(2)	99
Gambar 2.197 : Fungsi tambahStock() pada halaman entry barang(3)	99
Gambar 2.198 : Fungsi AutoIncrement() pada halaman entry barang(1).....	100
Gambar 2.199 : Fungsi AutoIncrement() pada halaman entry barang(2)	100
Gambar 2.200 : Fungsi Search() pada halaman entry barang(1)	101
Gambar 2.201 : Fungsi Search() pada halaman entry barang(2)	101
Gambar 2.202 : Fungsi Search() pada halaman entry barang(3)	102
Gambar 2.203 : Fungsi functionEdit() pada halaman entry barang	102
Gambar 2.204 : Fungsi Event dan isinya pada halaman entry barang (1)	103
Gambar 2.205 : Fungsi Event dan isinya pada halaman entry barang (2)	103
Gambar 2.206 : Fungsi Event dan isinya pada halaman entry barang (3)	104
Gambar 2.207 : Fungsi Event dan isinya pada halaman entry barang (4)	104
Gambar 2.208 : Fungsi Event dan isinya pada halaman entry barang (5)	105
Gambar 2.209 : Fungsi Event dan isinya pada halaman entry barang (6)	105
Gambar 2.210 : Halaman laporan transaksi.....	106
Gambar 2.211 : Class Datechooser pada library datechooser.....	107
Gambar 2.212 : Properti textReference tiap komponen datechooser.....	108
Gambar 2.213 : DateFormat untuk datechooser	109
Gambar 2.214 : Library yang dibutuhkan untuk sistem laporan transaksi (1).....	109
Gambar 2.215 : Library yang dibutuhkan untuk sistem laporan transaksi (2).....	110
Gambar 2.216 : Variable scope global pada sistem halaman laporan transaksi	110
Gambar 2.217 : Constructor pada halaman laporan transaksi	110
Gambar 2.218 : Fungsi Kosong() di halaman laporan transaksi.....	111

Gambar 2.219 : Fungsi sistemTabelTransaksi() pada halaman laporan transaksi	111
Gambar 2.220 : Fungsi isi_tableTransaksi() pada halaman laporan transaksi (1)	111
Gambar 2.221 : Fungsi isi_tableTransaksi() pada halaman laporan transaksi (2)	112
Gambar 2.222 : Fungsi isi_tableTransaksi() pada halaman laporan transaksi (3)	112
Gambar 2.223 : Fungsi isi_tableTransaksiSemua() pada halaman laporan transaksi (1)	113
Gambar 2.224 : Fungsi isi_tableTransaksiSemua() pada halaman laporan transaksi (2)	113
Gambar 2.225 : Fungsi ambilJarakTanggal() pada halaman laporan transaksi (1).....	114
Gambar 2.226 : Fungsi ambilJarakTanggal() pada halaman laporan transaksi (2).....	114
Gambar 2.227 : Fungsi hitungDigitAngka() pada halaman laporan transaksi.....	115
Gambar 2.228 : Fungsi printData() pada halaman laporan transaksi (1)	115
Gambar 2.229 : Fungsi printData() pada halaman laporan transaksi (2)	116
Gambar 2.230 : Fungsi printData() pada halaman laporan transaksi (3)	116
Gambar 2.231 : Fungsi Event dan isinya pada halaman laporan transaksi (1)	117
Gambar 2.232 : Fungsi Event dan isinya pada halaman laporan transaksi (2)	117
Gambar 2.233 : Tool untuk clean & build	118
Gambar 2.234 : Build project berhasil dilakukan	118
Gambar 2.235 : Halaman home screen software advance installer	119
Gambar 2.236 : Platform instalasi produk berubah menjadi 64 bit	120
Gambar 2.237 : Folder src sudah masuk pada list installed file.	121
Gambar 2.238 : Shortcut aplikasi Makmur jaya	122
Gambar 2.239 : Prompt konfirmasi perubahan package type.....	122
Gambar 2.240 : Kompresi diubah menjadi LZMA.....	123

Bab 1 - Hal – Hal yang Dibutuhkan

A. Perangkat Lunak

Perangkat lunak yang dibutuhkan:

- a. Windows 11 sebagai Operating System.
- b. MariaDB sebagai RDBMS. MariaDB ada di Software XAMPP versi php 8.2.
- c. Netbeans versi 8.2.
- d. Plugin ireport versi 3.5.1 sudah terinstal di Netbeans.
- e. TIBCO Jaspersoft Studio-6.20.0

Untuk neatbeans 8.2, dan plugin ireport versi 3.5.1, xampp, jasper studio bisa di download pada link ini:

https://drive.google.com/drive/folders/1lEEXMMEknSGi0kNYjeb6G7ariQ3U5x6U?usp=share_link

B. Perangkat Keras

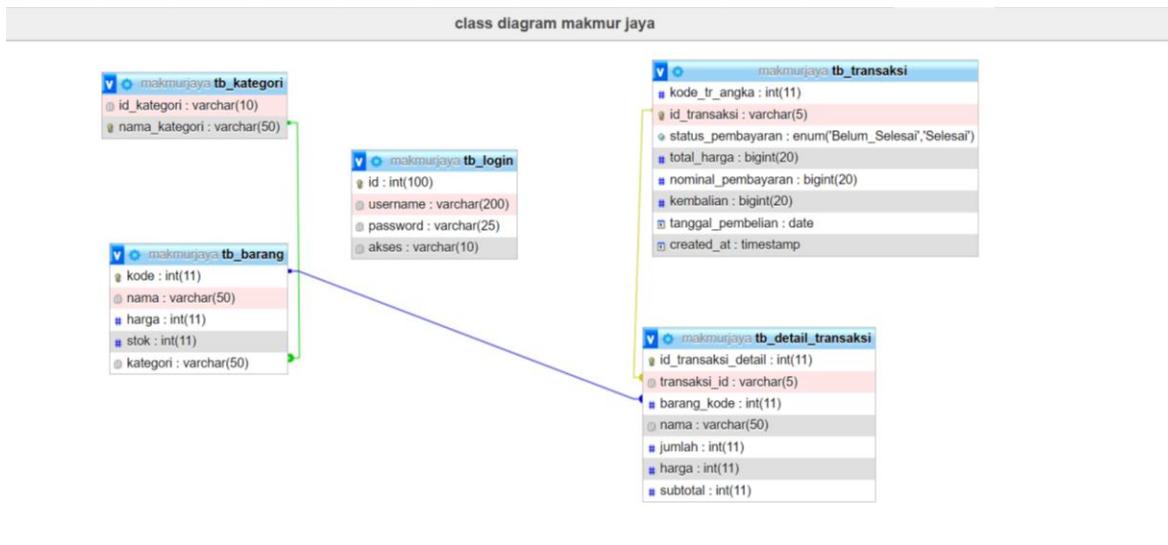
Perangkat keras yang dibutuhkan untuk membuat aplikasi ini:

- a. Laptop dengan spesifikasi 11th Gen Intel(R) Core(TM) i3-1115G4 3.00GHz; RAM 8Gb DDR4.
- b. Mouse atau TouchPad sebagai peralatan antarmuka.
- c. Keyboard sebagai peralatan antarmuka.
- d. Monitor laptop dengan ukuran 14inch.

Bab 2 - Pembuatan Aplikasi

A. Pembuatan Database

Database ini digunakan untuk menyimpan data – data untuk barang, transaksi, dan lain – lain. Untuk membuat database ini anda bisa menggunakan tool phpMyAdmin. Di phpMyAdmin anda bisa membuat database dengan nama “**makmurjaya**”. Setelah itu anda bisa membuat beberapa tabel dengan spesifikasi seperti pada gambar 2.1 berikut.



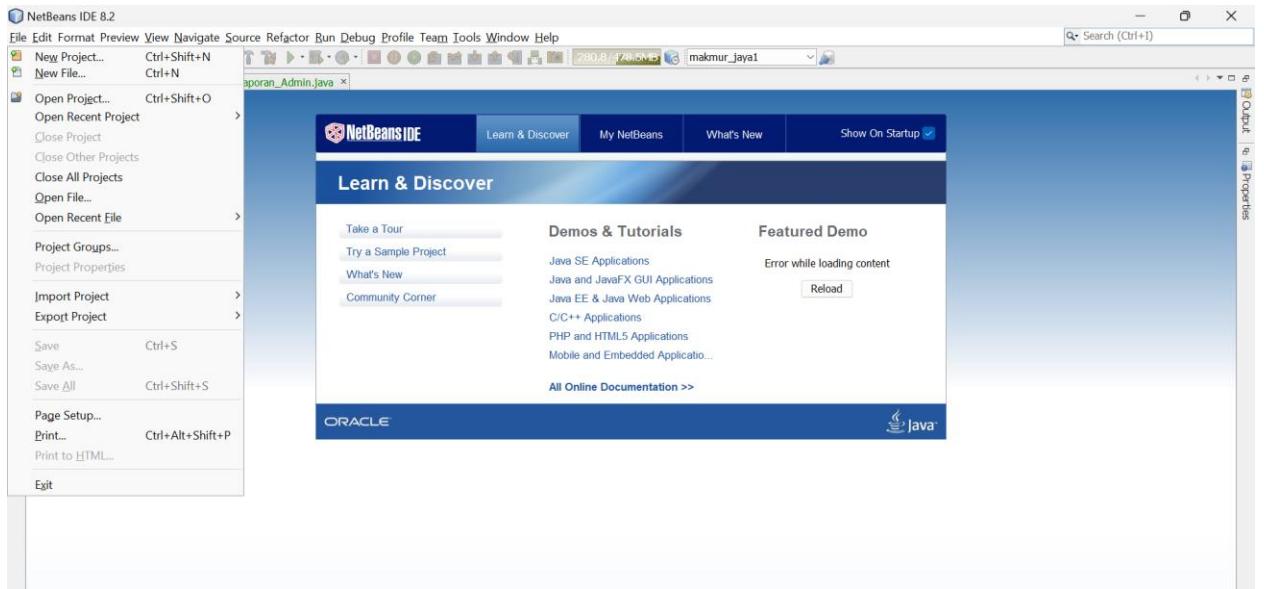
Gambar 2.1: Class Diagram untuk spesifikasi tabel yang harus dibuat

Setelah semua tabel dibuat, lalu isikan dan password pada tabel login untuk login user (kasir) dan login admin.

B. Pembuatan Project Pada Netbeans

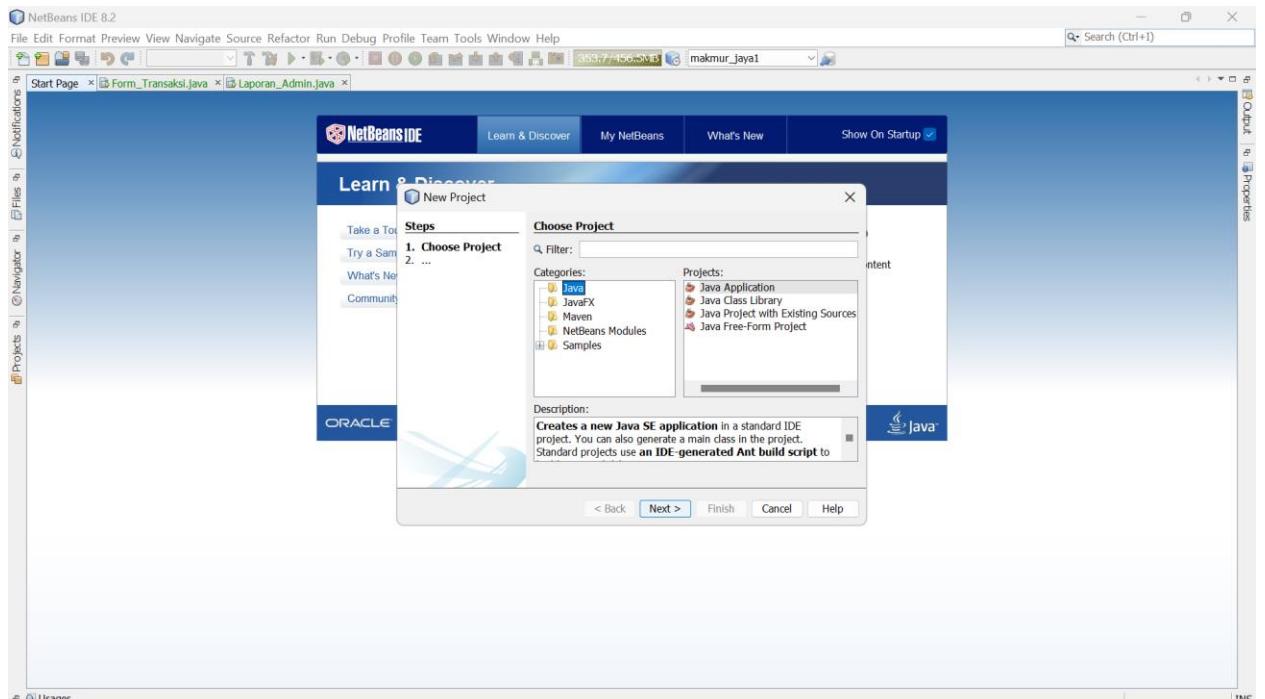
Pembuatan project pada netbeans ini bertujuan agar anda bisa mengelola file, package, dan library yang digunakan. Untuk membuat project pada netbeans ini ada beberapa cara yang harus anda lakukan, antara lain:

1. Klik tab file dan pilih New Project atau tekan tombol ‘Ctrl + Shift + N’.



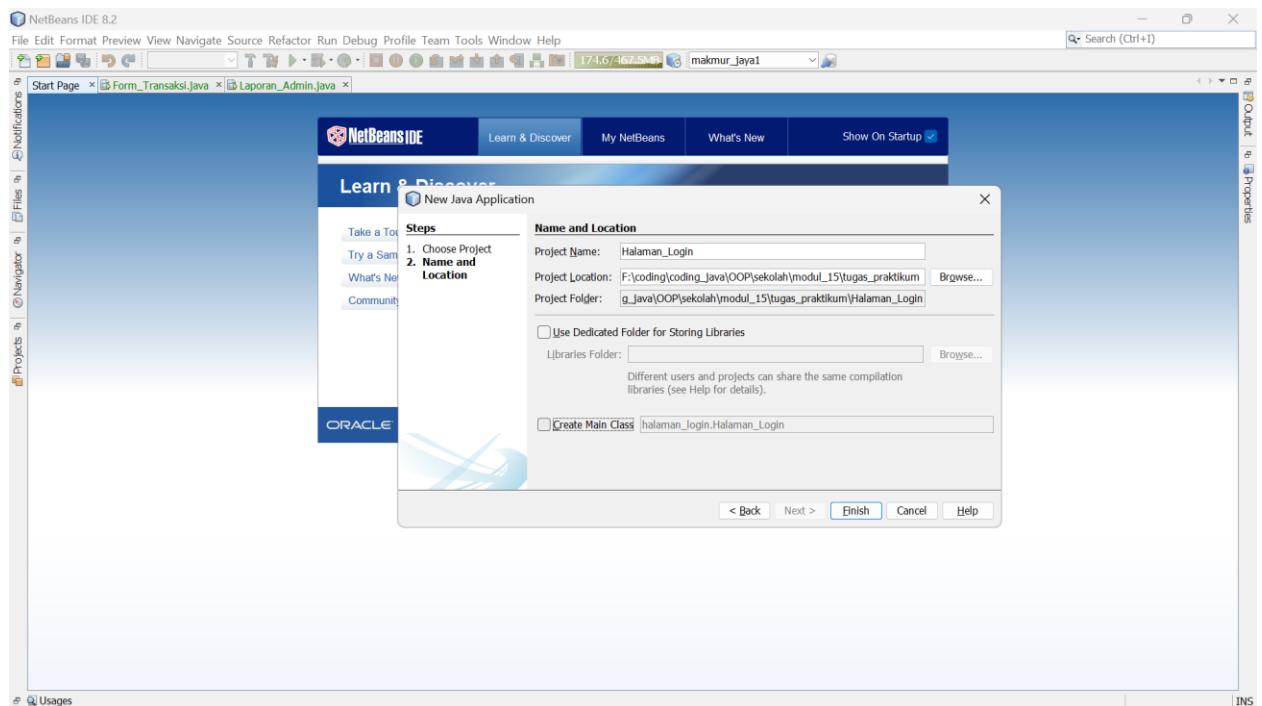
Gambar 2.2 : Tata letak tombol new project pada tab file netbeans

2. Pilih ‘Java Application’ pada jenis project.



Gambar 2.3 : Pemilihan jenis project di netbeans

3. Isi nama project dengan ‘Halaman Login’ pada bagian ‘Project Name’. Pada bagian ‘Project Location’ isi dengan path dimana anda mau menyimpan project aplikasinya. Jika pada bagian ‘Create Main Class’ tercentang, hilangkan centangnya. Jika sudah klik finish.



Gambar 2.4 : Form pengisian nama project, dan lokasi penyimpanan projectnya

C. Import Library yang dibutuhkan

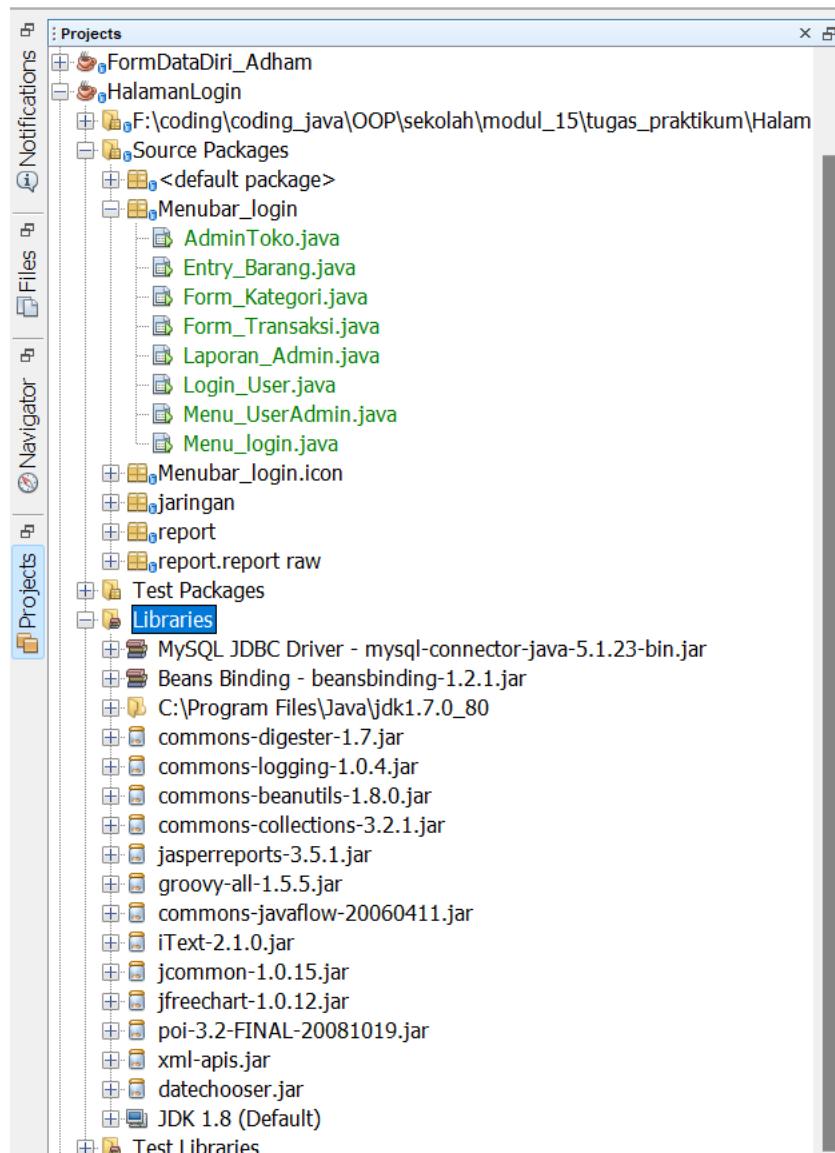
Import library ini digunakan untuk agar sistem pada project ini dapat berjalan dengan lancar. Library yang diimport ini sebagian besar harus menggunakan penyimpanan eksternal, yang mana tidak disediakan oleh java jdk versi 8. Untuk mendownload library yang dibutuhkan, anda bisa mendownload dari link berikut:

https://drive.google.com/drive/folders/1A0Voqf4YSOt0jfQ3BWzrZX7V9R6BhY7s?usp=share_link

Jika library sudah terdownload, maka anda bisa mengimport library tersebut ke project yang sudah anda buat. Berikut cara mengimport library ke project anda:

1. Klik kanan pada bagian folder libraries => pilih Add Project atau Add Library atau Add Jar / Folder. Untuk saat ini penulis hanya bisa mencontohkan untuk import dari jar atau folder saja.
2. Cari tempat file jar berada => pilih file jar tersebut => pilih open. Maka file jar yang dipilih akan sudah tampil di bagian library.

Berikut beberapa file jar yang harus anda import pada project ini



Gambar 2.5 : Library yang diimport pada project
Halaman_Login

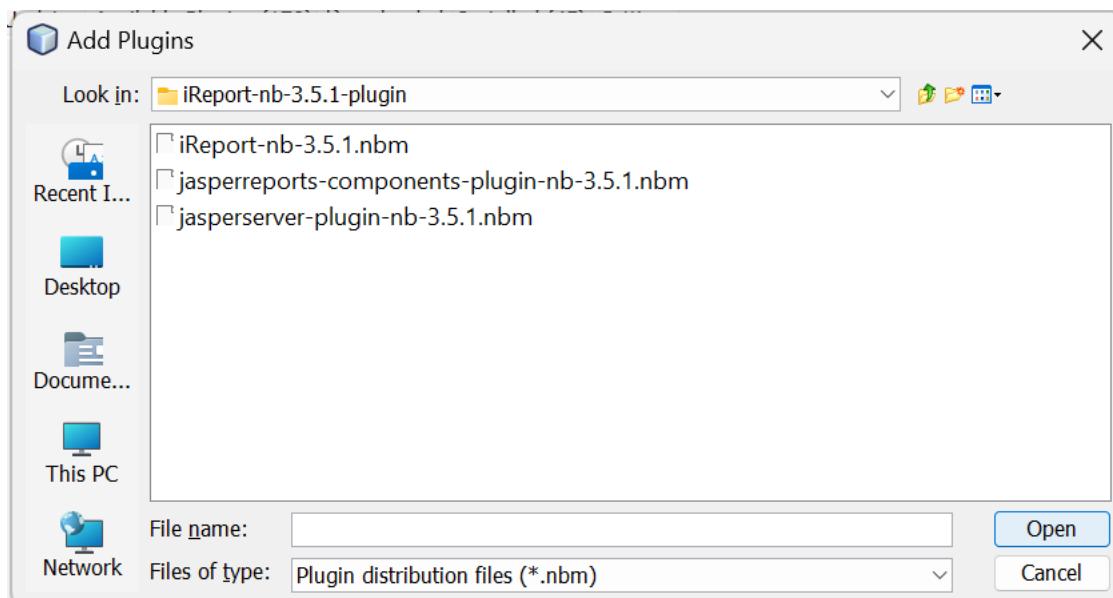
D. Instalasi Plugin ireport Pada Netbeans

Instal plugin ireport ini bertujuan agar anda bisa membuat file report pada netbeans. Kalau anda tidak menginstal plugin ireport pada netbeans, maka untuk membuat report harus menggunakan software terpisah. Berikut cara instalasi plugin ireport pada netbeans:

1. Download file zip pluginnya pada link berikut:

https://drive.google.com/file/d/1lloqUSpnXjRgXaUbbhzVW2iLMl1htP3T/view?usp=share_link

2. Ekstrak file zip tersebut.
3. Pada netbeans, tekan tab Tools pada Toolbar => Plugins => pilih tab downloaded => tekan tombol Add Plugins.
4. Cari dimana file jar plugin => pilih 3 file seperti pada gambar 2.6 => tekan open.



Gambar 2.6 : File yang dibutuhkan intalasi plugin ireport di netbeans

5. Pada saat ini beberapa file yang dibutuhkan untuk instalasi plugin sudah muncul di pop up menu Plugins. Centang semua file => tekan tombol instal.
6. Akan muncul prompt instalasi, klik next saja sampai selesai.
7. Jika ireport sudah terinstal, anda bisa merestart netbeans.
8. Anda sudah bisa membuat ireport pada netbeans.

E. Pembuatan Package

Package ini digunakan untuk mengorganisir atau mengelompokkan file – file ‘source code’ pada project. Untuk membuat package, yang harus dilakukan yaitu:

1. Klik tab project, lalu pilih project ‘Halaman_Login’. Klik kanan pada bagian Source Packages’. Pilih ‘new’, dan pilih ‘Java Package’.
2. Isi nama package dan path lokasi penyimpanan package projectnya. Untuk lokasi penyimpanan, penulis menyarankan untuk menggunakan lokasi penyimpanan default saja.
3. Jika sudah mengisi semua bagian, klik finish.

F. Pembuatan Halaman Utama User Kasir

Halaman utama untuk user kasir ini adalah halaman yang digunakan untuk user menuju halaman **Login** dan juga digunakan untuk menjadi perantara ke halaman **Transaksi**. Halaman utama user kasir ini cuma memiliki 2 tab menu, tab user dengan 2 sub menu (login, logout), tab transaksi dengan sub menu form transaksi. Berikut cara membuat halaman utama user kasir:

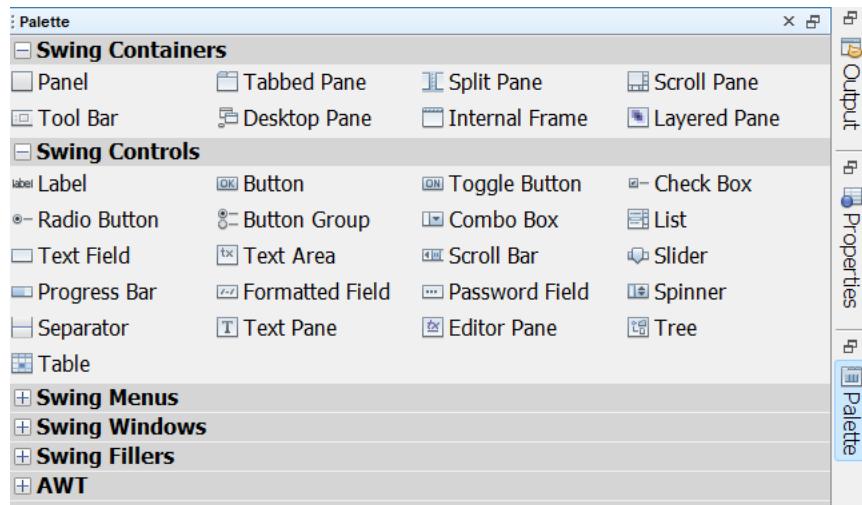
1. Klik kanan pada package yang sudah dibuat => New => JFrame Form.
2. Isi ‘Class Name’, ‘location’ untuk tempat penyimpanan file Jframe form nya. Untuk location disarankan untuk menggunakan pilihan default saja.
3. Jika semua sudah terisi, klik finish.

Jika sudah membuat file jframe-nya anda bisa mulai mendesain halaman utama user kasir ini di bagian tab ‘Design’.



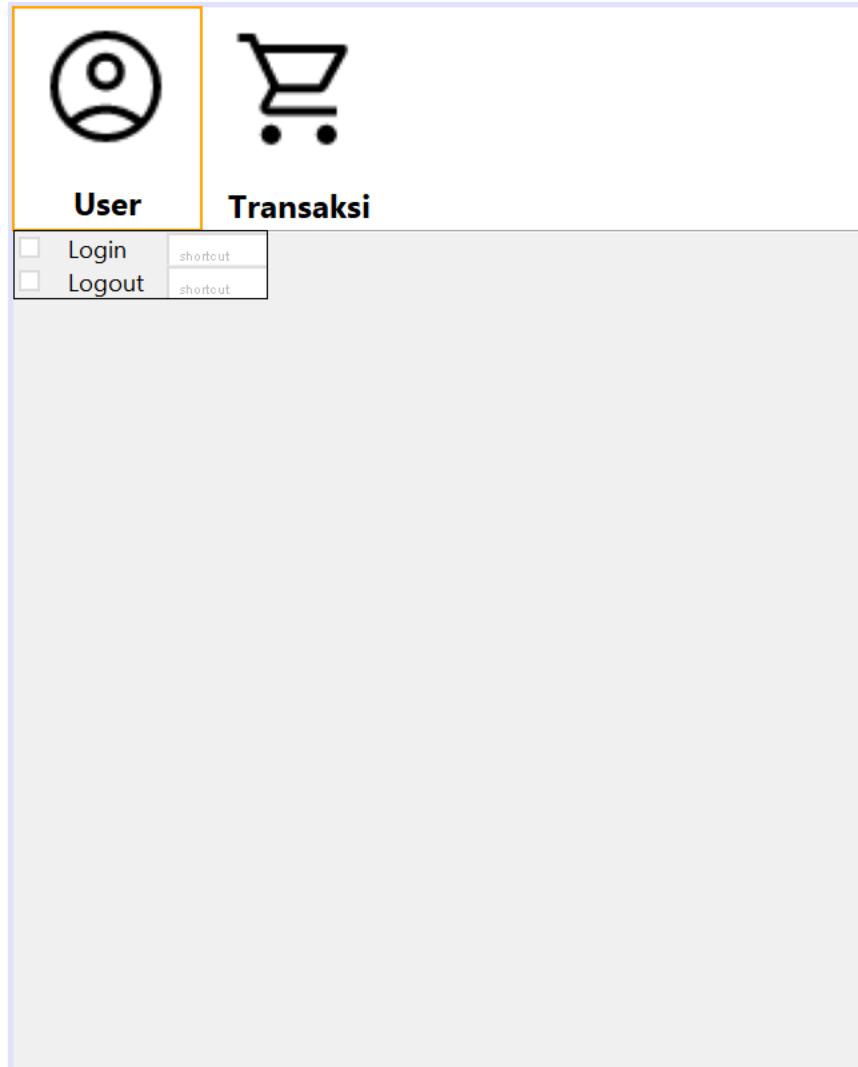
Gambar 2.7 : Tab Design terbuka

Untuk mendesain elemen halaman ini, anda bisa memilih elemen yang disediakan pada tab ‘pallete’.

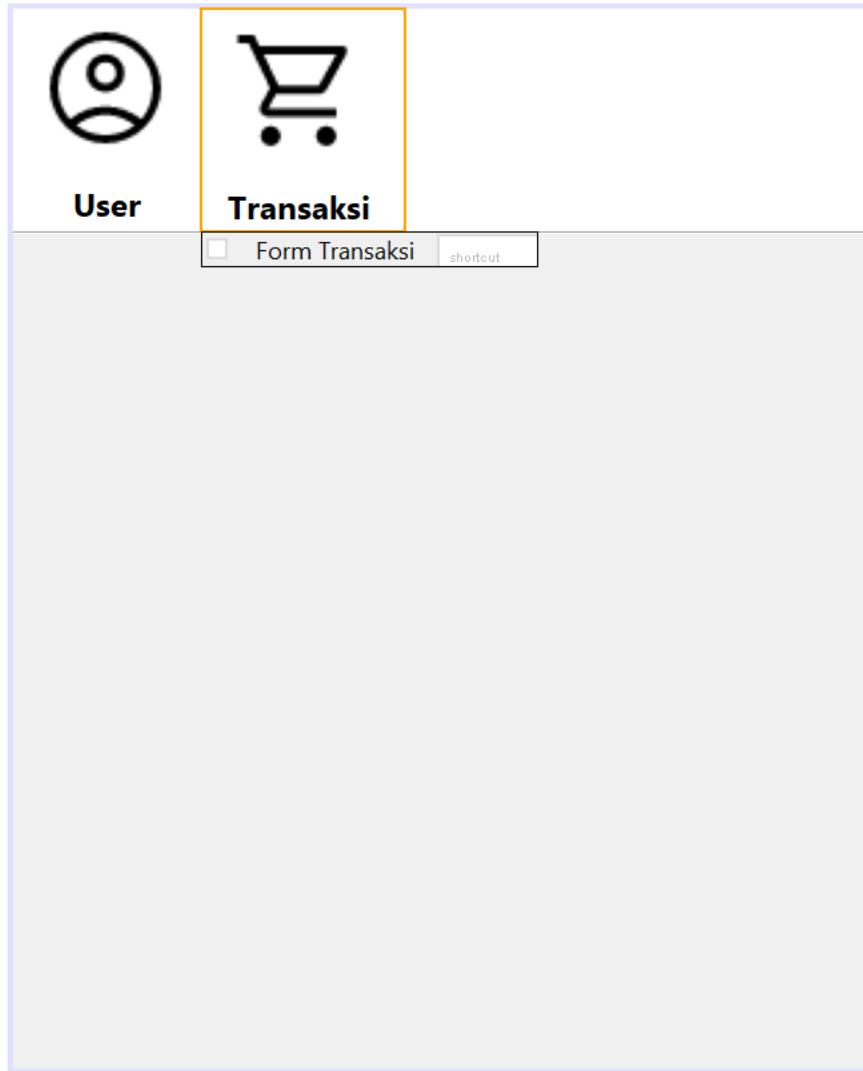


Gambar 2.8 : Tab pallete

Untuk menggunakan elemen pada pallete ini, anda cukup menekan elemen yang ingin dipakai ke canvas form, atau anda juga bisa melakukan drag and drop elemen ke canvas form. Untuk halaman utama user kasir ini, anda bisa mendesain seperti yang ada di gambar 2.9 dan 2.10. Jangan lupa untuk selalu menengahkan tiap frame.



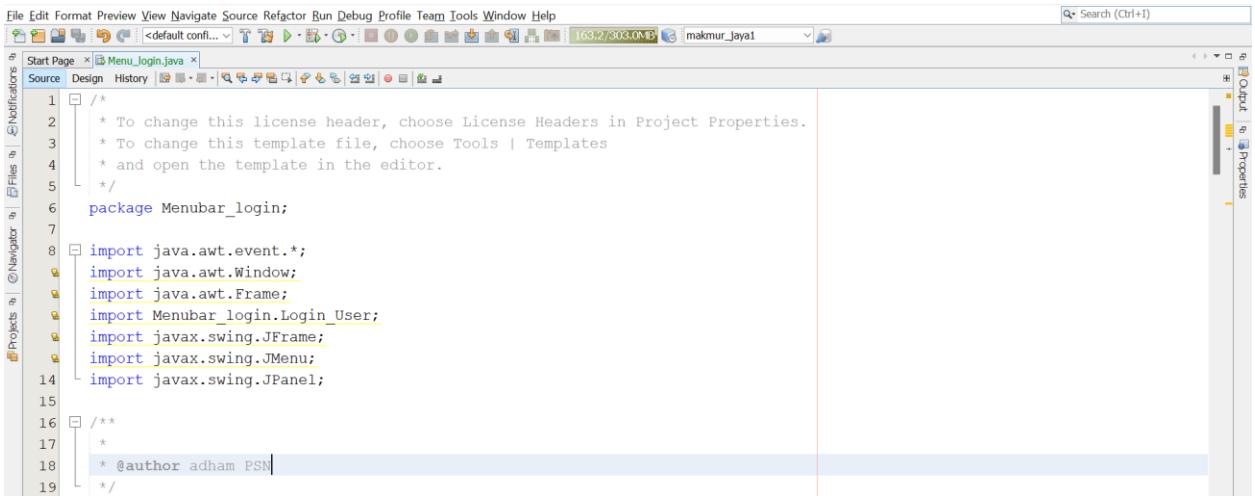
Gambar 2.9 : Halaman utama user kasir dengan tab user di klik



Gambar 2.10 : Halaman utama user kasir dengan tab transaksi terbuka

Jika desain sudah dibuat, jangan lupa untuk mengganti nama variable tiap elemen, agar mempermudah untuk diingat. Setelah itu buka tab Soure pada halaman ini, lalu buat sistemnya agar halaman utama user ini dapat berjalan dengan baik. Berikut cara membuat sistem untuk halaman utama user kasir:

1. Import beberapa library yang dibutuhkan.



```
1  /*
2   * To change this license header, choose License Headers in Project Properties.
3   * To change this template file, choose Tools | Templates
4   * and open the template in the editor.
5   */
6  package Menubar_login;
7
8  import java.awt.event.*;
9  import java.awt.Window;
10 import java.awt.Frame;
11 import Menubar_login.Login_User;
12 import javax.swing.JFrame;
13 import javax.swing.JMenu;
14 import javax.swing.JPanel;
15
16 /**
17 *
18 * @author adham PSN
19 */
```

Gambar 2.11 : Library yang harus dimport pada halaman utama user kasir

2. Buat beberapa variabel pada scope global.

```
20 public class Menu_login extends javax.swing.JFrame {
21
22     private boolean sudahLogin = false;
23     private boolean sudahLogout = false;
24     public static Menu_login Instance;
25     public boolean masukLogin = false;
26     public JPanel jPanel11;
27 }
```

Gambar 2.12 : Beberapa variabel scope global yang harus diibuat pada halaman menu utama kasir.

3. Pada constructor halaman ini, isi seperti gambar 2.13.

```
29 /**
30  * Creates new form Menu_login
31  */
32 public Menu_login() {
33     initComponents();
34
35     Instance = this;
36
37     jPanel11 = jPanel11;
38
39     jikaLoginUser();
40
41     keyPressed();
42     Kosong();
43 }
44
45 }
```

Gambar 2.13 : Isi dari constructor halaman utama user kasir

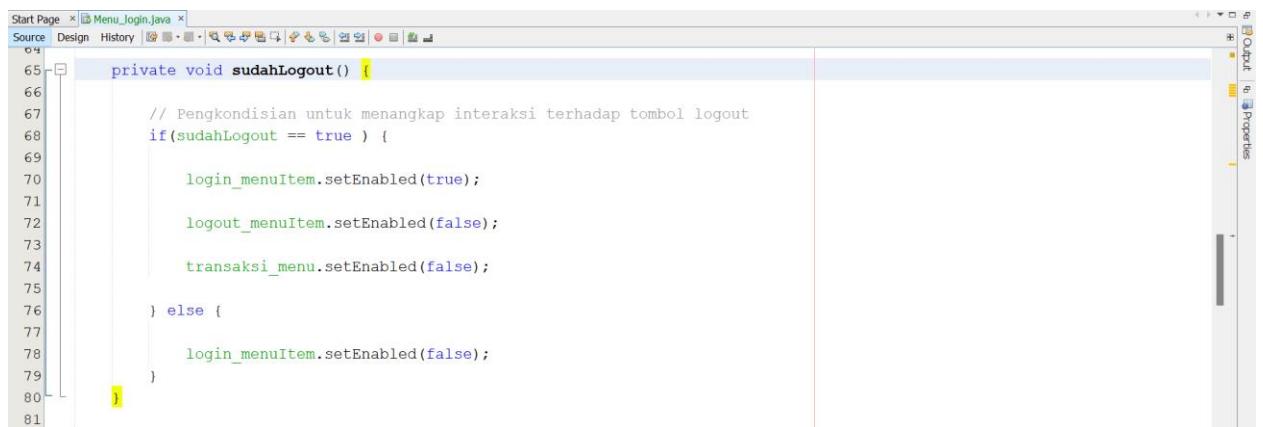
4. Buat fungsi sudahLogin() => Isi fungsi sesuai pada gambar 2.14.



```
Start Page x Menu_Login.java x
Source Design History
46  private void sudahLogin() {
47
48     // Pengkondision untuk menangkap interaksi terhadap tombol login
49     if(sudahLogin == true ) {
50
51         logout_menuItem.setEnabled(true);
52
53         transaksi_menu.setEnabled(true);
54
55         sudahLogout();
56     } else {
57
58         logout_menuItem.setEnabled(false);
59
60         transaksi_menu.setEnabled(false);
61     }
62 }
```

Gambar 2.14 : Fungsi sudahLogin() pada halaman Utama User Kasir

5. Buat fungsi SudahLogout() => Isi fungsi sesuai pada gambar 2.15.



```
Start Page x Menu_Login.java x
Source Design History
65  private void SudahLogout() {
66
67     // Pengkondision untuk menangkap interaksi terhadap tombol logout
68     if(sudahLogout == true ) {
69
70         login_menuItem.setEnabled(true);
71
72         logout_menuItem.setEnabled(false);
73
74         transaksi_menu.setEnabled(false);
75
76     } else {
77
78         login_menuItem.setEnabled(false);
79
80     }
81 }
```

Gambar 2.15 : Fungsi SudahLogout() pada halaman utama user kasir.

6. Buat fungsi jikaLoginUser() => Isi fungsi sesuai pada gambar 2.16.

```
Start Page × Menu_Login.java ×
Source Design History
83 public void jikaLoginUser() {
84
85     // Menunda proses login berjalan, sampai memasukkan username dan password di halaman Login_user
86     if(masukLogin == true) {
87
88         sudahLogin = true;
89
90         sudahLogin();
91
92         sudahLogin = false;
93
94     } else {
95
96         sudahLogin();
97
98     }
99 }
```

Gambar 2.16 : Fungsi jikaLoginUser() pada halaman utama user kasir

7. Buat Fungsi Kosong() => Isi fungsi sesuai pada gambar 2.17.

```
100
101 private void Kosong() {
102
103     transaksi_menu.setEnabled(false);
104 }
```

Gambar 2.17 : Fungsi Kosong() pada halaman utama user kasir

8. Buat fungsi KeyPressed() => Isi fungsi sesuai pada gambar 2.18.

```
Start Page × Menu_Login.java ×
Source Design History
107 private void keyPressed() {
108
109     User_menu.setMnemonic(KeyEvent.VK_U);
110
111     if(login_menuItem.isEnabled()) {
112
113         login_menuItem.setMnemonic(KeyEvent.VK_L);
114     } else if(logout_menuItem.isEnabled()) {
115
116         logout_menuItem.setMnemonic(KeyEvent.VK_L);
117     }
118 }
119 }
```

Gambar 2.18 : Fungsi KeyPressed() pada halaman utama user kasir

9. Buat fungsi event. Fungsi event ini digunakan agar element pada halaman jframe dapat berjalan sesuai dengan event yang anda tentukan. Untuk membuat event, anda bisa ke tab design => tekan kanan mouse pada elemen yang ingin kasih event => pilih menu event => pilih event yang anda inginkan. Event yang tersedia untuk elemen itu

bermacam – macam, sesuaikan dengan kebutuhan anda. Berikut beberapa contoh event yang digunakan pada halaman utama user kasir:



The screenshot shows a Java code editor with the file 'Menu_Login.java' open. The code contains two event handlers: 'User_menuSelected' and 'login_menuItemActionPerformed'. Both handlers have a placeholder comment // TODO add your handling code here. The 'login_menuItemActionPerformed' handler also includes code to set the visibility of a 'Menu_UserAdmin' window and dispose of the current window.

```
private void User_menuSelected(javax.swing.event.MenuEvent evt) {
    // TODO add your handling code here:
}

private void login_menuItemActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:

    new Menu_UserAdmin().setVisible(true);

    dispose();
}
```

Gambar 2.19 : Fungsi Event dan isinya pada halaman utama user kasir (1)



The screenshot shows a Java code editor with the file 'Menu_Login.java' open. It contains three event handlers: 'logout_menuItemActionPerformed', 'transaksi_menuActionPerformed', and 'Form_Transaksi_menuItemActionPerformed'. Each handler has a placeholder comment // TODO add your handling code here. The 'logout_menuItemActionPerformed' handler also includes code to set the value of a boolean variable 'sudahLogout' to true and call a method 'sudahLogout()'. The 'transaksi_menuActionPerformed' and 'Form_Transaksi_menuItemActionPerformed' handlers both include code to dispose of the current window and set the visibility of a 'Form_Transaksi' window to true.

```
private void logout_menuItemActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:

    sudahLogout = true;

    sudahLogout();

    sudahLogout = false;
}

private void transaksi_menuActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:

    dispose();
    new Form_Transaksi().setVisible(true);
}

private void Form_Transaksi_menuItemActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:

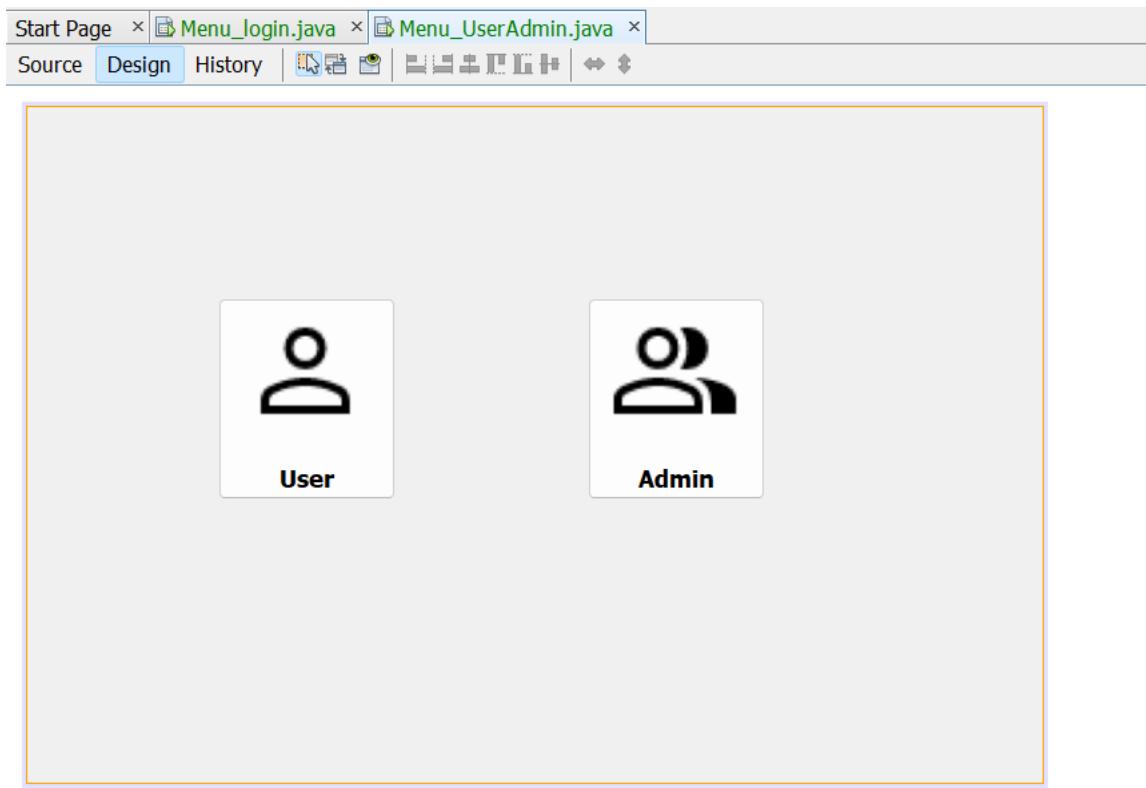
    dispose();
    new Form_Transaksi().setVisible(true);
}
```

Gambar 2.20 : Fungsi Event dan isinya pada halaman utama user kasir (2)

G. Pembuatan Halaman Menu User Admin

Halaman Menu User Admin ini adalah halaman perantara untuk menuju halaman login. Halaman utama user kasir ini cuma memiliki 2 menu, menu User dan menu Admin. Untuk membuat file jframe, anda bisa melihat pada ‘**Pembuatan Halaman Utama User Kasir**’.

Untuk desain yang ada di halaman menu user admin ini, anda harus membuat desain seperti gambar 2.21 ini.



Gambar 2.21 : Halaman Menu User Admin

Jika desain sudah dibuat, jangan lupa untuk mengganti nama variable tiap elemen, agar mempermudah untuk diingat. Setelah itu buka tab Soure pada halaman ini, lalu buat sistemnya agar halaman menu user admin ini dapat berjalan dengan baik. Berikut cara membuat sistem untuk halaman menu user admin:

1. Import beberapa library yang dibutuhkan.

A screenshot of a Java code editor. The title bar shows 'Start Page', 'Menu_login.java', and 'Menu_UserAdmin.java'. The code editor displays the following Java code:

```
1  /*
2   * To change this license header, choose License Headers in Project Properties.
3   * To change this template file, choose Tools | Templates
4   * and open the template in the editor.
5   */
6  package Menubar_login;
7
8  import java.awt.event.KeyEvent;
9
10 /**
11  *
12  * @author adham PSN
13 */
```

The code includes standard Java imports and a package declaration. It also contains a multi-line comment at the top and a single-line author annotation.

Gambar 2.22 : Library yang harus dimport pada halaman menu user admin

2. Pada constructor halaman ini, isi seperti gambar 2.23.

Gambar 2.23 : Isi dari constructor halaman menu user admin

3. Buat Fungsi keyPressed() => isi fungsi sesuai pada gambar 2.24

```
26
27     private void keyPressed() {
28
29         btn_user.setMnemonic(KeyEvent.VK_U);
30         btn_admin.setMnemonic(KeyEvent.VK_A);
31     }
32 }
```

Gambar 2.24 : Fungsi keyPressed() pada halaman menu user admin

4. Buat fungsi event. Fungsi event ini digunakan agar element pada halaman jframe dapat berjalan sesuai dengan event yang anda tentukan. Untuk membuat event, anda bisa ke tab design => tekan kanan mouse pada elemen yang ingin kasih event => pilih menu event => pilih event yang anda inginkan. Event yang tersedia untuk elemen itu bermacam – macam, sesuaikan dengan kebutuhan anda. Berikut beberapa contoh event yang digunakan pada halaman menu user admin:

Gambar 2.25 : Fungsi Event dan isinya pada halaman menu user admin(1)

H. Pembuatan Sistem Koneksi Database Pada Project Ini

Project yang dibuat ini pasti memerlukan database, yang mana anda harus menghubungkan terlebih dahulu ke database agar bisa mengakses data pada database. Berikut cara membuat sistem koneksi database:

1. Buat package baru dengan nama jaringan.
 2. Pada package jaringan, buat ‘Java Class’ baru dengan nama ‘Jaringan’.
 3. Import beberapa library yang dibutuhkan.

Gambar 2.26 : Library yang ada di class Jaringan

4. Buat variabel dengan class ‘Connection’

```
public Connection connect = null;
```

Gambar 2.27 : Variabel yang menggunakan class ‘Connection’ pada class jaringan

5. Buat konfigurasi pada function config().

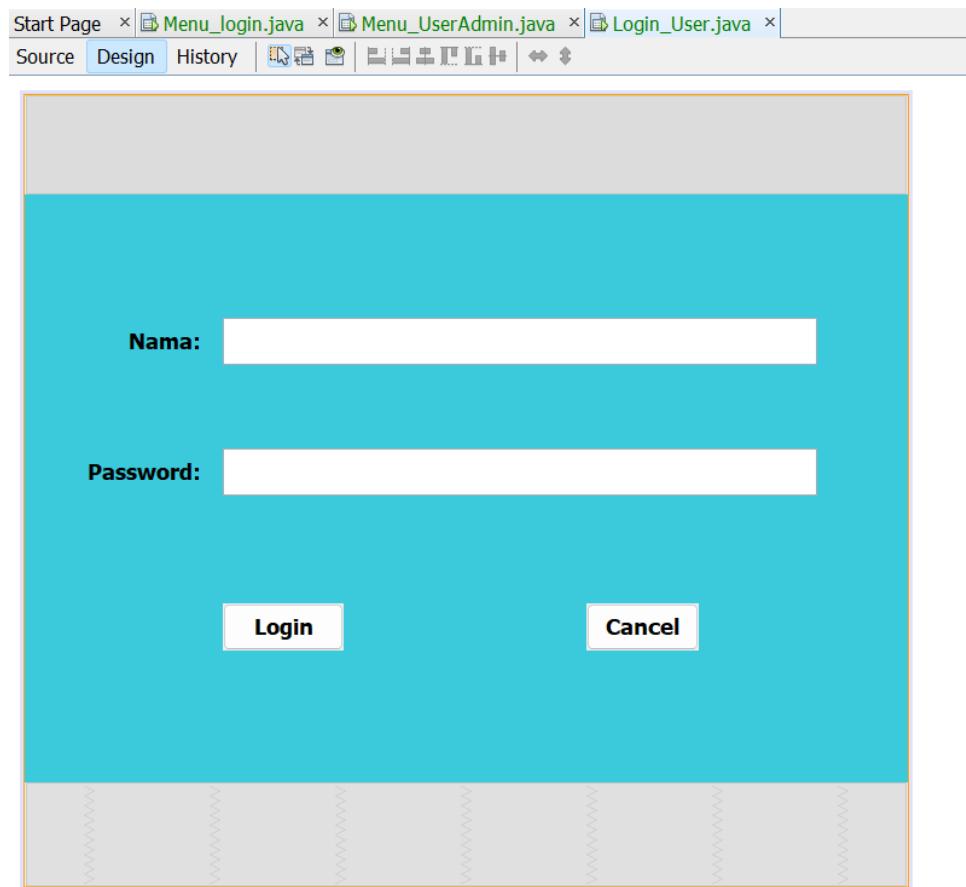
```
public static Connection config() {  
    try {  
        Class.forName("com.mysql.jdbc.Driver");  
        Connection connect = DriverManager.getConnection("jdbc:mysql://localhost/makmurjaya", "root", "");  
        return connect;  
    } catch(Exception e) {  
        JOptionPane.showMessageDialog(null, e);  
        return null;  
    }  
}
```

Gambar 2.28 : Konfigurasi koneksi database

Pada bagian getConnection(), ada 3 argument yang dimasukkan yang pertama database, kedua username, ketiga password.

I. Pembuatan Halaman Login

Halaman Login ini adalah halaman yang digunakan untuk masuk ke halaman utama user atau admin. Halaman utama user kasir ini cuma 2 kolom (kolom username dan password), dan ada juga tombol login atau cancel. Untuk membuat file jframe, anda bisa melihat pada ‘**Pembuatan Halaman Utama User Kasir**’. Untuk desain yang ada di halaman login ini, anda harus membuat desain seperti gambar 2.29 ini.

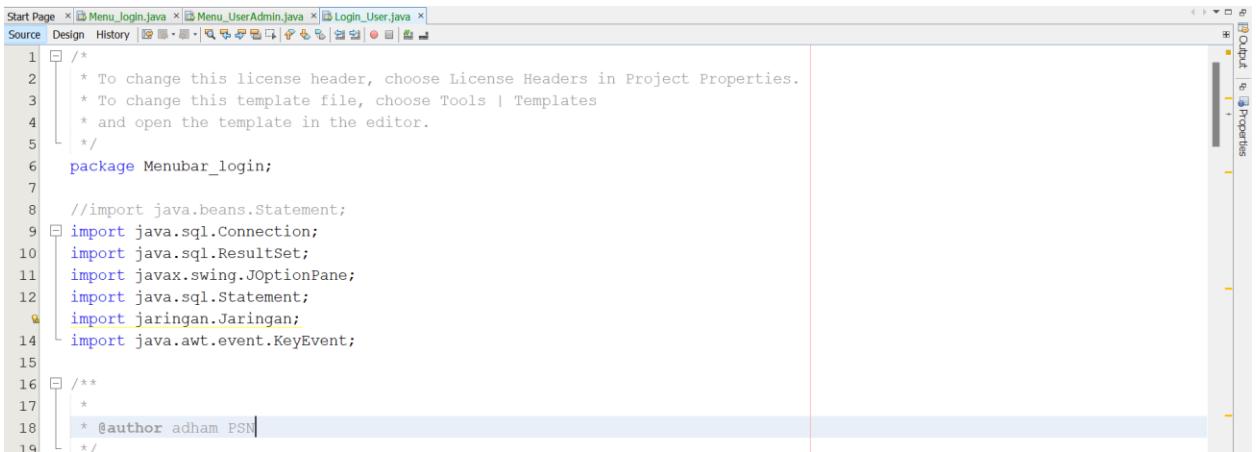


Gambar 2.29 : Halaman Login

Jika desain sudah dibuat, jangan lupa untuk mengganti nama variable tiap elemen, agar mempermudah untuk diingat. Setelah itu buka tab Soure pada halaman ini, lalu buat

sistemnya agar halaman menu login ini dapat berjalan dengan baik. Berikut cara membuat sistem untuk halaman login:

1. Mengimport library yang dibutuhkan



```
Start Page × Menu_login.java × Menu_UserAdmin.java × Login_User.java ×
Source Design History Tools - Help Properties

1  /*
2   * To change this license header, choose License Headers in Project Properties.
3   * To change this template file, choose Tools | Templates
4   * and open the template in the editor.
5   */
6  package Menubar_login;
7
8  //import java.beans.Statement;
9  import java.sql.Connection;
10 import java.sql.ResultSet;
11 import javax.swing.JOptionPane;
12 import java.sql.Statement;
13 import jaringan.Jaringan;
14 import java.awt.event.KeyEvent;
15
16 /**
17  *
18  * @author adham PSN
19 */
```

Gambar 2.30 : Library yang harus diimport pada halaman login

2. Membuat beberapa variabel scope global



```
21
22     Connection connect;
23     Statement stat;
24     ResultSet rs;
25     String sql;
26     public static int id;
27     boolean masukLoginTemp;
```

Gambar 2.31 : Beberapa variabel scope global yang ada di halaman login

3. Buat fungsi baru dengan nama login(), untuk mengkonfigurasi sistem login.



```
29
30     public void Login() {
31
32         try{
33
34             String username = txt_username.getText();
35             String password = txt_password.getText();
36             String sql = "SELECT * FROM tb_login WHERE username = '" + username+ "' AND password = '" + password + "'";
37
38             connect = jaringan.Jaringan.config();
39             stat = connect.createStatement();
40             rs = stat.executeQuery(sql);
41
42             if(rs.next()) {
43
44                 String akses = rs.getString("akses");
45                 System.out.println(akses);
46
47                 if(akses.equals("kasir")) {
48
49                     JOptionPane.showMessageDialog(null, "Berhasil Login");
50
51                     Menu_login user = new Menu_login();
52
53                     user.show();
54             }
55         }
56     }
57 }
```

Gambar 2.32 : Fungsi Login() pada halaman login (1)

```
55     Menu_login.Instance.masukLogin = masukLoginTemp;
56
57     user.jikaLoginUser();
58
59 } else if(akses.equals("admin")) {
60
61     JOptionPane.showMessageDialog(null, "Berhasil Login");
62
63     AdminToko admin = new AdminToko();
64     admin.show();
65
66 }
67 } else {
68     JOptionPane.showMessageDialog(null, "Login gagal !");
69     new Login_User().setVisible(true);
70 }
71
72 } catch (Exception e) {
73
74     System.out.println(e.getMessage());
75 }
76
77 }
```

Gambar 2.33 : Fungsi Login() pada halaman login (2)

4. Buat fungsi dengan nama getId().

```
Start Page | Menu_Login.java | Menu_Admin.java | Login_User.java | Properties
Source Design History Tools Window Help
78     public static int getId() {
79
80         return id;
81     }
82
83 }
```

Gambar 2.34 : Fungsi getId() pada halaman login

5. Buat fungsi dengan nama JikaAktif().

```
84
85     public void jikaAktif() {
86
87         if(jPanel1.isVisible()) {
88             this.masukLoginTemp = true;
89         }
90
91     }
92 }
```

Gambar 2.35 : Fungsi jikaAktif() pada halaman login

6. Buat fungsi dengan nama keyPressed().

```
93
94     private void keyPressed() {
95
96         btn_login.setMnemonic(KeyEvent.VK_L);
97         btn_login.setMnemonic(KeyEvent.VK_ENTER);
98
99         btn_cancel.setMnemonic(KeyEvent.VK_C);
100        btn_cancel.setMnemonic(KeyEvent.VK_BACK_SPACE);
101
102    }
103 }
```

Gambar 2.36 : fungsi keyPressed() pada halaman login

7. Constructor pada halaman login



```
103
104 	/** 
105 	* Creates new form Login_User
106  */
107 	public Login_User() {
108 		initComponents();
109 		setLocationRelativeTo(null);
110
111 		jikaAktif();
112 		keyPressed();
113 	}
114 }
```

Gambar 2.37 : Constructor pada halaman login

8. Buat fungsi event. Fungsi event ini digunakan agar element pada halaman jframe dapat berjalan sesuai dengan event yang anda tentukan. Untuk membuat event, anda bisa ke tab design => tekan kanan mouse pada elemen yang ingin kasih event => pilih menu event => pilih event yang anda inginkan. Event yang tersedia untuk elemen itu bermacam – macam, sesuaikan dengan kebutuhan anda. Berikut beberapa contoh event yang digunakan pada halaman login:



```
231 	private void btn_loginActionPerformed(java.awt.event.ActionEvent evt) {
232 // TODO add your handling code here:
233 	Login();
234 	dispose();
235 }
236
237 	private void btn_cancelActionPerformed(java.awt.event.ActionEvent evt) {
238 // TODO add your handling code here:
239 	dispose();
240 	new Menu_login().setVisible(true);
241 }
242 }
```

Gambar 2.38 : Fungsi Event dan isinya pada halaman login (1)

J. Pembuatan Halaman Transaksi

Halaman Transaksi ini adalah halaman yang digunakan oleh kasir untuk melakukan proses transaksi. Jadi halaman ini kasir mencatat proses transaksi yang dilakukan oleh pembeli, seperti barang apa saja yang dibeli, perhitungan total harga dari barang yang dibeli, perhitungan jumlah kembalian dari pembayaran suatu transaksi, dan pembuatan nota pembelian. Halaman utama user kasir ini memiliki banyak kolom dan tombol, yang mana semuanya memiliki fungsinya masing – masing. Untuk membuat file jframe, anda bisa melihat

pada ‘**Pembuatan Halaman Utama User Kasir**’. Untuk desain yang ada di halaman login ini, anda harus membuat desain seperti gambar 2.39 dan 2.40 ini.

The screenshot shows a Windows application window titled "Design Preview [Form_Transaksi]". The main title bar is yellow with the title "Transaksi". Below the title bar are three buttons: "Refresh", "Back", and "Keluar".

The interface is divided into several sections:

- Left Side:** A search section with "Cari Barang" and a search button. Below it is a table with columns: Kode Barang, Nama Barang, Harga, Stok, and Kategori. The table has multiple rows.
- Middle Left:** A "Daftar Pembelian" section containing a table with columns: Id, Nama, Harga, Jumlah, and Subtotal. This table also has multiple rows.
- Middle Right:** Input fields for transaction details: "Kode Transaksi", "Kode Barang", "Nama Barang", "Harga", "Jumlah", "Subtotal", and three buttons: "Reset", "Tambah", and "Edit".
- Bottom:** Payment section with "Cash" and "Kembalian" input fields, a "Bayar" button, and a "Cetak Bukti Pembayaran" button.

Gambar 2.39 : Halaman Form Transaksi (1)

Design Preview [Form_Transaksi]

Transaksi

Refresh Back Keluar

	<p>Nama Barang <input type="text"/></p> <p>Harga <input type="text"/></p> <p>Jumlah <input type="text"/></p> <p>Subtotal <input type="text"/></p> <p style="text-align: right;"><input type="button" value="Reset"/> <input type="button" value="Tambah"/> <input type="button" value="Edit"/></p>
--	--

Daftar Pembelian Cari Transaksi

Id	Nama	Harga	Jumlah	Subtotal

Total Tagihan

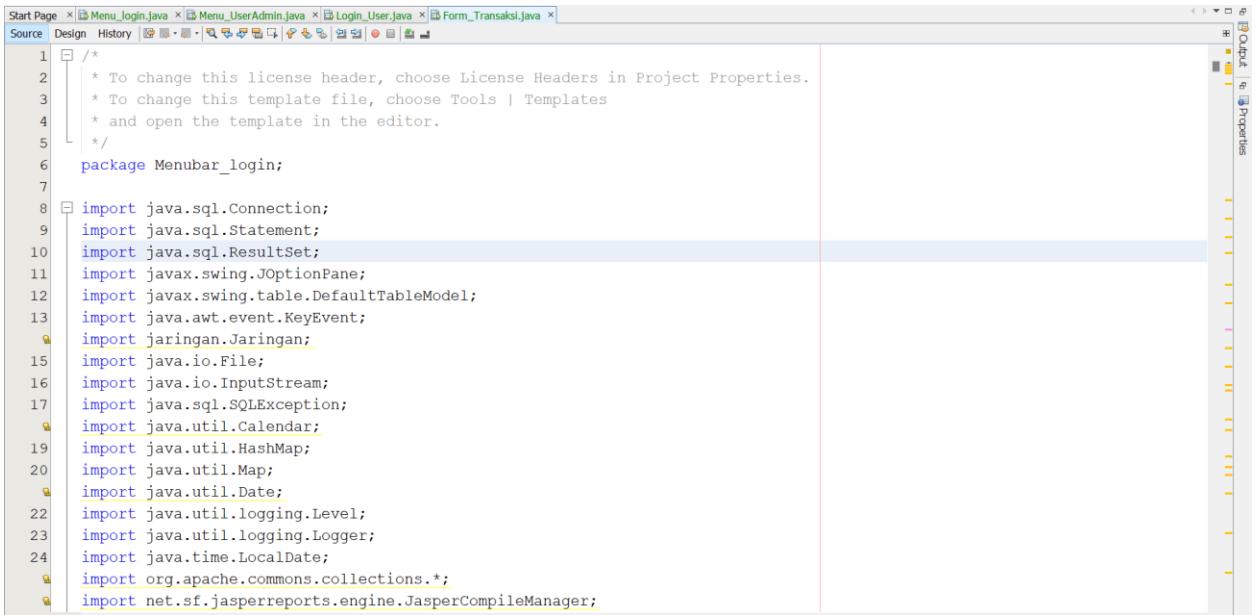
Cash

Kembalian

Gambar 2.40 : Halaman Form Transaksi (2)

Jika desain sudah dibuat, jangan lupa untuk mengganti nama variable tiap elemen, agar mempermudah untuk diingat. Setelah itu buka tab Soure pada halaman ini, lalu buat sistemnya agar halaman transaksi ini dapat berjalan dengan baik. Berikut cara membuat sistem untuk halaman transaksi:

1. Import library yang dibutuhkan untuk sistem transaksi.



```
Start Page | Menu_Login.java | Menu_UserAdmin.java | Login_User.java | Form_Transaksi.java | Source | Design | History | Output | Properties
```

```
1  /*
2  * To change this license header, choose License Headers in Project Properties.
3  * To change this template file, choose Tools | Templates
4  * and open the template in the editor.
5  */
6 package Menubar_login;
7
8 import java.sql.Connection;
9 import java.sql.Statement;
10 import java.sql.ResultSet;
11 import javax.swing.JOptionPane;
12 import javax.swing.table.DefaultTableModel;
13 import java.awt.event.KeyEvent;
14 import jaringan.Jaringan;
15 import java.io.File;
16 import java.io.InputStream;
17 import java.sql.SQLException;
18 import java.util.Calendar;
19 import java.util.HashMap;
20 import java.util.Map;
21 import java.util.Date;
22 import java.util.logging.Level;
23 import java.util.logging.Logger;
24 import java.time.LocalDate;
25 import org.apache.commons.collections.*;
26 import net.sf.jasperreports.engine.JasperCompileManager;
```

Gambar 2.41 : Library yang harus diimport pada halaman transaksi (1)



```
Start Page | Menu_Login.java | Menu_UserAdmin.java | Login_User.java | Form_Transaksi.java | Source | Design | History | Output | Properties
```

```
27 import net.sf.jasperreports.engine.JasperFillManager;
28 import net.sf.jasperreports.engine.JasperReport;
29 import net.sf.jasperreports.engine.JRDataSource;
30 import net.sf.jasperreports.view.JasperViewer;
31 import net.sf.jasperreports.engine.JasperPrint;
32 import net.sf.jasperreports.engine.design.JasperDesign;
33 import net.sf.jasperreports.engine.util.JRLoader;
34 import net.sf.jasperreports.engine.xml.JRXmlLoader;
35 import org.apache.commons.digester.Digester;
36 import org.apache.commons.logging.LogFactory;
37 import org.apache.commons.beanutils.MethodUtils;
38 import org.apache.poi.hssf.record.formula.functions.Today;
39
40 /**
41 * @author adham
42 */
43
```

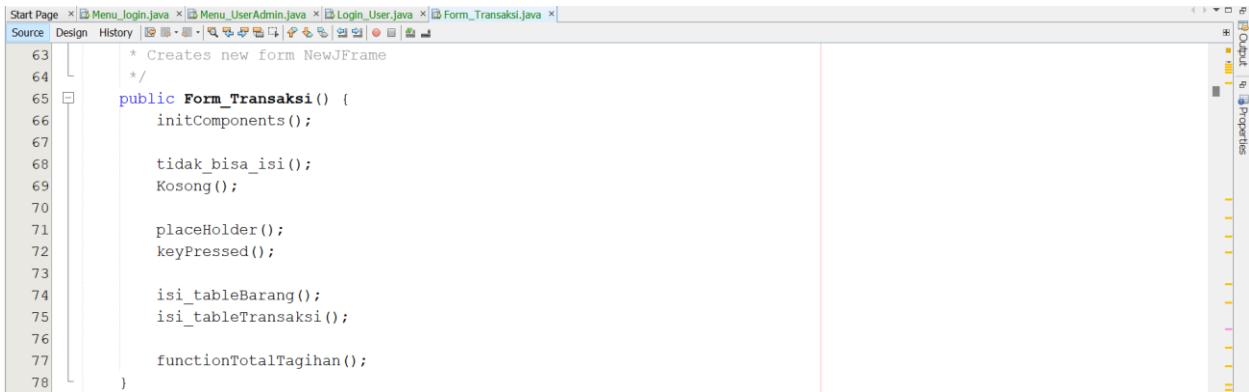
Gambar 2.42 : Library yang harus diimport pada halaman transaksi (2)

2. Membuat beberapa variabel scope global

```
46 //ireport
47 JasperReport JasRep;
48 JasperPrint JasPri;
49 Map param = new HashMap();
50 JasperDesign JasDes;
51 InputStream jasperStream;
52
53 private Connection koneksi;
54 private Statement pernyataan;
55 private ResultSet hasil;
56 private String sql;
57 private String sql2;
58 private String hasilPecahanChar;
59 private int cekTombol = 0;
60 private boolean masukLoginTemp;
```

Gambar 2.43 : Variabel scope global yang ada di sistem form transaksi

3. Isi Constructor pada halaman ini..

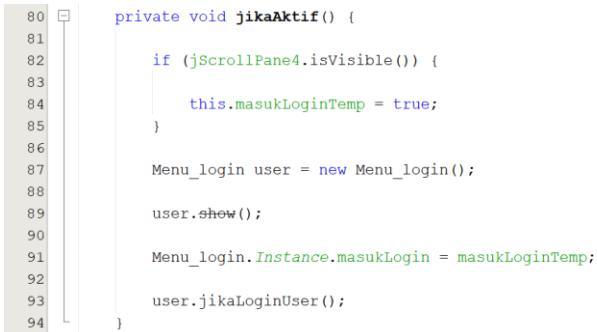


The screenshot shows a Java IDE interface with multiple tabs at the top: Start Page, Menu_login.java, Menu_UserAdmin.java, Login_User.java, and Form_Transaksi.java. The Form_Transaksi.java tab is active. The code in the editor is as follows:

```
63 * Creates new form NewJFrame
64 */
65 public Form_Transaksi() {
66     initComponents();
67
68     tidak_bisa_isi();
69     Kosong();
70
71     placeHolder();
72     keyPressed();
73
74     isi_tableBarang();
75     isi_tableTransaksi();
76
77     functionTotalTagihan();
78 }
```

Gambar 2.44 : Constructor pada halaman form transaksi

4. Buat fungsi jikaAktif, dan isi fungsi tersebut.

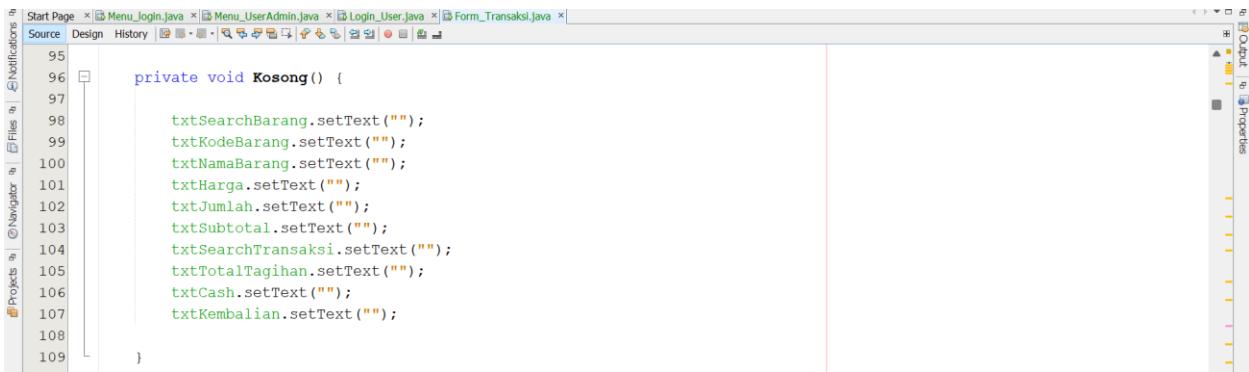


The screenshot shows a Java IDE interface with the Form_Transaksi.java tab active. The code in the editor is as follows:

```
80 private void jikaAktif() {
81
82     if (jScrollPane4.isVisible()) {
83
84         this.masukLoginTemp = true;
85     }
86
87     Menu_login user = new Menu_login();
88
89     user.show();
90
91     Menu_login.Instance.masukLogin = masukLoginTemp;
92
93     user.jikaLoginUser();
94 }
```

Gambar 2.45 : Fungsi jikaAktif() pada halaman form transaksi

5. Buat fungsi Kosong(), dan isi fungsi tersebut.

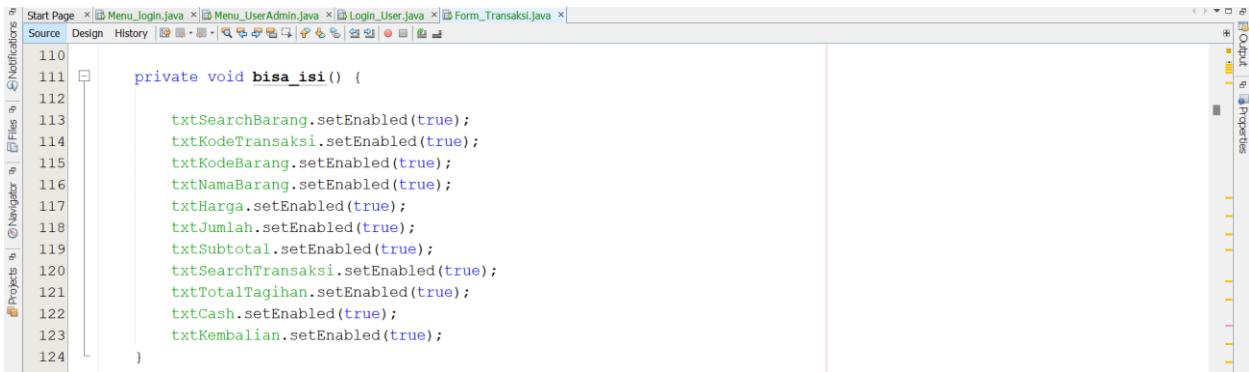


The screenshot shows the Eclipse IDE interface with the code editor open. The file is Form_Transaksi.java. The code defines a private void method named Kosong(). Inside the method, there is a block of code that sets various text fields to empty strings. The code is as follows:

```
private void Kosong() {  
    txtSearchBarang.setText("");  
    txtKodeBarang.setText("");  
    txtNamaBarang.setText("");  
    txtHarga.setText("");  
    txtJumlah.setText("");  
    txtSubtotal.setText("");  
    txtSearchTransaksi.setText("");  
    txtTotalTagihan.setText("");  
    txtCash.setText("");  
    txtKembalian.setText("");  
}
```

Gambar 2.46 : Fungsi Kosong() pada halaman form transaksi.

6. Buat fungsi bisa_isi(), dan isi fungsi tersebut.



The screenshot shows the Eclipse IDE interface with the code editor open. The file is Form_Transaksi.java. The code defines a private void method named bisa_isi(). Inside the method, there is a block of code that enables various text fields. The code is as follows:

```
private void bisa_isi() {  
    txtSearchBarang.setEnabled(true);  
    txtKodeTransaksi.setEnabled(true);  
    txtKodeBarang.setEnabled(true);  
    txtNamaBarang.setEnabled(true);  
    txtHarga.setEnabled(true);  
    txtJumlah.setEnabled(true);  
    txtSubtotal.setEnabled(true);  
    txtSearchTransaksi.setEnabled(true);  
    txtTotalTagihan.setEnabled(true);  
    txtCash.setEnabled(true);  
    txtKembalian.setEnabled(true);  
}
```

Gambar 2.47 : Fungsi bisa_isi() pada halaman form transaksi

7. Buat fungsi tidak_bisa_isi(), dan isi fungsi tersebut.

The screenshot shows the Eclipse IDE interface with the 'Form_Transaksi.java' file open in the source editor. The code snippet is as follows:

```
125  
126     private void tidak_bisa_isi() {  
127  
128         txtSearchBarang.setEnabled(true);  
129         txtKodeTransaksi.setEnabled(false);  
130         txtKodeBarang.setEnabled(false);  
131         txtNamaBarang.setEnabled(false);  
132         txtHarga.setEnabled(false);  
133         txtJumlah.setEnabled(true);  
134         txtSubtotal.setEnabled(false);  
135         txtSearchTransaksi.setEnabled(true);  
136         txtTotalTagihan.setEnabled(false);  
137         txtCash.setEnabled(true);  
138         txtKembalian.setEnabled(false);  
139     }
```

Gambar 2.48 : Fungsi tidak_bisa_isi() pada halaman transaksi

8. Buat fungsi tombol_mati(), dan isi fungsi tersebut.

The screenshot shows the Eclipse IDE interface with the 'Form_Transaksi.java' file open in the source editor. The code snippet is as follows:

```
140  
141     private void tombol_mati() {  
142  
143         btnBack.setEnabled(false);  
144         btnKeluar.setEnabled(false);  
145         btnReset.setEnabled(false);  
146         btnTambah.setEnabled(false);  
147         btnTambah.setText("Tambah");  
148         btnHapus.setEnabled(false);  
149         btnNewTransaksi.setEnabled(true);  
150         btnBayar.setEnabled(false);  
151         btnCetakBuktiPembayaran.setEnabled(false);  
152         btnRefresh.setEnabled(true);  
153     }
```

Gambar 2.49 : Fungsi tombol_mati() pada halaman form transaksi

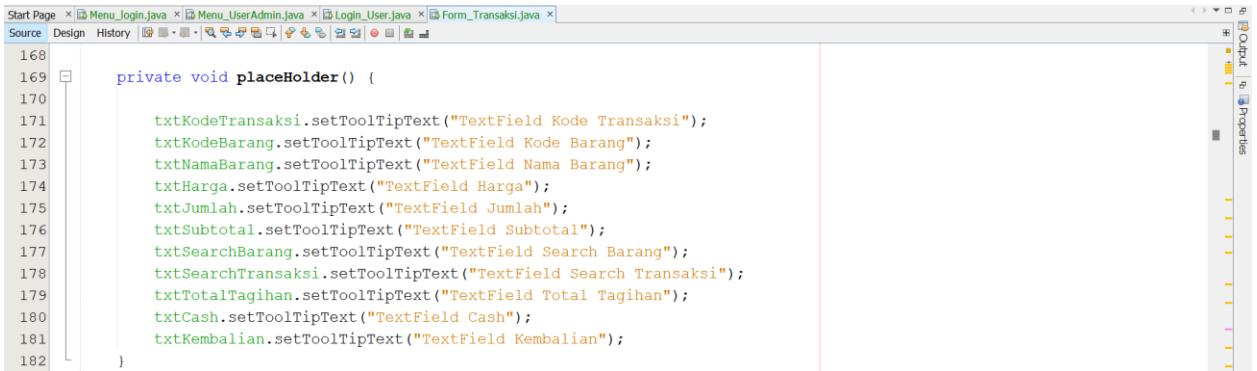
9. Buat fungsi tombol_hidup(), dan isi fungsi tersebut.

The screenshot shows the Eclipse IDE interface with the 'Form_Transaksi.java' file open in the source editor. The code snippet is as follows:

```
154  
155     private void tombol_hidup() {  
156  
157         btnBack.setEnabled(true);  
158         btnKeluar.setEnabled(true);  
159         btnReset.setEnabled(true);  
160         btnTambah.setEnabled(true);  
161         btnTambah.setText("Tambah");  
162         btnHapus.setEnabled(true);  
163         btnNewTransaksi.setEnabled(true);  
164         btnBayar.setEnabled(true);  
165         btnCetakBuktiPembayaran.setEnabled(true);  
166         btnRefresh.setEnabled(true);  
167     }
```

Gambar 2.50 : Fungsi tombol_hidup() pada halaman form transaksi

10. Buat fungsi placeholder(), dan isi fungsi tersebut.

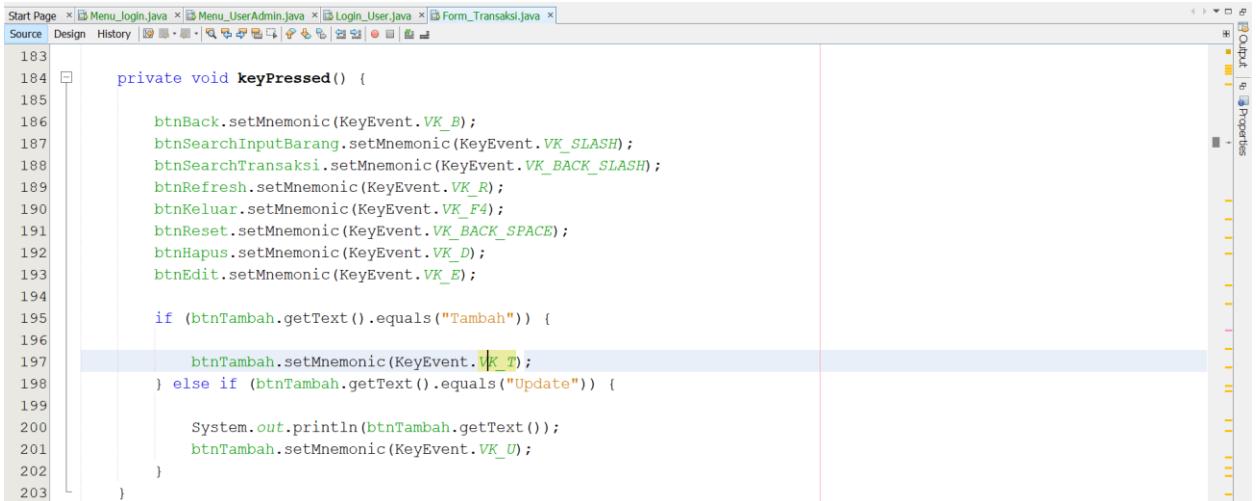


The screenshot shows the Eclipse IDE interface with the code editor open. The file being edited is Form_Transaksi.java. The code defines a private void method named placeHolder() which iterates through several JTextField components and sets their tool tip text. The code is as follows:

```
168  
169     private void placeHolder() {  
170         txtKodeTransaksi.setToolTipText("TextField Kode Transaksi");  
171         txtKodeBarang.setToolTipText("TextField Kode Barang");  
172         txtNamaBarang.setToolTipText("TextField Nama Barang");  
173         txtHarga.setToolTipText("TextField Harga");  
174         txtJumlah.setToolTipText("TextField Jumlah");  
175         txtSubtotal.setToolTipText("TextField Subtotal");  
176         txtSearchBarang.setToolTipText("TextField Search Barang");  
177         txtSearchTransaksi.setToolTipText("TextField Search Transaksi");  
178         txtTotalTagihan.setToolTipText("TextField Total Tagihan");  
179         txtCash.setToolTipText("TextField Cash");  
180         txtKembalian.setToolTipText("TextField Kembalian");  
181     }  
182 }
```

Gambar 2.51 : Fungsi placeholder() pada halaman form transaksi

11. Buat fungsi keyPressed(), dan isi fungsi tersebut.



The screenshot shows the Eclipse IDE interface with the code editor open. The file being edited is Form_Transaksi.java. The code defines a private void method named keyPressed() which sets mnemonic keys for various buttons and checks if the 'Tambah' button is pressed. The code is as follows:

```
183  
184     private void keyPressed() {  
185         btnBack.setMnemonic(KeyEvent.VK_B);  
186         btnSearchInputBarang.setMnemonic(KeyEvent.VK_SLASH);  
187         btnSearchTransaksi.setMnemonic(KeyEvent.VK_BACK_SLASH);  
188         btnRefresh.setMnemonic(KeyEvent.VK_R);  
189         btnKeluar.setMnemonic(KeyEvent.VK_F4);  
190         btnReset.setMnemonic(KeyEvent.VK_BACK_SPACE);  
191         btnHapus.setMnemonic(KeyEvent.VK_D);  
192         btnEdit.setMnemonic(KeyEvent.VK_E);  
193  
194         if (btnTambah.getText().equals("Tambah")) {  
195             btnTambah.setMnemonic(KeyEvent.VK_T);  
196         } else if (btnTambah.getText().equals("Update")) {  
197             System.out.println(btnTambah.getText());  
198             btnTambah.setMnemonic(KeyEvent.VK_U);  
199         }  
200     }  
201 }  
202 }  
203 }
```

Gambar 2.52 : Fungsi keyPressed() pada halaman form transaksi

12. Buat fungsi ambilDataTabelBarang(), dan isi fungsi tersebut.

The screenshot shows a Java IDE interface with multiple tabs at the top: Start Page, Menu_Login.java, Menu_UserAdmin.java, Login_User.java, and Form_Transaksi.java. The Form_Transaksi.java tab is active. The code editor displays the following Java code:

```
204  
205     private void ambilDataTabelBarang() {  
206  
207         // untuk mendisable editing column pada tabel  
208         tabelBarang.setDefaultEditor(Object.class, null);  
209  
210         int tabelData = tabelBarang.getSelectedRow();  
211  
212         txtKodeBarang.setText(tabelBarang.getValueAt(tabelData, 0).toString());  
213         txtNamaBarang.setText(tabelBarang.getValueAt(tabelData, 1).toString());  
214         txtHarga.setText(tabelBarang.getValueAt(tabelData, 2).toString());  
215  
216         tombol_hidup();  
217         ambilKodeTransaksiTerakhir();  
218         subtotalHargaBarang();  
219         btnTambah.setEnabled(true);  
220     }
```

Gambar 2.53 : Fungsi ambilDataTabelBarang() pada halaman form transaksi

13. Buat fungsi sistemTabelTransaksi(), dan isi fungsi tersebut.

The screenshot shows a Java IDE interface with multiple tabs at the top: Start Page, Menu_Login.java, Menu_UserAdmin.java, Login_User.java, and Form_Transaksi.java. The Form_Transaksi.java tab is active. The code editor displays the following Java code:

```
222     private void sistemTabelTransaksi() {  
223  
224         // untuk mendisable editing column pada tabel  
225         tabelTransaksi.setDefaultEditor(Object.class, null);  
226  
227         int tabelData = tabelTransaksi.getSelectedRow();  
228  
229         txtKodeTransaksi.setText(tabelTransaksi.getValueAt(tabelData, 0).toString());  
230         txtNamaBarang.setText(tabelTransaksi.getValueAt(tabelData, 1).toString());  
231         txtHarga.setText(tabelTransaksi.getValueAt(tabelData, 2).toString());  
232         txtJumlah.setText(tabelTransaksi.getValueAt(tabelData, 3).toString());  
233         txtSubtotal.setText(tabelTransaksi.getValueAt(tabelData, 4).toString());  
234  
235         tombol_hidup();  
236         ambilKodeBarangTerkait();  
237         btnTambah.setEnabled(true);  
238     }
```

Gambar 2.54 : Fungsi sistemTabelTransaksi() pada halaman form transaksi

14. Buat fungsi simpan(), dan isi fungsi tersebut.

The screenshot shows a Java code editor with multiple tabs at the top: Start Page, Menu_Login.java, Menu_UserAdmin.java, Login_User.java, and Form_Transaksi.java. The Form_Transaksi.java tab is active. The code is as follows:

```
240     private void Simpan() throws SQLException {
241
242         // Jika tidak ada data yang dimasukkan
243         if (txtKodeTransaksi.getText().equals("")) || txtKodeBarang.getText().equals("") || txtNamaBarang.getText().equals("")
244
245             JOptionPane.showMessageDialog(null, "Tidak ada data yang dimasukkan / Data yang dimasukkan tidak sesuai dengan format yang dibutuhkan! Silahkan memasukkan data lagi");
246
247         if (txtKodeTransaksi.getText().equals("")) {
248
249             try {
250                 sql = "SELECT MAX(kode_tr_angka) AS kode_transaksi_angka, id_transaksi"
251                     + " FROM tb_transaksi";
252
253                 koneksi = jaringan.Jaringan.config();
254                 pernyataan = koneksi.createStatement();
255                 hasil = pernyataan.executeQuery(sql);
256
257                 while (hasil.next()) {
258
259                     txtKodeTransaksi.setText(hasil.getString("id_transaksi"));
260                 }
261             } catch (SQLException e) {
262
263                 JOptionPane.showMessageDialog(null, "Keterangan Error :" + e);
264             }
265         }
266     }
```

Gambar 2.55 : Fungsi Simpan() pada halaman form transaksi (1)

The screenshot continues the Java code from the previous screenshot. The code is as follows:

```
240     private void Simpan() throws SQLException {
241
242         // Jika tidak ada data yang dimasukkan
243         if (txtKodeTransaksi.getText().equals("")) || txtKodeBarang.getText().equals("") || txtHarga.getText().equals("") || txtJumlah.getText().equals("") || txtSubtotal.getText().equals("")
244
245             JOptionPane.showMessageDialog(null, "Tidak ada data yang dimasukkan / Data yang dimasukkan tidak sesuai dengan format yang dibutuhkan! Silahkan memasukkan data lagi");
246
247
248         if (txtKodeTransaksi.getText().equals("")) {
249
250             try {
251                 sql = "SELECT MAX(kode_tr_angka) AS kode_transaksi_angka, id_transaksi"
252                     + " FROM tb_transaksi";
253
254                 koneksi = jaringan.Jaringan.config();
255                 pernyataan = koneksi.createStatement();
256                 hasil = pernyataan.executeQuery(sql);
257
258                 while (hasil.next()) {
259                     txtKodeTransaksi.setText(hasil.getString("id_transaksi"));
260                 }
261             } catch (SQLException e) {
262
263                 JOptionPane.showMessageDialog(null, "Keterangan Error :" + e);
264             }
265         }
266     }
```

Gambar 2.56 : Fungsi Simpan() pada halaman form transaksi (2)

A screenshot of a Java IDE interface. The title bar shows multiple tabs: Start Page, Menu_login.java, Menu_UserAdmin.java, Login_User.java, and Form_Transaksi.java. The main area displays Java code for the Form_Transaksi.java class. The code includes a check for empty fields and a message indicating that fields must be filled according to a specific format. Lines are numbered from 240 to 265.

```
240
241
242
243     xt().equals("") || txtHarga.getText().equals("") || txtJumlah.getText().equals("") || txtSubtotal.getText().equals("")) {
244     sesuai dengan format yang dibutuhkan! Silahkan memasukkan data lagi");
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
```

Gambar 2.57 : Fungsi Simpan() pada halaman form transaksi (3)

A screenshot of a Java IDE interface, continuing from the previous code block. The title bar shows the same tabs. The code block continues with the save function, including a try-catch block for database operations, a query to check if a product exists in a transaction, and logic to handle auto-increment values. Lines are numbered from 265 to 290.

```
265
266
267 } else {
268
269     try {
270
271         //Mengatur agar tidak langsung di commit ketika proses transaksi terjadi
272         koneksi = jaringan.Jaringan.config();
273         koneksi.setAutoCommit(false);
274
275         String nama_barang = txtNamaBarang.getText();
276         String transaksi_id = txtKodeTransaksi.getText();
277
278         sql = "SELECT *"
279             + " FROM tb_detail_transaksi"
280             + " WHERE nama LIKE '%" + nama_barang + "%' AND transaksi_id LIKE '%" + transaksi_id + "%'";
281
282         pernyataan = koneksi.createStatement();
283         hasil = pernyataan.executeQuery(sql);
284
285         // token
286         boolean valueAutoIncrement = hasil.next();
287
288         // Pengecekan sudah ada tidaknya barang dalam suatu transaksi
289         if (valueAutoIncrement == true) {
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
599
600
601
602
603
604
605
606
607
608
609
609
610
611
612
613
614
615
616
617
618
619
619
620
621
622
623
624
625
626
627
628
629
629
630
631
632
633
634
635
636
637
638
639
639
640
641
642
643
644
645
646
647
648
649
649
650
651
652
653
654
655
656
657
658
659
659
660
661
662
663
664
665
666
667
668
669
669
670
671
672
673
674
675
676
677
678
679
679
680
681
682
683
684
685
686
687
687
688
689
689
690
691
692
693
694
695
696
697
697
698
699
699
700
701
702
703
704
705
706
707
708
709
709
710
711
712
713
714
715
716
717
718
719
719
720
721
722
723
724
725
726
727
728
729
729
730
731
732
733
734
735
736
737
738
739
739
740
741
742
743
744
745
746
747
748
749
749
750
751
752
753
754
755
756
757
758
759
759
760
761
762
763
764
765
766
767
768
769
769
770
771
772
773
774
775
776
777
778
779
779
780
781
782
783
784
785
786
787
787
788
789
789
790
791
792
793
794
795
796
797
798
799
799
800
801
802
803
804
805
806
807
808
809
809
810
811
812
813
814
815
816
817
818
819
819
820
821
822
823
824
825
826
827
828
829
829
830
831
832
833
834
835
836
837
838
839
839
840
841
842
843
844
845
846
847
848
849
849
850
851
852
853
854
855
856
857
858
859
859
860
861
862
863
864
865
866
867
868
869
869
870
871
872
873
874
875
876
877
878
879
879
880
881
882
883
884
885
886
887
888
889
889
890
891
892
893
894
895
896
897
898
899
899
900
901
902
903
904
905
906
907
908
909
909
910
911
912
913
914
915
916
917
918
919
919
920
921
922
923
924
925
926
927
928
929
929
930
931
932
933
934
935
936
937
938
939
939
940
941
942
943
944
945
946
947
948
949
949
950
951
952
953
954
955
956
957
958
959
959
960
961
962
963
964
965
966
967
968
969
969
970
971
972
973
974
975
976
977
978
979
979
980
981
982
983
984
985
986
987
987
988
989
989
990
991
992
993
994
995
996
997
998
999
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1097
1098
1099
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1187
1188
1189
1189
1190
1191
1192
1193
1194
1195
1196
1197
1197
1198
1199
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1287
1288
1289
1289
1290
1291
1292
1293
1294
1295
1296
1297
1297
1298
1299
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1387
1388
1389
1389
1390
1391
1392
1393
1394
1395
1396
1397
1397
1398
1399
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1488
1489
1489
1490
1491
1492
1493
1494
1495
1496
1497
1497
1498
1499
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1588
1589
1589
1590
1591
1592
1593
1594
1595
1596
1597
1597
1598
1599
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1688
1689
1689
1690
1691
1692
1693
1694
1695
1696
1697
1697
1698
1699
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1788
1789
1789
1790
1791
1792
1793
1794
1795
1796
1797
1797
1798
1799
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1888
1889
1889
1890
1891
1892
1893
1894
1895
1896
1897
1897
1898
1899
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1988
1989
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2088
2089
2089
2090
2091
2092
2093
2094
2095
2096
2097
2097
2098
2099
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2188
2189
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2198
2199
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2259
2260
2261
2262
2263
2264
2264
2265
2266
2267
2268
2269
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2288
2289
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2298
2299
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2359
2360
2361
2362
2363
2364
2365
2366

```

Gambar 2.59 : Fungsi Simpan() pada halaman form transaksi (5)

```
291  
292  
293  
294  
295  
296     total) VALUES"  
297     txtNamaBarang.getText() + "", " + txtHarga.getText() + ", " + txtJumlah.getText() + ", " + txtSubtotal.getText() + ")";  
298  
299  
300  
301  
302  
303  
304  
305  
306  
307  
308  
309  
310  
311  
312  
313  
314  
315  
316
```

Gambar 2.60 : Fungsi Simpan() pada halaman form transaksi (6)

```

295     sql = "INSERT INTO tb_detail_transaksi(transaksi_id, barang_kode, nama, harga, jumlah, subtotal) VALUE
296         + "(" + txtKodeTransaksi.getText() + ", " + txtKodeBarang.getText() + ", " + txtNamaBaran
297
298     pernyataan = koneksi.createStatement();
299     pernyataan.execute(sql);
300 }
301
302 // commit jika sistem simpan barang berhasil dijalankan
303 koneksi.commit();
304 } catch (SQLException e) {
305
306     JOptionPane.showMessageDialog(null, "Keterangan Error :" + e);
307
308     // rollback autoincrement kode transaksi gagal dijalankan
309     koneksi.rollback();
310 }
311
312 tidak_bisa_isi();
313
314 isi_tableTransaksi();
315 isi_tableBarang();
316
317 }
318 }

```

Gambar 2.61 : Fungsi Simpan() pada halaman form transaksi (7)

15. Buat fungsi updateJumlah(), dan isi fungsi tersebut.

```

320 private void updateJumlah() {
321
322     // Jika tidak ada data yang dimasukkan
323     if (txtKodeTransaksi.getText().equals("") || txtNamaBarang.getText().equals("") || txtHarga.getText().equals("") ||
324
325         JOptionPane.showMessageDialog(null, "Tidak ada data yang dimasukkan / Data yang dimasukkan tidak sesuai dengan");
326
327     } else {
328
329         try {
330
331             sql = "UPDATE tb_detail_transaksi"
332                 + " SET jumlah = '" + txtJumlah.getText() + "', subtotal = '" + txtSubtotal.getText() + "'"
333                 + " WHERE transaksi_id LIKE '%" + txtKodeTransaksi.getText() + "%' AND nama LIKE '%" + txtNamaBara
334
335             koneksi = jaringan.Jaringan.config();
336             pernyataan = koneksi.createStatement();
337             pernyataan.executeUpdate(sql);
338
339             JOptionPane.showMessageDialog(null, "Update Jumlah Barang Berhasil");
340         } catch (Exception e) {
341
342             JOptionPane.showMessageDialog(null, "Keterangan Error :" + e);
343         }
344
345         isi_tableTransaksi();

```

Gambar 2.62 : Fungsi updateJumlah() pada halaman form transaksi (1)

The screenshot shows a Java IDE interface with multiple tabs at the top: Start Page, Menu_Login.java, Menu_UserAdmin.java, Login_User.java, and Form_Transaksi.java. The main area displays the following Java code:

```
320
321
322
323     equals("") || txtJumlah.getText().equals("") || txtSubtotal.getText().equals("")) {
324
325     sesuai dengan format yang dibutuhkan! Silahkan memasukkan data lagi");
326
327
328
329
330
331
332     + """
333     txtNamaBarang.getText() + "%";
334
335
336
337
338
339
340
341
342
343
344
345
```

Gambar 2.63 : Fungsi updateJumlah() pada halaman form transaksi (2)

The screenshot shows the continuation of the Java code from the previous screenshot. The code is enclosed in a try-catch block:

```
327     } else {
328
329         try {
330
331             sql = "UPDATE tb_detail_transaksi"
332                 + " SET jumlah = '" + txtJumlah.getText() + "', subtotal = '" + txtSubtotal.getText() + "'"
333                 + " WHERE transaksi_id LIKE '%" + txtKodeTransaksi.getText() + "%' AND nama LIKE '%"
334                 + txtNamaBarang.getText() + "%'";
335
336             koneksi = jaringan.Jaringan.config();
337             pernyataan = koneksi.createStatement();
338             pernyataan.executeUpdate(sql);
339
340             JOptionPane.showMessageDialog(null, "Update Jumlah Barang Berhasil");
341         } catch (Exception e) {
342
343             JOptionPane.showMessageDialog(null, "Keterangan Error :" + e);
344
345             isi_tableTransaksi();
346             isi_tableBarang();
347
348             btnEdit.setEnabled(true);
349
350         }
351     }
352 }
```

Gambar 2.64 : Fungsi updateJumlah() pada halaman form transaksi (3)

16. Buat fungsi hapus(), dengan isi fungsi tersebut.

```
352     private void hapus() {
353
354         // Jika tidak ada data yang dimasukkan
355         if (txtKodeTransaksi.getText().equals("") || txtNamaBarang.getText().equals("") || txtHarga.getText().equals("") ||
356
357             JOptionPane.showMessageDialog(null, "Tidak ada data yang dipilih! Silahkan memilih data lagi");
358
359     } else {
360
361         int dialogButton = JOptionPane.YES_NO_OPTION;
362         int dialogResult = JOptionPane.showConfirmDialog(this, "Data Akan di Hapus", "Hapus Data", dialogButton);
363
364         if (dialogResult == 0) {
365
366             try {
367
368                 sql = "DELETE"
369                     + " FROM tb_detail_transaksi"
370                     + " WHERE barang_kode LIKE '%" + txtKodeBarang.getText() + "%' AND transaksi_id LIKE '%" + txt
371
372                 koneksi = jaringan.Jaringan.config();
373                 pernyataan = koneksi.createStatement();
374                 pernyataan.execute(sql);
375
376                 JOptionPane.showMessageDialog(null, "Berhasil di Hapus");
377
378             }
379         }
380     }
381 }
```

Gambar 2. 65 : Fungsi hapus() pada halaman form transaksi (1)

Gambar 2.66 : Fungsi hapus() pada halaman form transaksi (2)

The screenshot shows a Java code editor with the file `Form_Transaksi.java` open. The code is part of a try-catch block. It constructs an SQL DELETE statement to remove a record from the `tb_detail_transaksi` table where the `barang_kode` matches the value in `txtKodeBarang.getText()` and the `transaksi_id` matches the value in `txtIdTransaksi.getText()`. It then creates a statement, executes it, and shows a success message dialog. If an exception occurs, it shows an error message dialog. Finally, it checks if the table is empty and displays a message dialog accordingly.

```
367  
368     sql = "DELETE"  
369         + " FROM tb_detail_transaksi"  
370         + " WHERE barang_kode LIKE '%" + txtKodeBarang.getText() + "%' AND transaksi_id LIKE '" + txtIdTransaksi.getText() + "'  
371  
372     koneksi = jaringan.Jaringan.config();  
373     pernyataan = koneksi.createStatement();  
374     pernyataan.execute(sql);  
375  
376     JOptionPane.showMessageDialog(null, "Berhasil di Hapus");  
377  
378     // kosong();  
379     isi_tableBarang();  
380     isi_tableTransaksi();  
381 } catch (Exception e) {  
382     JOptionPane.showMessageDialog(null, "Keterangan Error :" + e);  
383 } else {  
384     JOptionPane.showMessageDialog(null, "Batal di Hapus");  
385 }  
386 }  
387 }  
388 }  
389 }  
390 }  
391 }
```

Gambar 2.67 : Fungsi hapus() pada halaman form transaksi (3)

17. Buat fungsi `isi_tableTransaksi()`, dengan isi fungsi sesuai dengan gambar 2.67 dan 2.68.

The screenshot shows a Java code editor with the file `Form_Transaksi.java` open. The code defines a private method `isi_tableTransaksi()`. It first creates a `DefaultTableModel` named `tbl` and adds columns for `id_transaksi`, `nama`, `harga`, `jumlah`, and `subtotal`. It then sets this model to a table component `tblTransaksi`. Next, it tries to execute a SELECT query to retrieve data from the `tb_detail_transaksi` table based on the `transaksi_id` from the `tb_transaksi` table. The result is stored in a `ResultSet` named `hasil`. Finally, it loops through the results, creating a new row in the table model for each record found.

```
392     private void isi_tableTransaksi() {  
393  
394         DefaultTableModel tbl = new DefaultTableModel();  
395         tbl.addColumn("id_transaksi");  
396         tbl.addColumn("nama");  
397         tbl.addColumn("harga");  
398         tbl.addColumn("jumlah");  
399         tbl.addColumn("subtotal");  
400         tabelTransaksi.setModel(tbl);  
401  
402         try {  
403             sql = "SELECT *"  
404                 + " FROM tb_detail_transaksi"  
405                 + " WHERE transaksi_id = (SELECT MAX(id_transaksi) AS id_transaksi"  
406                 + " FROM tb_transaksi)";  
407  
408             koneksi = jaringan.Jaringan.config();  
409             pernyataan = koneksi.createStatement();  
410             hasil = pernyataan.executeQuery(sql);  
411  
412             while (hasil.next()) {  
413                 tbl.addRow(new Object[]{  
414                     hasil.getString("transaksi_id"),  
415                     hasil.getString("nama"),  
416                     hasil.getString("harga"),  
417                     hasil.getString("jumlah"),  
418                     hasil.getString("subtotal")});  
419             }  
420         } catch (Exception e) {  
421             JOptionPane.showMessageDialog(null, "Keterangan Error :" + e);  
422         }  
423     }  
424 }
```

Gambar 2.68 : Fungsi `isi_tableTransaksi()` pada halaman form transaksi (1)

The screenshot shows a Java code editor with multiple tabs at the top: Start Page, Menu_login.java, Menu_UserAdmin.java, Login_User.java, and Form_Transaksi.java. The code in the editor is as follows:

```
405     + " FROM tb_detail_transaksi"
406     + " WHERE transaksi_id = (SELECT MAX(id_transaksi) AS id_transaksi"
407     + " FROM tb_transaksi)";
408
409     koneksi = jaringan.Jaringan.config();
410     pernyataan = koneksi.createStatement();
411     hasil = pernyataan.executeQuery(sql);
412
413     while (hasil.next()) {
414
415         tbl.addRow(new Object[]{
416             hasil.getString("transaksi_id"),
417             hasil.getString("nama"),
418             hasil.getString("harga"),
419             hasil.getString("jumlah"),
420             hasil.getString("subtotal"),
421         });
422
423         tabelTransaksi.setModel(tbl);
424     }
425 } catch (Exception e) {
426 }
427
428 }
429 }
```

Gambar 2.69 : Fungsi isi_tableTransaksi() pada halaman form transaksi (2)

18. Buat fungsi isi_tableBarang(), dengan isi fungsi tersebut.

The screenshot shows a Java code editor with multiple tabs at the top: Start Page, Menu_login.java, Menu_UserAdmin.java, Login_User.java, and Form_Transaksi.java. The code in the editor is as follows:

```
430 private void isi_tableBarang() {
431
432     DefaultTableModel tbl = new DefaultTableModel();
433     tbl.addColumn("kode");
434     tbl.addColumn("nama");
435     tbl.addColumn("harga");
436     tbl.addColumn("stok");
437     tbl.addColumn("kategori");
438     tabelBarang.setModel(tbl);
439
440     try {
441
442         sql = "SELECT *"
443         + " FROM tb_barang";
444
445         koneksi = jaringan.Jaringan.config();
446         pernyataan = koneksi.createStatement();
447         hasil = pernyataan.executeQuery(sql);
448
449         while (hasil.next()) {
450
451             tbl.addRow(new Object[]{
452                 hasil.getString("kode"),
453                 hasil.getString("nama"),
454                 hasil.getString("harga"),
455                 hasil.getString("stok"),
456             });
457         }
458     }
459 }
```

Gambar 2. 70 : Fungsi isi_tableBarang() pada halaman form transaksi (1)

```
Start Page × Menu_Login.java × Menu_UserAdmin.java × Login_User.java × Form_Transaksi.java ×
Source Design History ☰ Properties

442         sql = "SELECT *"
443             + " FROM tb_barang";
444
445     koneksi = jaringan.Jaringan.config();
446     pernyataan = koneksi.createStatement();
447     hasil = pernyataan.executeQuery(sql);
448
449     while (hasil.next()) {
450
451         tb1.addRow(new Object[]{
452             hasil.getString("kode"),
453             hasil.getString("nama"),
454             hasil.getString("harga"),
455             hasil.getString("stok"),
456             hasil.getString("kategori"),});
457     }
458
459     tabelBarang.setModel(tb1);
460
461 } catch (Exception e) {
462 }
463
464
465 }
```

Gambar 2. 71 : Fungsi isi_tableBarang() pada halaman form transaksi (2)

19. Buat fungsi AutoIncrement(), dan isi fungsi tersebut.

Gambar 2. 72 : Fungsi AutoIncrement() pada halaman form transaksi (1)

The screenshot shows a Java code editor with several tabs at the top: Start Page, Menu_Login.java, Menu_UserAdmin.java, Login_User.java, and Form_Transaksi.java. The code in the editor is as follows:

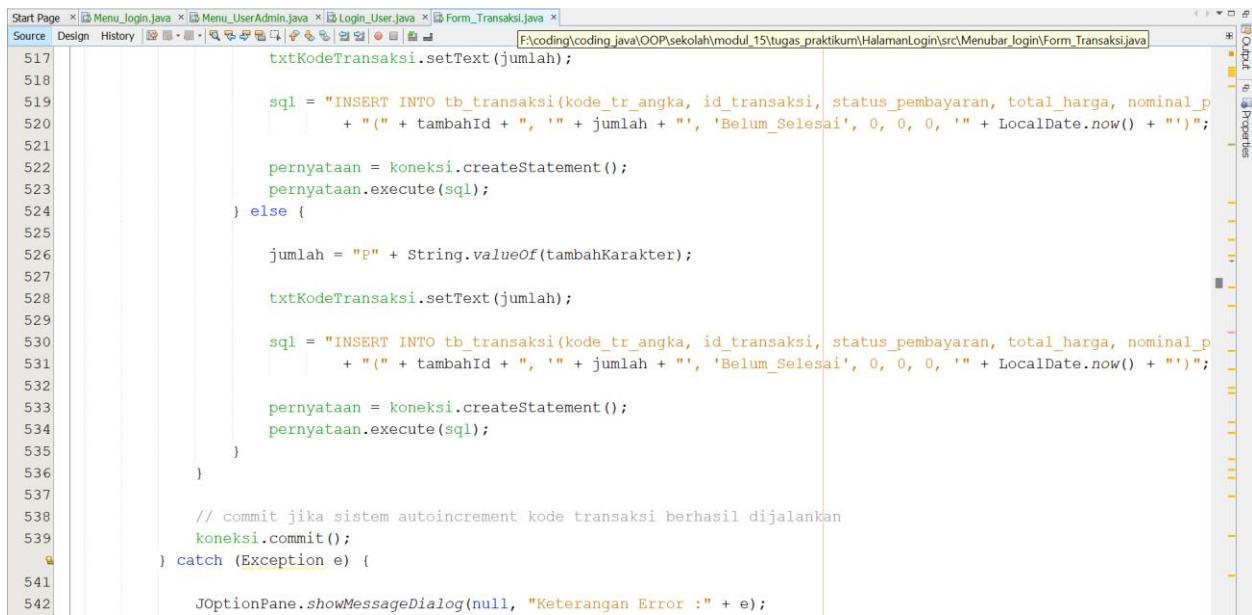
```
492 jumlah = "P" + 0 + id_transaksi;
493 txtKodeTransaksi.setText(jumlah);
494
495 sql = "INSERT INTO tb_transaksi(kode_tr_angka, id_transaksi, status_pembayaran, total_harga, nominal_pembayaran) VALUES"
496     + "(" + id_transaksi + ", '" + jumlah + "', 'Belum_Selesai', 0, 0, 0, '" + LocalDate.now() + "')";
497 pernyataan = koneksi.createStatement();
498 pernyataan.execute(sql);
499 } else {
500
501     // hasil dari query sql
502     hasilHitung = hasil.getInt("kode_transaksi_angka");
503     kodeTransaksi = hitungKarakter(hasil.getString("id_transaksi"));
504
505     tambahId = hasilHitung + 1;
506     tambahKarakter = Integer.parseInt(kodeTransaksi) + 1;
507
508     // pengkondisian berdasarkan digit angka
509     if (hitungDigitAngka(tambahKarakter) == 1) {
510
511         tambahKarakter1 = "0" + Integer.toString(tambahKarakter);
512
513         jumlah = "P" + tambahKarakter1;
514
515         txtKodeTransaksi.setText(jumlah);
516
517 }
```

Gambar 2.73 : Fungsi AutoIncrement() pada halaman form transaksi (2)

The screenshot shows the same Java code editor with the same tabs. The code has been modified to show the final SQL statement:

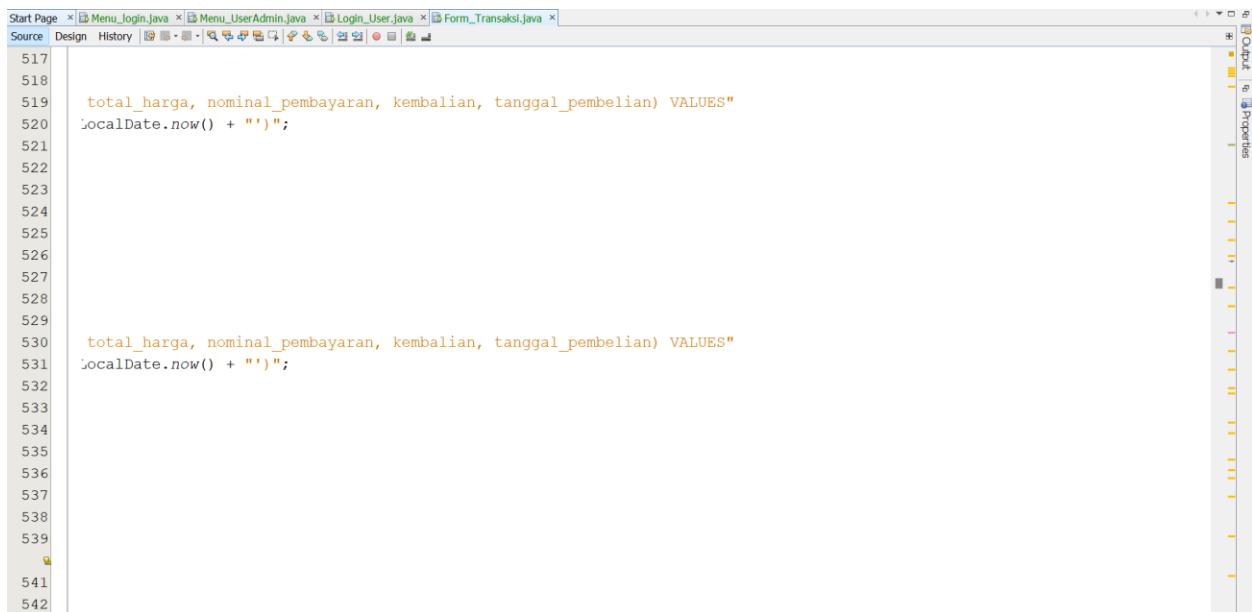
```
492
493
494
495
496     total_harga, nominal_pembayaran, kembalian, tanggal_pembelian) VALUES"
497     + LocalDate.now() + "';";
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
```

Gambar 2.74 : Fungsi AutoIncrement() pada halaman form transaksi (3)



```
517     txtKodeTransaksi.setText(jumlah);
518
519     sql = "INSERT INTO tb_transaksi(kode_tr_angka, id_transaksi, status_pembayaran, total_harga, nominal_p
520         + "(" + tambahId + ", '" + jumlah + "', 'Belum_Selesai', 0, 0, 0, '" + LocalDate.now() + "')";
521
522     pernyataan = koneksi.createStatement();
523     pernyataan.execute(sql);
524 } else {
525
526     jumlah = "P" + String.valueOf(tambahKarakter);
527
528     txtKodeTransaksi.setText(jumlah);
529
530     sql = "INSERT INTO tb_transaksi(kode_tr_angka, id_transaksi, status_pembayaran, total_harga, nominal_p
531         + "(" + tambahId + ", '" + jumlah + "', 'Belum_Selesai', 0, 0, 0, '" + LocalDate.now() + "')";
532
533     pernyataan = koneksi.createStatement();
534     pernyataan.execute(sql);
535 }
536
537 // commit jika sistem autoincrement kode transaksi berhasil dijalankan
538 koneksi.commit();
539 } catch (Exception e) {
540
541     JOptionPane.showMessageDialog(null, "Keterangan Error :" + e);
542 }
```

Gambar 2.75 : Fungsi AutoIncrement() pada halaman form transaksi (4)



```
517     total_harga, nominal_pembayaran, kembalian, tanggal_pembelian) VALUES"
518     .localDate.now() + ")";
519
520
521
522
523
524
525
526
527
528
529
530     total_harga, nominal_pembayaran, kembalian, tanggal_pembelian) VALUES"
531     .localDate.now() + ")";
532
533
534
535
536
537
538
539
540
541
542
```

Gambar 2.76 : Fungsi AutoIncrement() pada halaman form transaksi (5)

The screenshot shows a Java code editor with several tabs at the top: Start Page, Menu_login.java, Menu_UserAdmin.java, Login_User.java, and Form_Transaksi.java. The code in the editor is as follows:

```
527     txtKodeTransaksi.setText(jumlah);
528
529     sql = "INSERT INTO tb_transaksi(kode_tr_angka, id_transaksi, status_pembayaran, total_harga, nominal_p_";
530     sql += " + tambahId + ", " + jumlah + ', 'Belum Selesai', 0, 0, " + LocalDate.now() + ')';
531
532     pernyataan = koneksi.createStatement();
533     pernyataan.execute(sql);
534
535 }
536
537 // commit jika sistem autoincrement kode transaksi berhasil dijalankan
538 koneksi.commit();
539 } catch (Exception e) {
540     JOptionPane.showMessageDialog(null, "Keterangan Error :" + e);
541
542     // rollback autoincrement kode transaksi gagal dijalankan
543     koneksi.rollback();
544 }
545
546 }
547
548 }
```

Gambar 2.77 : Fungsi AutoIncrement() pada halaman form transaksi (6)

20. Buat fungsi hitungDigitAngka(), dan isi fungsi tersebut.

The screenshot shows a Java code editor with several tabs at the top: Start Page, Menu_login.java, Menu_UserAdmin.java, Login_User.java, and Form_Transaksi.java. The code in the editor is as follows:

```
549 private int hitungDigitAngka(int angka) {
550
551     int count = 0;
552
553     while (angka != 0) {
554
555         angka /= 10;
556
557         count++;
558     }
559
560     return count;
561
562 }
563
```

Gambar 2.78 : Fungsi hitungDigitAngka() pada halaman form transaksi

21. Buat fungsi hitungKarakter(), dengan isi fungsi tersebut.

A screenshot of a Java IDE showing the code for the `hitungKarakter` method. The code uses multiple local variables (`karakter1` through `karakter12`) to store characters from the input string. It then checks the length of the string to determine which characters to use based on the number of digits. The first few lines of the code are:

```
564 // Digunakan untuk memisahkan angka dengan huruf
565 private String hitungKarakter(String karakterTemp) {
566
567     String karakter;
568     String karakter2;
569     String karakter3;
570     String karakter4;
571     String karakter5;
572     String karakter6;
573     String karakter7;
574     String karakter8;
575     String karakter9;
576     String karakter10;
577     String karakter11;
578     String karakter12;
579
580     String str = new String(karakterTemp);
581
582     if (str.length() == 3) {
583
584         // Digit angka puluhan
585         karakter2 = String.valueOf(str.charAt(str.length() - 2));
586
587         // Digit angka satuan
588         karakter = String.valueOf(str.charAt(str.length() - 1));
589     }
```

Gambar 2.79 : Fungsi hitungKarakter() pada halaman form transaksi (1)

A screenshot of a Java IDE showing the continuation of the `hitungKarakter` method. The code handles cases for strings of length 4, 5, and 6 by concatenating characters from the input string into local variables (`karakter3`, `karakter2`, `karakter`, `karakter4`, `karakter3`, `karakter2`). The code is as follows:

```
590 // Penggabungan karakter 1 dan 2
591 this.hasilPecahanChar = karakter2 + karakter;
592
593 } else if (str.length() == 4) {
594
595     // Digit angka ratusan
596     karakter3 = String.valueOf(str.charAt(str.length() - 3));
597
598     // Digit angka puluhan
599     karakter2 = String.valueOf(str.charAt(str.length() - 2));
600
601     // Digit angka satuan
602     karakter = String.valueOf(str.charAt(str.length() - 1));
603
604     // Penggabungan karakter 1, 2, 3
605     this.hasilPecahanChar = karakter3 + karakter2 + karakter;
606 } else if (str.length() == 5) {
607
608     // Digit angka ribuan
609     karakter4 = String.valueOf(str.charAt(str.length() - 4));
610
611     // Digit angka ratusan
612     karakter3 = String.valueOf(str.charAt(str.length() - 3));
613
614     // Digit angka puluhan
615     karakter2 = String.valueOf(str.charAt(str.length() - 2));
```

Gambar 2.80 : Fungsi hitungKarakter() pada halaman form transaksi (2)

The screenshot shows a Java code editor with several tabs at the top: Start Page, Menu_login.java, Menu_UserAdmin.java, Login_User.java, and Form_Transaksi.java. The current tab is Form_Transaksi.java. The code is part of a class and shows logic for concatenating characters based on their position in a string. It handles cases for lengths of 1, 2, 3, 4, and 6 characters. The code uses String.valueOf() to convert integer indices to character positions.

```
616  
617     // Digit angka satuan  
618     karakter = String.valueOf(str.charAt(str.length() - 1));  
619  
620     // Penggabungan karakter 1, 2, 3, 4  
621     this.hasilPecahanChar = karakter4 + karakter3 + karakter2 + karakter;  
622 } else if (str.length() == 6) {  
623  
624     // Digit angka puluhan ribu  
625     karakter5 = String.valueOf(str.charAt(str.length() - 5));  
626  
627     // Digit angka ribuan  
628     karakter4 = String.valueOf(str.charAt(str.length() - 4));  
629  
630     // Digit angka ratusan  
631     karakter3 = String.valueOf(str.charAt(str.length() - 3));  
632  
633     // Digit angka puluhan  
634     karakter2 = String.valueOf(str.charAt(str.length() - 2));  
635  
636     // Digit angka satuan  
637     karakter = String.valueOf(str.charAt(str.length() - 1));  
638  
639     // Penggabungan karakter 1, 2, 3, 4, 5  
640     this.hasilPecahanChar = karakter5 + karakter4 + karakter3 + karakter2 + karakter;  
641 }  
642  
643     // Penggabungan karakter 1, 2, 3, 4, 5  
644     this.hasilPecahanChar = karakter5 + karakter4 + karakter3 + karakter2 + karakter;  
645 }  
646  
647     return hasilPecahanChar;  
648 }
```

Gambar 2.81 : Fungsi hitungKarangter() pada halaman form transaksi (3)

This screenshot shows the final part of the calculate function from the previous image. It concatenates the fifth character and returns the result.

```
639  
640     // Penggabungan karakter 1, 2, 3, 4, 5  
641     this.hasilPecahanChar = karakter5 + karakter4 + karakter3 + karakter2 + karakter;  
642 }  
643  
644     return hasilPecahanChar;  
645 }
```

Gambar 2.82 : Fungsi hitungKarangter() pada halaman form transaksi (4)

22. Buat fungsi searchBarang(), dengan isi fungsi sesuai dengan gambar 2.82 sampai 2.84.

This screenshot shows the implementation of the searchBarang() function. It checks if a search term is provided in a JTextField. If not, it focuses on the field. Otherwise, it creates a DefaultTableModel for a table and sets it as the model for a JTable. It then constructs a SQL query to search for items by code or name and executes it using a PreparedStatement.

```
646 private void searchBarang() {  
647  
648     // Jika TextField search kosong  
649     if (txtSearchBarang.getText().equals("")) {  
650         txtSearchBarang.requestFocus();  
651     } else {  
652  
653         DefaultTableModel tbl = new DefaultTableModel();  
654         tbl.addColumn("kode");  
655         tbl.addColumn("nama");  
656         tbl.addColumn("harga");  
657         tbl.addColumn("stok");  
658         tbl.addColumn("kategori");  
659         tabelBarang.setModel(tbl);  
660  
661         try {  
662             sql = "SELECT *"  
663                 + " FROM tb_barang"  
664                 + " WHERE kode = '" + txtSearchBarang.getText() + "' || nama LIKE '%" + txtSearchBarang.getText()  
665  
666             koneksi = jaringan.Jaringan.config();  
667             pernyataan = koneksi.createStatement();  
668             hasil = pernyataan.executeQuery(sql);  
669  
670         } catch (Exception e) {  
671             System.out.println(e);  
672         }  
673     }  
674 }
```

Gambar 2.83 : Fungsi searchBarang() pada halaman form transaksi (1)

A screenshot of a Java IDE interface. The top menu bar includes 'Start Page', 'Menu_login.java', 'Menu_UserAdmin.java', 'Login_User.java', and 'Form_Transaksi.java'. The main area shows Java code with line numbers from 646 to 671. The code includes a search condition for a category field:

```
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667 + txtSearchBarang.getText() + "%" || kategori LIKE "%" + txtSearchBarang.getText() + "%";
668
669
670
671
```

Gambar 2.84 : Fungsi searchBarang() pada halaman form transaksi (2)

A screenshot of a Java IDE interface. The top menu bar includes 'Start Page', 'Menu_login.java', 'Menu_UserAdmin.java', 'Login_User.java', and 'Form_Transaksi.java'. The main area shows Java code with line numbers from 667 to 690. The code constructs an SQL query for a database search:

```
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
```

```
+ " WHERE kode = '" + txtSearchBarang.getText() + "' || nama LIKE '%" + txtSearchBarang.getText()
koneksi = jaringan.Jaringan.config();
pernyataan = koneksi.createStatement();
hasil = pernyataan.executeQuery(sql);

while (hasil.next()) {
    tb1.addRow(new Object[]{
        hasil.getString("kode"),
        hasil.getString("nama"),
        hasil.getString("harga"),
        hasil.getString("stok"),
        hasil.getString("kategori"),});
}

tabelBarang.setModel(tb1);

// txtKode.setEnabled(true);
} catch (Exception e) {
}
}
```

Gambar 2.85 : Fungsi searchBarang() pada halaman form transaksi (2)

23. Buat fungsi SearchTransaksi(), dengan isi fungsi tersebut.

Gambar 2.86 : Fungsi searchTransaksi() pada halaman form transaksi (1)

```
Start Page x Menu_login.java x Menu_UserAdmin.java x Login_User.java x Form_Transaksi.java x
Source Design History [File] [Edit] [View] [Format] [Tools] [Help]
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713 || nama LIKE '%" + txtSearchTransaksi.getText() + "%' AND transaksi_id LIKE '%" + txtKodeTransaksi.getText() + "%'";
714
715
716
```

Gambar 2.87 : Fungsi searchTransaksi() pada halaman form transaksi (2)

```
Source Design History | 709 try { 710     711         sql = "SELECT * 712             + " FROM tb_detail_transaksi" 713             + " WHERE subtotal = '" + txtSearchTransaksi.getText() + "' || nama LIKE '%" + txtSearchTransaksi. 714     715         koneksi = jaringan.Jaringan.config(); 716         pernyataan = koneksi.createStatement(); 717         hasil = pernyataan.executeQuery(sql); 718     719         while (hasil.next()) { 720             721             tbl.addRow(new Object[]{ 722                 hasil.getString("transaksi_id"), 723                 hasil.getString("nama"), 724                 hasil.getString("harga"), 725                 hasil.getString("jumlah"), 726                 hasil.getString("subtotal")}); 727         } 728     729         tabelTransaksi.setModel(tbl); 730     } catch (Exception e) { 731     } 732 } 733 }
```

Gambar 2.88 : Fungsi searchTransaksi() pada halaman form transaksi (3)

24. Buat fungsi functionEdit(), isi fungsi tersebut.

```
Start Page | 734 public void functionEdit() { 735     736         if (txtKodeTransaksi.getText().equals("") || txtNamaBarang.getText().equals("") || txtHarga.getText().equals("") || 737             JOptionPane.showMessageDialog(null, "Tidak ada data yang dipilih! Silahkan memilih data lagi"); 738     } else { 739         740             txtJumlah.setEnabled(true); 741             tombol_mati(); 742             txtJumlah.requestFocus(); 743             btnTambah.setEnabled(true); 744             btnTambah.setText("Update"); 745             btnEdit.setEnabled(false); 746             txtKodeTransaksi.setEnabled(false); 747             txtKodeBarang.setEnabled(false); 748     } 749 } 750 }
```

Gambar 2.89 : Fungsi functionEdit() pada halaman form transaksi

25. Buat fungsi `functionTotalTagihan()`, dengan isi fungsi tersebut.

Gambar 2.90 : Fungsi functionTotalTagihan() pada halaman form transaksi

26. Buat fungsi bayar(), dengan isi fungsi tersebut.

Gambar 2.91 : Fungsi bayar() pada halaman form transaksi (1)

The screenshot shows a Java code editor with several tabs at the top: Start Page, Menu_Login.java, Menu_UserAdmin.java, Login_User.java, and Form_Transaksi.java. The current view is the Form_Transaksi.java file. The code is part of a method, likely a button's action listener, which handles payment confirmation. It includes logic to check if the amount entered matches the total bill and to set the payment status to 'Selesai'.

```
779
780
781
782
783
784
785
786
787
788
789     "Konfirmasi Pembayaran Transaksi", dialogButton);
790
791
792
793
794
795
796
797     sama dengan total tagihan!");
798
799
800     t().equals(" , ")
801
802
803
804     t!=");
```

Gambar 2.92 : Fungsi bayar() pada halaman form transaksi (2)

This screenshot continues the Java code from the previous one. It shows the completion of the payment confirmation logic, including the update of the transaction status to 'Selesai' and the reduction of item stock in the database. The code uses JDBC to execute SQL UPDATE and SELECT statements.

```
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830 } else {
    hasil = ConvUang - ConvTotalTagihan;
    ConvHasil = Integer.toString(hasil);

    txtKembalian.setText(ConvHasil);

    ambilKodeTransaksiTerakhir();
    // Menganti status pembayaran pada transaksi yang terkait
    try {
        //Mengatur agar tidak langsung di commit ketika proses transaksi terjadi
        koneksi = jaringan.Jaringan.config();
        koneksi.setAutoCommit(false);

        sql = "UPDATE tb_transaksi"
            + " SET status_pembayaran = 'Selesai', total_harga =" + txtTotalTagihan.getText() + ", nomor_kode_transaksi = '" + txtKodeTransaksi.getText() + "'"
            + " WHERE id_transaksi LIKE '%" + txtKodeTransaksi.getText() + "%'";

        pernyataan = koneksi.createStatement();
        pernyataan.execute(sql);

        // Mengurangi stok yang sudah ada dengan jumlah barang yang dibeli
        sql = "SELECT *"
            + " FROM tb_detail_transaksi"
            + " WHERE transaksi_id LIKE '%" + txtKodeTransaksi.getText() + "%'";
```

Gambar 2.93 : Fungsi bayar() pada halaman form transaksi (3)

A screenshot of a Java IDE interface, likely Eclipse, showing the source code for a Java file named Form_Transaksi.java. The code is part of a payment function, specifically handling the calculation of change. The code includes line numbers from 805 to 830. The IDE has tabs for Start Page, Menu_Login.java, Menu_UserAdmin.java, Login_User.java, and Form_Transaksi.java. The Properties and Output panes are visible on the right side of the interface.

```
805  
806  
807  
808  
809  
810  
811  
812  
813  
814  
815  
816  
817  
818  
819  
820  
821     t() + ", nominal_pembayaran =" + txtCash.getText() + ", kembalian = " + txtKembalian.getText()  
822  
823  
824  
825  
826  
827  
828  
829  
830
```

Gambar 2.94 : Fungsi bayar() pada halaman form transaksi (4)

A screenshot of a Java IDE interface, likely Eclipse, showing the source code for a Java file named Form_Transaksi.java. This part of the code handles the update of database records after a payment is made. It uses Statement objects to execute SQL queries for updating stock levels and committing transactions. The code includes line numbers from 831 to 856. The IDE has tabs for Start Page, Menu_Login.java, Menu_UserAdmin.java, Login_User.java, and Form_Transaksi.java. The Properties and Output panes are visible on the right side of the interface.

```
831  
832     pernyataan = koneksi.createStatement();  
833     this.hasil = pernyataan.executeQuery(sql);  
834  
835     while (this.hasil.next()) {  
836  
837         namaBarang = this.hasil.getString("nama");  
838         transaksi_id = this.hasil.getString("transaksi_id");  
839  
840         sql2 = "UPDATE tb_barang"  
841             + " SET stok = (SELECT (tb_barang.stok - tb_detail_transaksi.jumlah) AS stok_update"  
842             + " FROM tb_barang"  
843             + " INNER JOIN tb_detail_transaksi ON (tb_detail_transaksi.nama = tb_barang.nama)"  
844             + " WHERE tb_detail_transaksi.nama LIKE '%" + namaBarang + "%' AND tb_detail_transaksi"  
845             + " WHERE nama LIKE '%" + namaBarang + "%'";  
846  
847         pernyataan = koneksi.createStatement();  
848         pernyataan.executeUpdate(sql2);  
849     }  
850  
851     // commit jika sistem bayar berhasil dijalankan  
852     koneksi.commit();  
853  
854     JOptionPane.showMessageDialog(null, "Pembayaran Berhasil");  
855 } catch (Exception e) {  
856 }
```

Gambar 2.95 : Fungsi bayar() pada halaman form transaksi (5)

A screenshot of a Java IDE interface, likely Eclipse, showing the source code for a Java file named `Form_Transaksi.java`. The code is part of a function definition, specifically the `bayar()` method. The code includes database queries using JDBC, such as `SELECT * FROM detail_transaksi WHERE transaksi_id LIKE '%' + transaksi_id + '%'`. The code is annotated with line numbers from 831 to 856.

```
831  
832  
833  
834  
835  
836  
837  
838  
839  
840  
841     stok_update"  
842  
843 rang.nama"  
844     detail_transaksi.transaksi_id LIKE '%' + transaksi_id + "%'"  
845  
846  
847  
848  
849  
850  
851  
852  
853  
854  
855  
856
```

Gambar 2.96 : Fungsi bayar() pada halaman form transaksi (6)

A screenshot of a Java IDE interface, likely Eclipse, showing the continuation of the `bayar()` function from the previous screenshot. The code handles the execution of SQL statements, committing transactions, and displaying success or error messages using `JOptionPane.showMessageDialog`. The code is annotated with line numbers from 846 to 870.

```
846  
847  
848         pernyataan = koneksi.createStatement();  
849         pernyataan.execute(sql2);  
850  
851         // commit jika sistem bayar berhasil dijalankan  
852         koneksi.commit();  
853  
854         JOptionPane.showMessageDialog(null, "Pembayaran Berhasil");  
855     } catch (Exception e) {  
856  
857         JOptionPane.showMessageDialog(null, "Keterangan Error :" + e);  
858  
859         // commit jika sistem bayar berhasil dijalankan  
860         koneksi.rollback();  
861     }  
862 }  
863 } else {  
864     JOptionPane.showMessageDialog(null, "Pembayaran Pada Transaksi Ini Dibatalkan");  
865 }  
866 }  
867 }  
868 }  
869 }  
870 }
```

Gambar 2.97 : Fungsi bayar() pada halaman form transaksi (7)

27. Buat fungsi subtotalHargaBarang(), isi fungsi tersebut.

The screenshot shows a Java code editor with the following code:

```
Start Page | Menu_Login.java | Menu_UserAdmin.java | Login_User.java | Form_Transaksi.java | Output | Properties
```

```
Source Design History | Back Forward Stop Refresh Home | Help | Options | Exit | Help | Options | Exit |
```

```
871 private void subtotalHargaBarang() {  
872  
873     String jumlah = txtJumlah.getText();  
874     String harga = txtHarga.getText();  
875  
876     if (!jumlah.isEmpty() && !harga.isEmpty()) {  
877  
878         int ConvJumlah = Integer.parseInt(jumlah);  
879         int ConvHarga = Integer.parseInt(harga);  
880         int hasil;  
881  
882         if (jumlah.equals("-") || jumlah.equals(".")) || jumlah.equals(",")  
883             || jumlah.equals("_") || jumlah.equals("k") || jumlah.equals("K")) {  
884             JOptionPane.showMessageDialog(null, "Format Angka yang Anda Masukkan Kurang Tepat!");  
885         } else {  
886             hasil = ConvJumlah * ConvHarga;  
887  
888             String ConvHasil = Integer.toString(hasil);  
889  
890             txtSubtotal.setText(ConvHasil);  
891         }  
892     }  
893 }  
894  
895 }
```

Gambar 2.98 : Fungsi subtotalHargaBarang() pada halaman form transaksi

28. Buat fungsi TransaksiBaru(), isi fungsi tersebut.

The screenshot shows a Java code editor with the following code:

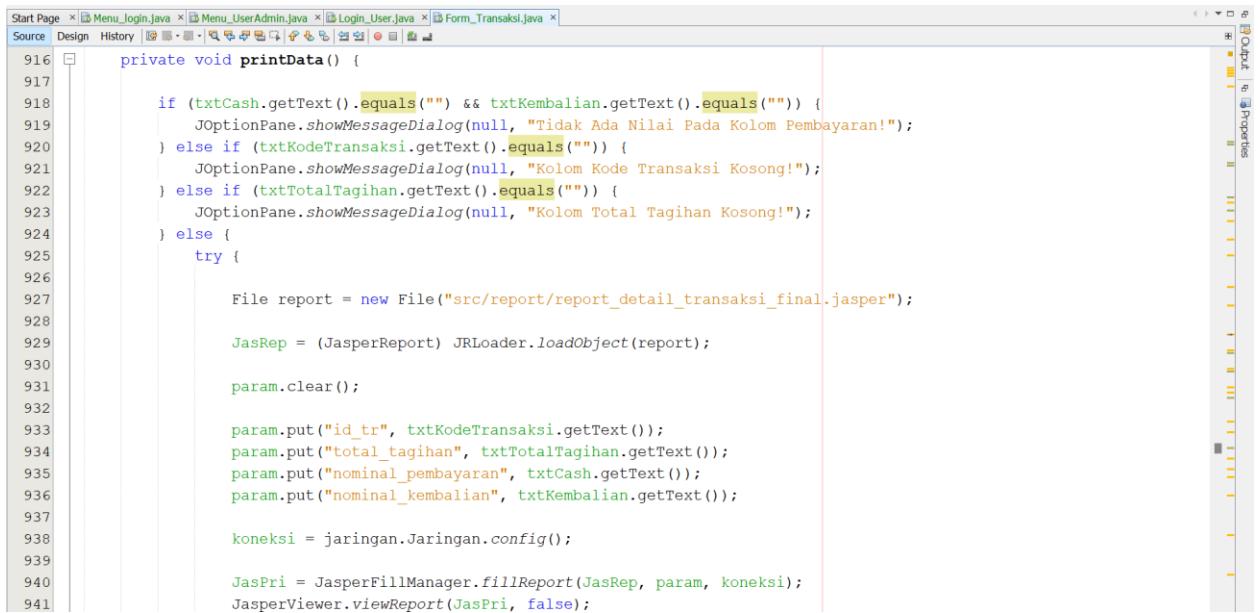
```
Start Page | Menu_Login.java | Menu_UserAdmin.java | Login_User.java | Form_Transaksi.java | Output | Properties
```

```
Source Design History | Back Forward Stop Refresh Home | Help | Options | Exit | Help | Options | Exit |
```

```
896 private void TransaksiBaru() throws SQLException {  
897     int dialogButton = JOptionPane.YES_NO_OPTION;  
898     int dialogResult = JOptionPane.showConfirmDialog(this, "Anda Ingin Membuat Transaksi Baru?", "Konfirmasi Pembuatan  
Transaksi", dialogButton);  
899  
900     if (dialogResult == 0) {  
901  
902         AutoIncrement();  
903  
904         tidak_bisa_isi();  
905  
906         isi_tableTransaksi();  
907         isi_tableBarang();  
908     } else {  
909         JOptionPane.showMessageDialog(null, "Pembuatan Transaksi Gagal");  
910     }  
911 }  
912  
913  
914 }  
915 }
```

Gambar 2.99 : Fungsi transaksiBaru() pada halaman form transaksi

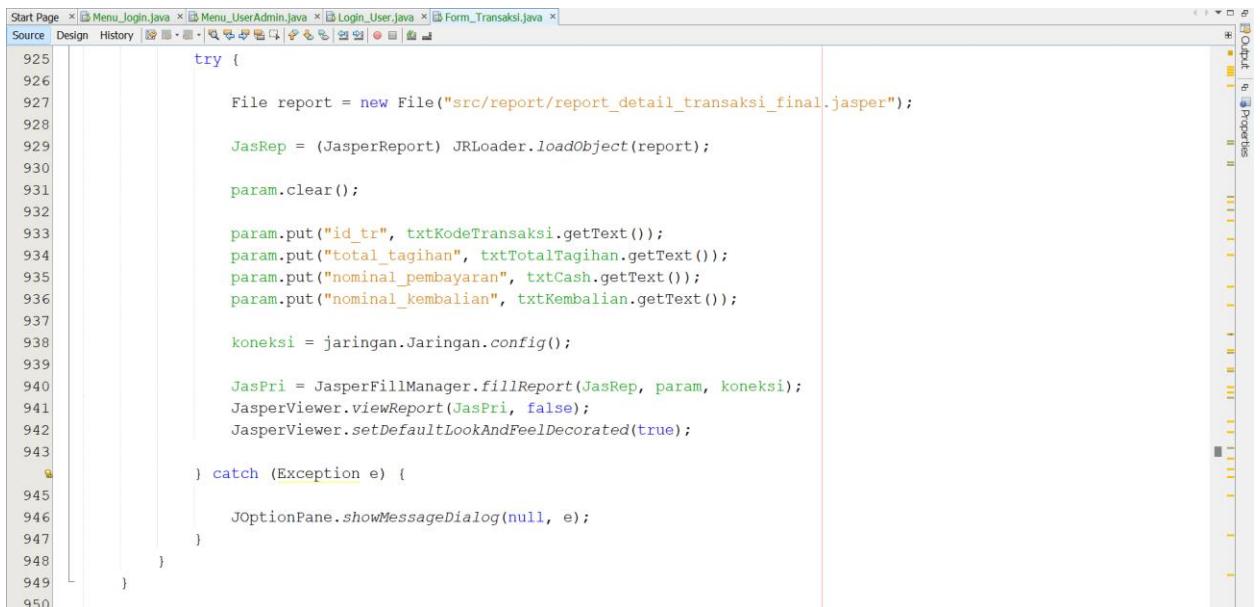
29. Buat fungsi printData(), isi fungsi tersebut.



The screenshot shows a Java code editor with the file `Form_Transaksi.java` open. The code defines a `private void printData()` method. It first checks if three text fields (`txtCash`, `txtKembalian`, and `txtKodeTransaksi`) are empty. If any are empty, it shows a message dialog. Otherwise, it creates a `JasperReport` object from a Jasper file, initializes parameters with values from the text fields, and connects to a database using a `Jaringan` configuration. Finally, it fills the report with the parameters and database connection, and displays it using a `JasperViewer`.

```
916     private void printData() {  
917  
918         if (txtCash.getText().equals("") && txtKembalian.getText().equals("")) {  
919             JOptionPane.showMessageDialog(null, "Tidak Ada Nilai Pada Kolom Pembayaran!");  
920         } else if (txtKodeTransaksi.getText().equals("")) {  
921             JOptionPane.showMessageDialog(null, "Kolom Kode Transaksi Kosong!");  
922         } else if (txtTotalTagihan.getText().equals("")) {  
923             JOptionPane.showMessageDialog(null, "Kolom Total Tagihan Kosong!");  
924         } else {  
925             try {  
926  
927                 File report = new File("src/report/report_detail_transaksi_final.jasper");  
928  
929                 JasRep = (JasperReport) JRLoader.loadObject(report);  
930  
931                 param.clear();  
932  
933                 param.put("id_tr", txtKodeTransaksi.getText());  
934                 param.put("total_tagihan", txtTotalTagihan.getText());  
935                 param.put("nominal_pembayaran", txtCash.getText());  
936                 param.put("nominal_kembalian", txtKembalian.getText());  
937  
938                 koneksi = jaringan.Jaringan.config();  
939  
940                 JasPri = JasperFillManager.fillReport(JasRep, param, koneksi);  
941                 JasperViewer.viewReport(JasPri, false);  
942                 JasperViewer.setDefaultLookAndFeelDecorated(true);  
943             } catch (Exception e) {  
944                 JOptionPane.showMessageDialog(null, e);  
945             }  
946         }  
947     }  
948  
949 }  
950
```

Gambar 2.100 : Fungsi printData() pada halaman form transaksi (1)



This screenshot shows the same `Form_Transaksi.java` file, but the `printData()` method now includes a `try-catch` block. The `catch` block handles any exceptions that might occur during the report filling process by displaying an error message dialog.

```
925     try {  
926  
927         File report = new File("src/report/report_detail_transaksi_final.jasper");  
928  
929         JasRep = (JasperReport) JRLoader.loadObject(report);  
930  
931         param.clear();  
932  
933         param.put("id_tr", txtKodeTransaksi.getText());  
934         param.put("total_tagihan", txtTotalTagihan.getText());  
935         param.put("nominal_pembayaran", txtCash.getText());  
936         param.put("nominal_kembalian", txtKembalian.getText());  
937  
938         koneksi = jaringan.Jaringan.config();  
939  
940         JasPri = JasperFillManager.fillReport(JasRep, param, koneksi);  
941         JasperViewer.viewReport(JasPri, false);  
942         JasperViewer.setDefaultLookAndFeelDecorated(true);  
943     } catch (Exception e) {  
944         JOptionPane.showMessageDialog(null, e);  
945     }  
946 }  
947  
948 }  
949  
950 }
```

Gambar 2.101 : Fungsi printData() pada halaman form transaksi (2)

30. Buat fungsi ambilKodeTransaksiTerakhir(), isi fungsi tersebut.

Gambar 2. 102 : Fungsi ambilKodeTransaksiTerakhir() pada halaman form transaksi

31. Buat fungsi ambilKodeBarangTerkait(), isi fungsi tersebut.

Gambar 2.103 : Fungsi ambilKodeBarangTerkait() pada halaman form transaksi

32. Buat fungsi mergePembelianBarang(), isi fungsi tersebut.

```
Start Page | Menu_Login.java | Menu_UserAdmin.java | Login_User.java | Form_Transaksi.java | 
Source Design History | 
995     private void mergePembelianBarang() throws SQLException {
996 
997         String nama_barang = txtNamaBarang.getText();
998         String transaksi_id = txtKodeTransaksi.getText();
999         int jumlah = Integer.parseInt(txtJumlah.getText());
1000        int subtotal = Integer.parseInt(txtSubtotal.getText());
1001 
1002        // Pengecekan kesamaan barang dengan data di database
1003        try {
1004 
1005            //Mengatur agar tidak langsung di commit ketika proses transaksi terjadi
1006            koneksi = jaringan.Jaringan.config();
1007            koneksi.setAutoCommit(false);
1008 
1009            sql = "SELECT *"
1010                + " FROM tb_detail_transaksi"
1011                + " WHERE nama LIKE '%" + nama_barang + "%' AND transaksi_id LIKE '%" + transaksi_id + "%'";
1012 
1013            pernyataan = koneksi.createStatement();
1014            hasil = pernyataan.executeQuery(sql);
1015 
1016            while (hasil.next()) {
1017 
1018                int jumlahl = Integer.parseInt(hasil.getString("jumlah"));
1019                int subtotall = Integer.parseInt(hasil.getString("subtotal"));
1020            }
1021        }
```

Gambar 2.104 : Fungsi mergePembelianBarang() pada halaman form transaksi (1)

```
Start Page | Menu_Login.java | Menu_UserAdmin.java | Login_User.java | Form_Transaksi.java | 
Source Design History | 
1018        int jumlahl = Integer.parseInt(hasil.getString("jumlah"));
1019        int subtotall = Integer.parseInt(hasil.getString("subtotal"));
1020 
1021        int jumlah_hasil = jumlahl + jumlah;
1022        int subtotal_hasil = subtotall + subtotal;
1023 
1024        // Mengupdate data detail_transaksi
1025        sql2 = "UPDATE tb_detail_transaksi"
1026            + " SET subtotal = " + subtotal_hasil + ", jumlah = " + jumlah_hasil
1027            + " WHERE nama LIKE '%" + nama_barang + "%' AND transaksi_id LIKE '%" + transaksi_id + "%'";
1028 
1029        pernyataan = koneksi.createStatement();
1030        pernyataan.executeUpdate(sql2);
1031 
1032        // commit jika sistem bayar berhasil dijalankan
1033        koneksi.commit();
1034    }
1035 } catch (Exception e) {
1036 
1037     JOptionPane.showMessageDialog(null, "Keterangan Error :" + e);
1038 
1039     // commit jika sistem bayar berhasil dijalankan
1040     koneksi.rollback();
1041 }
1042 }
```

Gambar 2.105 : Fungsi mergePembelianBarang() pada halaman form transaksi (2)

33. Buat fungsi event. Fungsi event ini digunakan agar element pada halaman jframe dapat berjalan sesuai dengan event yang anda tentukan. Untuk membuat event, anda bisa ke tab design => tekan kanan mouse pada elemen yang ingin kasih event => pilih menu event =>

pilih event yang anda inginkan. Event yang tersedia untuk elemen itu bermacam – macam, sesuaikan dengan kebutuhan anda. Berikut beberapa contoh event yang digunakan pada halaman form transaksi:

```
Start Page | Menu_Login.java | Menu_UserAdmin.java | Login_User.java | Form_Transaksi.java | 
Source Design History | 
1527 | private void btnHapusActionPerformed(java.awt.event.ActionEvent evt) { 
1528 |     // TODO add your handling code here: 
1529 | 
1530 |     hapus(); 
1531 | 
1532 |     tombol_hidup(); 
1533 |     Kosong(); 
1534 |     functionTotalTagihan(); 
1535 | } 
1536 | 
1537 | private void btnBayarActionPerformed(java.awt.event.ActionEvent evt) { 
1538 | 
1539 |     try { 
1540 |         // TODO add your handling code here: 
1541 | 
1542 |         bayar(); 
1543 |         isi_tableBarang(); 
1544 |     } catch (SQLException ex) { 
1545 |         Logger.getLogger(Form_Transaksi.class.getName()).log(Level.SEVERE, null, ex); 
1546 |     } 
1547 | 
1548 |     functionTotalTagihan(); 
1549 | } 
1550 | 
1551 | }
```

Gambar 2.106 : Fungsi Event dan isinya pada halaman Transaksi (1)

```
Start Page | Menu_Login.java | Menu_UserAdmin.java | Login_User.java | Form_Transaksi.java | 
Source Design History | 
1551 | private void btnCetakBuktiPembayaranActionPerformed(java.awt.event.ActionEvent evt) { 
1552 |     // TODO add your handling code here: 
1553 | 
1554 |     ambilKodeTransaksiTerakhir(); 
1555 |     printData(); 
1556 |     functionTotalTagihan(); 
1557 | } 
1558 | 
1559 | private void btnSearchInputBarangActionPerformed(java.awt.event.ActionEvent evt) { 
1560 |     // TODO add your handling code here: 
1561 | 
1562 |     SearchBarang(); 
1563 |     functionTotalTagihan(); 
1564 | } 
1565 | 
1566 | private void btnBackActionPerformed(java.awt.event.ActionEvent evt) { 
1567 |     // TODO add your handling code here: 
1568 | 
1569 |     jikaAktif(); 
1570 | 
1571 |     dispose(); 
1572 | } 
1573 | 
1574 | }
```

Gambar 2.107 : Fungsi Event dan isinya pada halaman Transaksi (2)

The screenshot shows a Java IDE interface with multiple tabs at the top: Start Page, Menu_Login.java, Menu_UserAdmin.java, Login_User.java, and Form_Transaksi.java. The Form_Transaksi.java tab is active. The code editor displays several event-handling methods:

```
1575 private void btnKeluarActionPerformed(java.awt.event.ActionEvent evt) {  
1576     // TODO add your handling code here:  
1577  
1578     dispose();  
1579 }  
1580  
1581 private void tabelBarangKeyPressed(java.awt.event.KeyEvent evt) {  
1582     // TODO add your handling code here:  
1583  
1584 }  
1585  
1586 private void tabelBarangMouseClicked(java.awt.event.MouseEvent evt) {  
1587     // TODO add your handling code here:  
1588  
1589     ambilDataTabelBarang();  
1590     functionTotalTagihan();  
1591 }  
1592  
1593  
1594 private void tabelTransaksiMouseClicked(java.awt.event.MouseEvent evt) {  
1595     // TODO add your handling code here:  
1596  
1597     sistemTabelTransaksi();  
1598     functionTotalTagihan();  
1599 }  
1600
```

Gambar 2.108 : Fungsi Event dan isinya pada halaman Transaksi (3)

The screenshot shows a Java IDE interface with multiple tabs at the top: Start Page, Menu_Login.java, Menu_UserAdmin.java, Login_User.java, and Form_Transaksi.java. The Form_Transaksi.java tab is active. The code editor displays event-handling code for the btnTambah button:

```
1600  
1601 private void btnTambahActionPerformed(java.awt.event.ActionEvent evt) {  
1602     // TODO add your handling code here:  
1603  
1604     if (btnTambah.getText() == "Tambah") {  
1605  
1606         try {  
1607             Simpan();  
1608         } catch (SQLException ex) {  
1609             Logger.getLogger(Form_Transaksi.class.getName()).log(Level.SEVERE, null, ex);  
1610         }  
1611  
1612         btnTambah.setEnabled(false);  
1613     } else {  
1614  
1615         updateJumlah();  
1616  
1617         tombol_hidup();  
1618         Kosong();  
1619     }  
1620  
1621     functionTotalTagihan();  
1622 }  
1623
```

Gambar 2.109 : Fungsi Event dan isinya pada halaman Transaksi (4)

The screenshot shows a Java IDE interface with multiple tabs at the top: Start Page, Menu_Login.java, Menu_UserAdmin.java, Login_User.java, and Form_Transaksi.java. The Form_Transaksi.java tab is active. The code editor displays several event-handling methods:

```
1623 private void btnResetActionPerformed(java.awt.event.ActionEvent evt) {  
1624     // TODO add your handling code here:  
1625  
1626     Kosong();  
1627  
1628     tombol_hidup();  
1629     functionTotalTagihan();  
1630 }  
1631  
1632 private void btnEditActionPerfomed(java.awt.event.ActionEvent evt) {  
1633     // TODO add your handling code here  
1634  
1635     functionEdit();  
1636     functionTotalTagihan();  
1637 }  
1638  
1639 private void btnSearchTransaksiActionPerfomed(java.awt.event.ActionEvent evt) {  
1640     // TODO add your handling code here:  
1641  
1642     SearchTransaksi();  
1643     functionTotalTagihan();  
1644 }  
1645
```

Gambar 2.110 : Fungsi Event dan isinya pada halaman Transaksi (5)

The screenshot shows a Java IDE interface with multiple tabs at the top: Start Page, Menu_login.java, Menu_UserAdmin.java, Login_User.java, and Form_Transaksi.java. The Form_Transaksi.java tab is active. The code editor displays two event-handling methods:

```
1646 private void btnNewTransaksiActionPerfomed(java.awt.event.ActionEvent evt) {  
1647     // TODO add your handling code here:  
1648  
1649     Kosong();  
1650  
1651     try {  
1652         TransaksiBaru();  
1653     } catch (SQLException ex) {  
1654         Logger.getLogger(Form_Transaksi.class.getName()).log(Level.SEVERE, null, ex);  
1655     }  
1656  
1657     functionTotalTagihan();  
1658  
1659     tombol_hidup();  
1660  
1661 }  
1662  
1663 private void txtJumlahActionPerfomed(java.awt.event.ActionEvent evt) {  
1664     // TODO add your handling code here:  
1665  
1666 }  
1667  
1668 }  
1669
```

Gambar 2.111 : Fungsi Event dan isinya pada halaman Transaksi (6)

Gambar 2.112 : Fungsi Event dan isinya pada halaman Transaksi (7)

K. Pembuatan report

Software Pihak Ketiga yang digunakan untuk membuat report pada project halaman login ini adalah ireport dari jasper report. Ireport ini bisa digunakan untuk segala macam, seperti membuat invoice tentang pembelian, membuat laporan, dan lain – lain. Sebelumnya anda kan sudah menginstal plugin ireport pada netbeans, sekarang anda harus mengisntal software TIBCO Jaspersoft Studio-6.20.0. Software Jaspersoft studio ini nantinya digunakan untuk mengcompile file jrxml yang sudah anda desain. Berikut cara membuat file jrxml ireport:

1. Tekan kanan mouse pada folder yang anda inginkan sebagai tempat untuk menyimpan file ireport => pilih New => pilih Report Wizard. Atau anda juga bisa Tekan kanan mouse pada folder yang anda inginkan sebagai tempat untuk menyimpan file ireport => pilih New => pilih Other => pilih categories-nya report => Report Wizard.
 2. Isi file name sesuai dengan keinginan anda, asalkan tidak ada spasi atau ‘-‘. Folder disarankan mengikuti default saja.
 3. Memilih ‘Connections’ atau ‘Data Source’. Kalau anda sudah pernah membuat connections, anda tinggal memilih nama ‘connections’ yang sesuai dengan keinginan anda. Kalau anda belum pernah membuat report, anda bisa koneksi terlebih dahulu. Untuk membuat koneksi

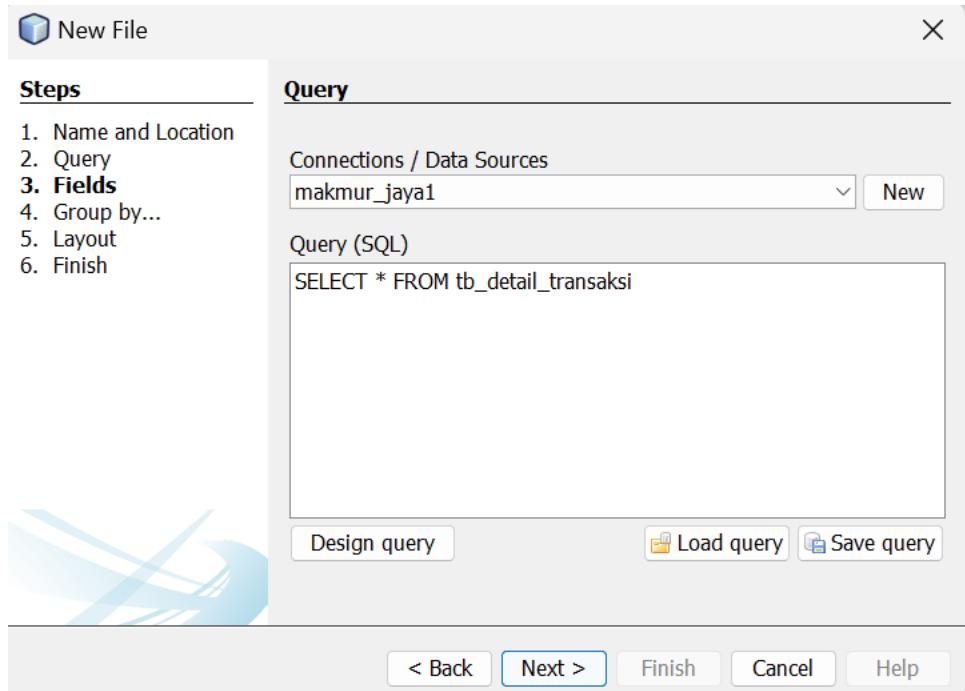
baru, anda bisa menekan tombol ‘New’ => Pilih jenis ‘datasource’ yang sesuai dengan kebutuhan anda (penulis memilih ‘Database JDBC Connections’), lalu tekan next => Isi nama koneksi, pilih jenis driver untuk koneksi (karena anda menggunakan mariaDB, maka menggunakan driver MySQL com.mysql.jdbc.Driver), isi JDBC URL, username dan password untuk mengakses database anda. Kalau anda bingung, kurang lebih setingannya seperti gambar 2.113.



Gambar 2113 : Settingan koneksi JDBC Ireport

4. Jika sudah menyeting koneksi JDBC, anda bisa mengisikan query pada kotak Query.

Query ini nantinya digunakan untuk mengambil data yang ada di database. Penulis mencotohkan jika ingin mengambil semua data pada tabel tb_detail_transaksi.



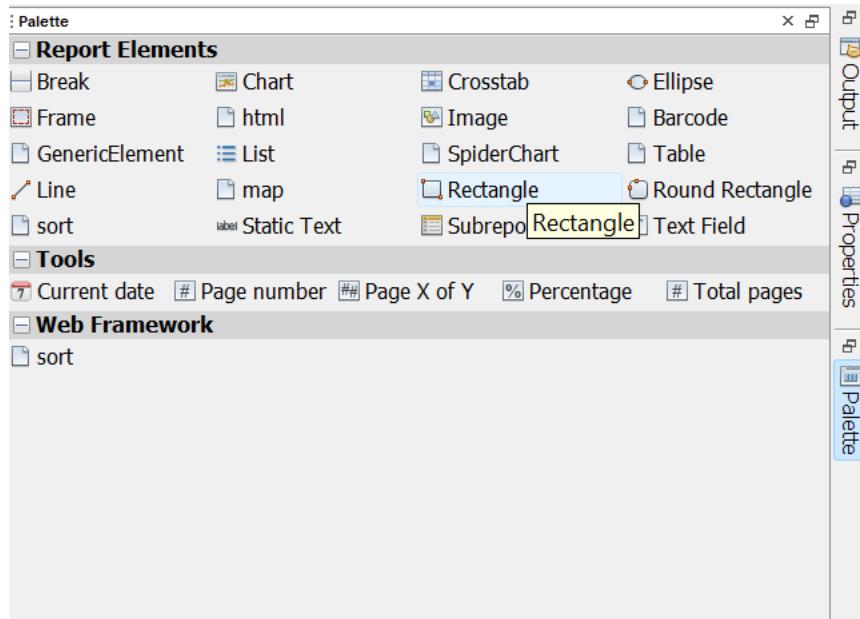
Gambar 2.114 : Query yang bisa diisikan pada kotak query Ireport

Jika query sudah diisikan, bisa menekan tombol next. Jika ada pop up konfirmasi password, isikan saja password dari database kita.

5. Setelah itu akan diarahkan ke group by, yang mana di halaman ini anda bisa memilih kolom mana yang akan ditampilkan di report nanti. Kalau anda bisa ingin memilih kolom yang akan ditampilkan, anda bisa menekan tombol lebih besar, untuk memindahkan kolom dari sisi kiri ke kanan.
6. Setelah itu ada pemilihan group by, di lompati saja tidak apa – apa.
7. Setelah itu diarahkan ke halaman terakhir setting, yang mana di halaman ini disuruh memilih tampilan ireport nanti, Columnar Layout atau Tabular layout.
8. Jika sudah klik finish, dan file jrxml ireport sudah berhasil dibuat.

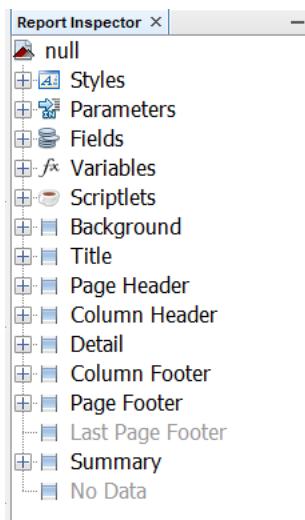
Jika file jrxml sudah dibuat, anda bisa mengedit file tersebut.

Kalau anda ingin menggunakan elemen yang disediakan oleh ireport, anda bisa menggunakan pallette.



Gambar 2.115 : Tab pallete pada ireport

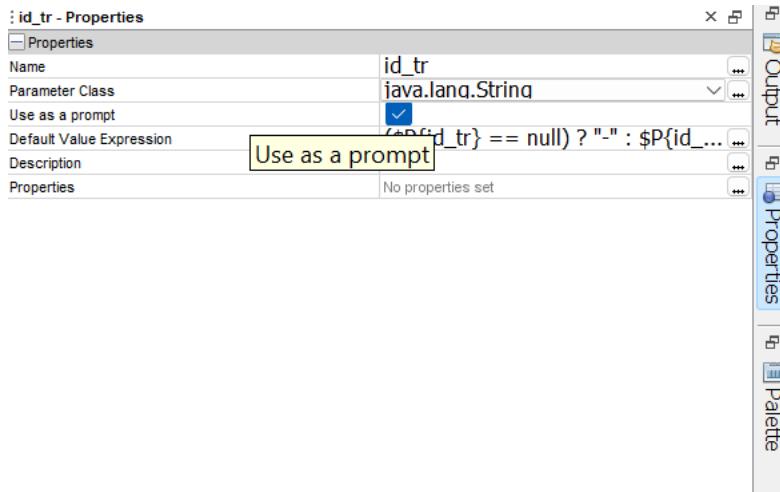
Anda juga bisa menambahkan beberapa komponen seperti variabel, parameter di tab report Inspector.



Gambar 2.116 : Tab report inspector

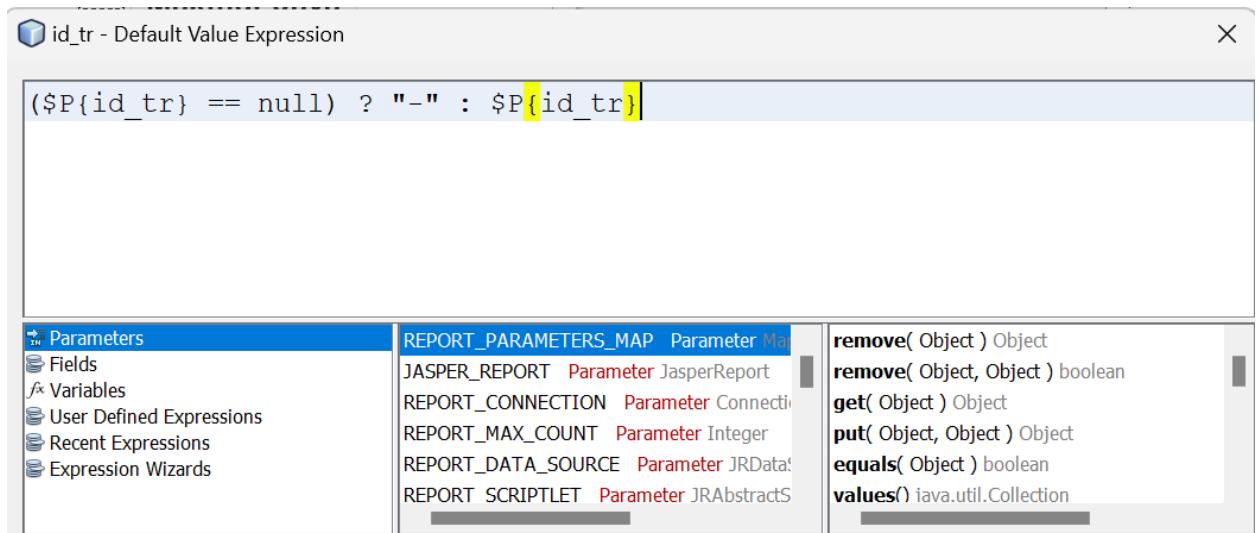
Anda harus menambahkan sebuah parameter yang nantinya bisa ditampilkan di report atau menjadi tumpuan saat mencari data pada database. Untuk menambahkan sebuah parameter, anda bisa menekan kanan mouse pada bagian ‘Parameters’ => lalu pilih add parameter => Jika parameter seudah dibuat, anda bisa merename parameter, dengan cara

klik kanan mouse dan pilih rename => Selanjutnya anda bisa merubah tipe data dari parameter tersebut dengan cara, tekan parameter yang dinginkan pilih properties => di bagian properties ini ada bisa mengganti tipe data, default value, dan sebagainya.



Gambar 2. 117 : properties parameter id_tr

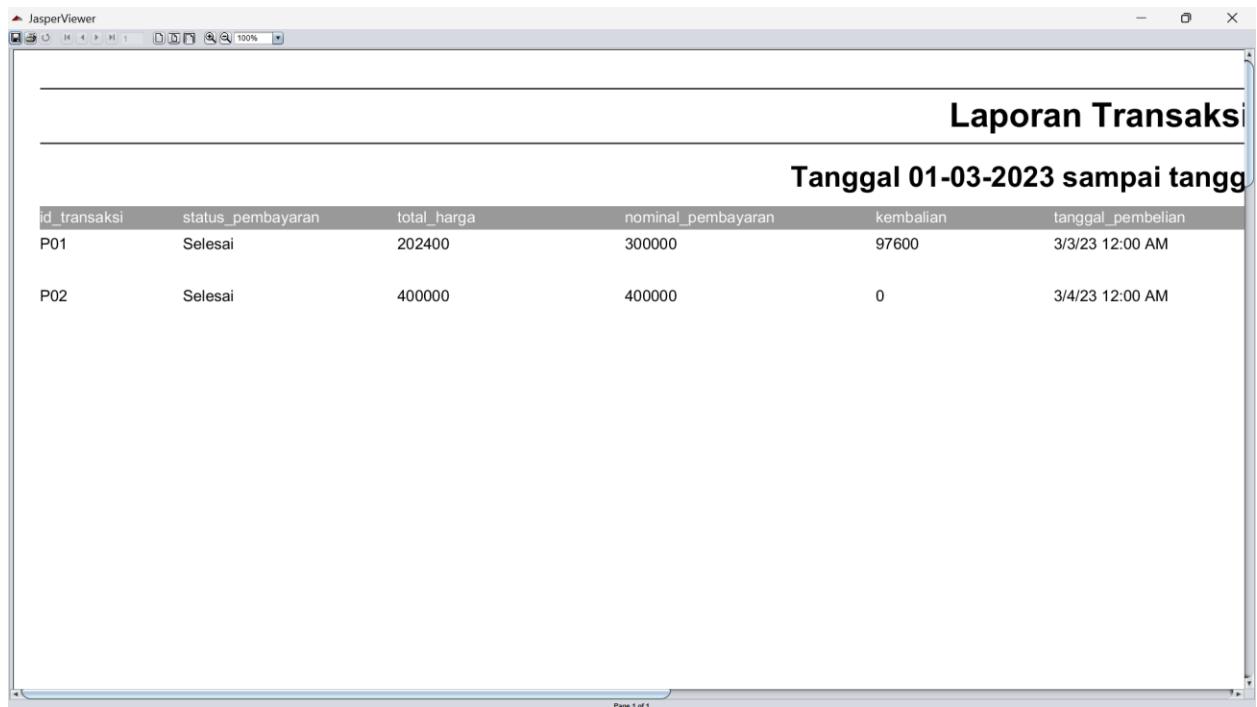
Anda juga harus menentukan default value dari setiap parameter, misalnya untuk parameter id_tr default valuenya seperti ini.



Gambar 2.118 : Default value parameter id_tr

Jika anda ingin memindahkan sebuah elemen ke bagian yang lain, misalkan dari page footer ke title. Anda harus memindahkan elemen dari bagian report inspector terlebih dahulu baru elemen dipindah dari layout ireport. Hal ini bertujuan agar tidak ada error elemen melebihi batas bagian.

Anda juga bisa membuat grouping untuk tiap datanya. Jadi nanti data yang tampil itu berdasarkan kolom tertentu. Misalnya saya ingin menggunakan 2 tabel pada ireport (tabel transaksi dan tabel detail transaksi), nah pada tabel detail transaksi kan berhubungan dengan tabel transaksi yang mana pada tabel detail transaksi terdapat data yang menggunakan 1 id transaksi yang sama. Nah dari data itu, penulis ingin membuat agar data yang ditampilkan di tabel transaksi cuma 1 saja, sedangkan untuk tabel detail bisa berkali kali. Seperti pada gambar 2.119 dan 2.120.



The screenshot shows a JasperViewer window displaying a report titled "Laporan Transaksi". The report includes a header section with the title and a date range "Tanggal 01-03-2023 sampai tanggal". Below this is a table with six columns: "id_transaksi", "status_pembayaran", "total_harga", "nominal_pembayaran", "kembalian", and "tanggal_pembelian". The table contains two rows of data:

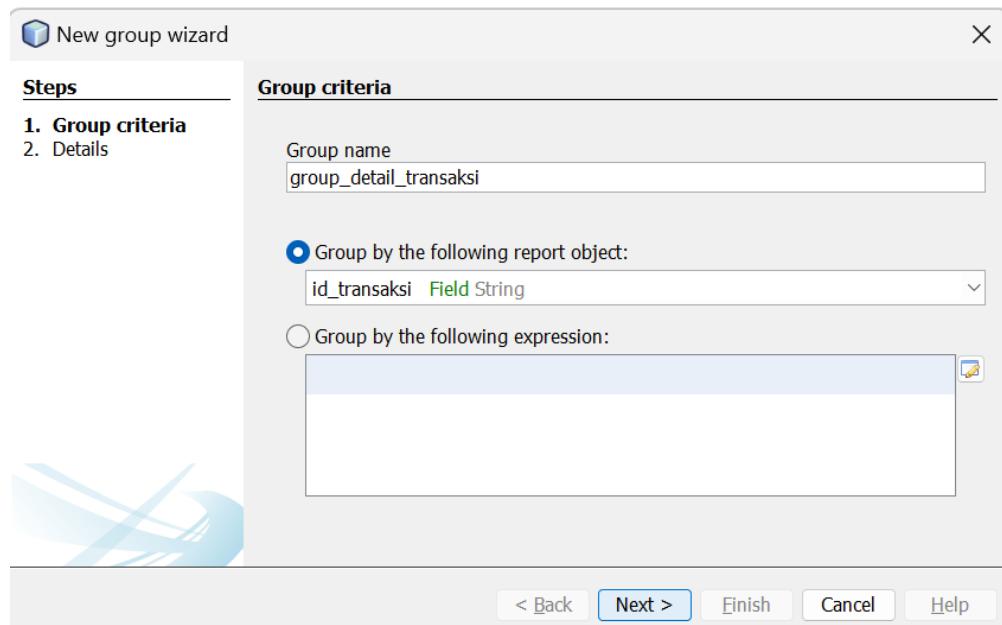
id_transaksi	status_pembayaran	total_harga	nominal_pembayaran	kembalian	tanggal_pembelian
P01	Selesai	202400	300000	97600	3/3/23 12:00 AM
P02	Selesai	400000	400000	0	3/4/23 12:00 AM

Gambar 2.119 : Contoh ireport yang menggunakan 2 tabel (1)

tanggal_pembelian	barang_kode	nama	jumlah	harga	subtotal
3/3/23 12:00 AM	4	Milo	2	100000	200000
	12	yakult	2	1200	2400
3/4/23 12:00 AM	4	Milo	2	100000	200000
	10	terang bulan	2	100000	200000

Gambar 2.120 : Contoh ireport yang menggunakan 2 tabel (2)

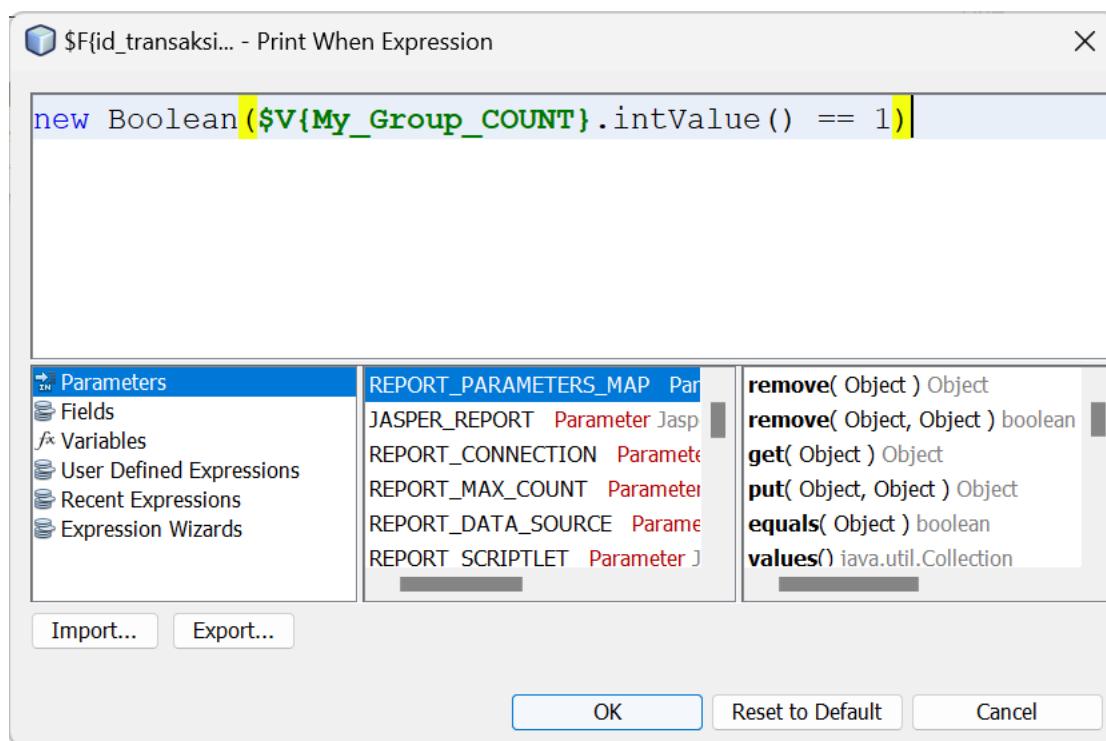
Untuk membuat grouping, anda bisa ke report inspector => tekan kanan mouse pada nama report => pilih add group => anda bisa mengisi nama group dan grouping berdasarkan apa (untuk kasus penulis menggunakan kolom id_transaksi).



Gambar 2.121 : Membuat group elemen ireport dengan group wizard

Jika sudah klik next => pada bagian details, anda bisa menambahkan header dan footer ireport (untuk kasus penulis header dan footer tidak dicentang) => jika sudah klik finish.

Jika sudah membuat group elemen, anda bisa menambahkan property printWhenExpression pada elemen yang ditampilkan dari tabel transaksi, agar elemen ini hanya tampil 1 kali per IDnya. Untuk menambahkan property, anda bisa ngeproperties elemen => pada bagian properties ada bagian printWhenExpression, isi sesuai gambar 2.122.

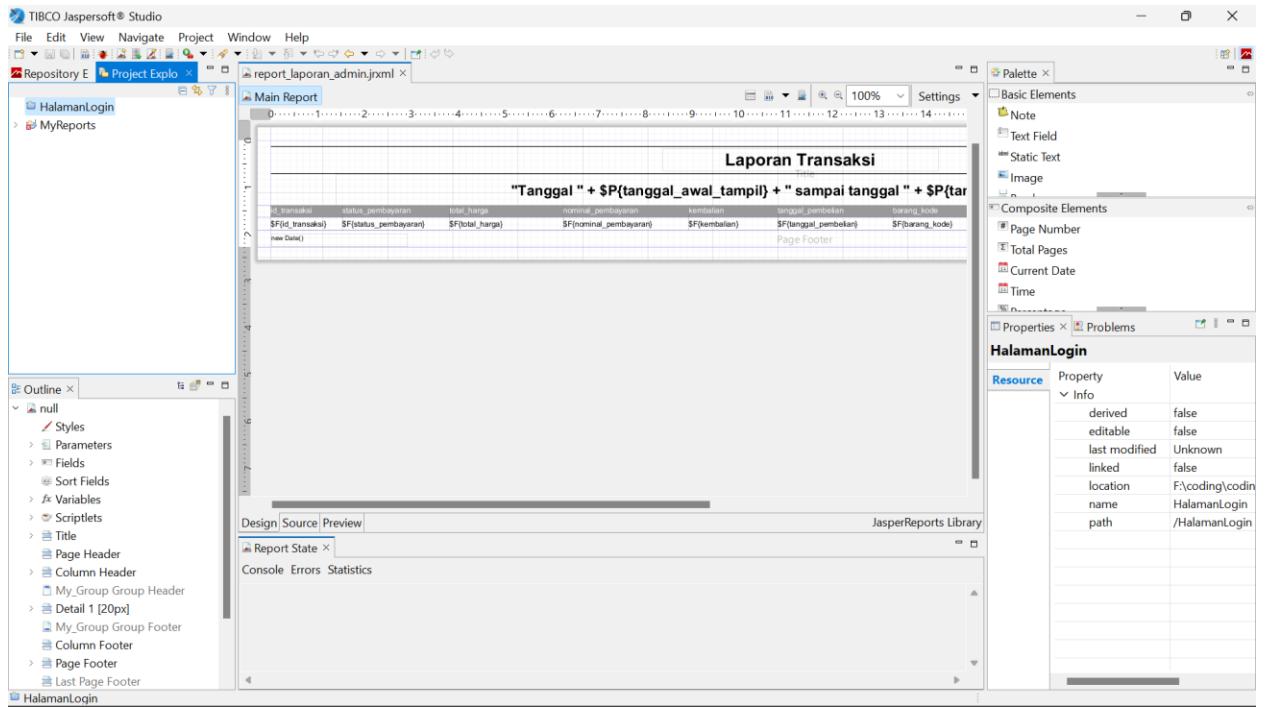


Gambar 2.122 : Contoh isi dari properti printWhenExpression

Pengkondisian itu akan membuat elemen hanya ditampilkan 1 kali saja, setiap id_transaksiya.

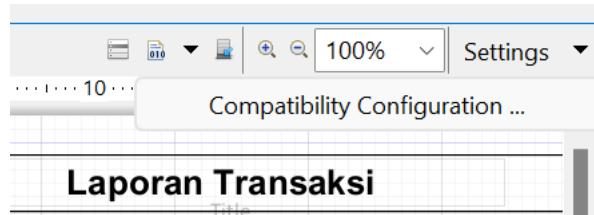
Jika anda sudah membuat mengedit ireport, anda bisa mengcompile file tersebut di software **TIBCO Jaspersoft Studio**. Berikut cara mengcompile file jrxml:

1. Buka software **TIBCO Jaspersoft Studio**.
2. Open file jrxml yang sudah diedit.



Gambar 2.123 : File ireport sudah terbuka

3. Pilih menu compatibility configuration pada bagian kanan atas.



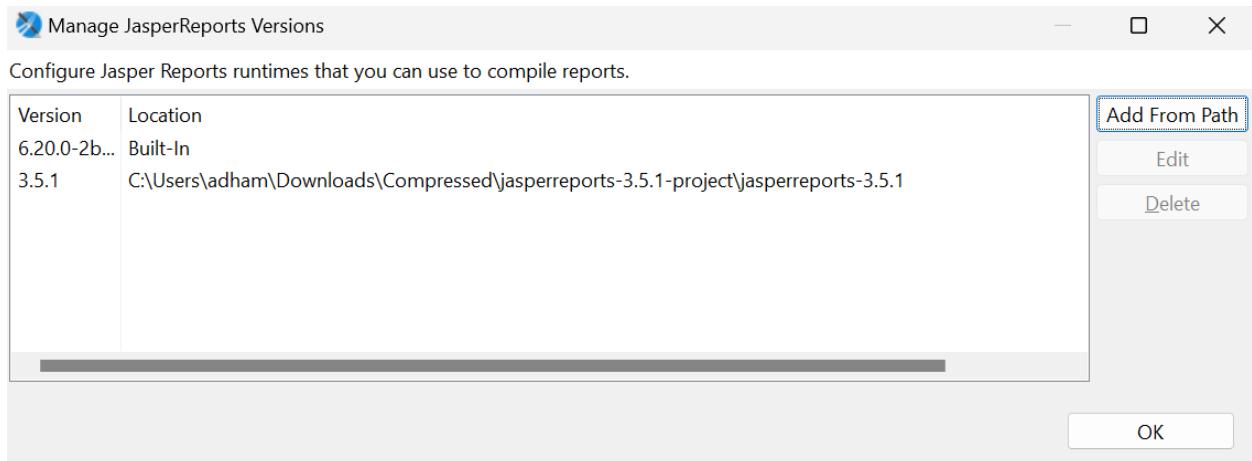
Gambar 2.124 : Compatibility configuration compiling

4. Dibagian setting compatibility compiler, anda bisa menggunakan project setting atau workspace setting. Sama saja keduanya. Penulis menggunakan workspace setting.

5. Jangan lupa untuk menentukan build compilernya. Untuk menentukan build version compiler, anda bisa menekan tulisan 'Manage Jasperreports Version'. Pilih version compiler 3.5.1. Jika belum ada anda harus menambahkan dari file manager terlebih dahulu. Tapi sebelum itu anda harus mempunyai folder compilernya, untuk mendownload file compiler yang dibutuhkan anda bisa mengunjungi link berikut:

https://drive.google.com/drive/folders/1jeDEzil3KcgIRHgoiI5tR1ajpRzEo1?usp=share_link

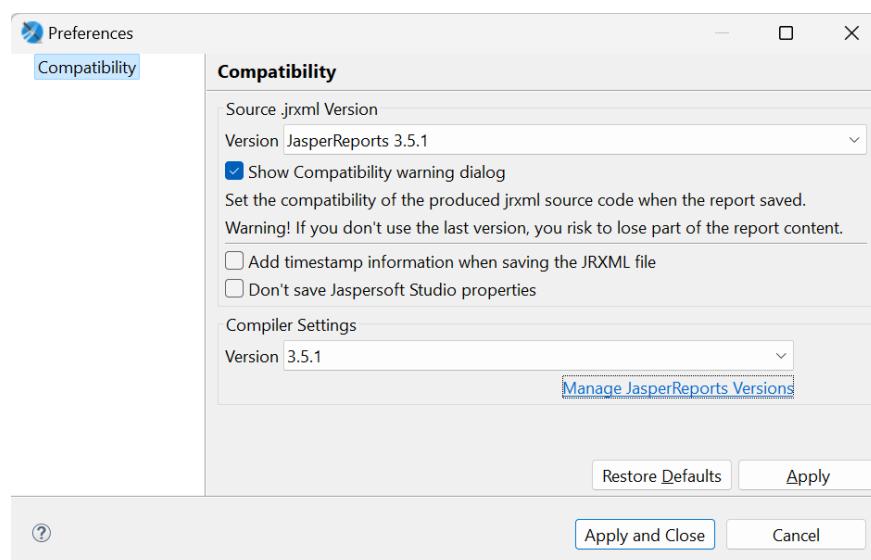
Download file zip jasper report versi 3.5.1 => ekstrak file zip tersebut => pada software TIBCO jasper report, tekan tombol add from path => tekan tombol browse untuk mencari folder jasper report yang sudah diekstrak => tekan ok. Maka compiler sudah terinstal.



Gambar 2.125 : compiler ireport 3.5.1 sudah terinstal

6. Jangan lupa untuk menentukan versi source jrxml dan compiler, pastikan sama versinya.

Jika sudah tekan apply.



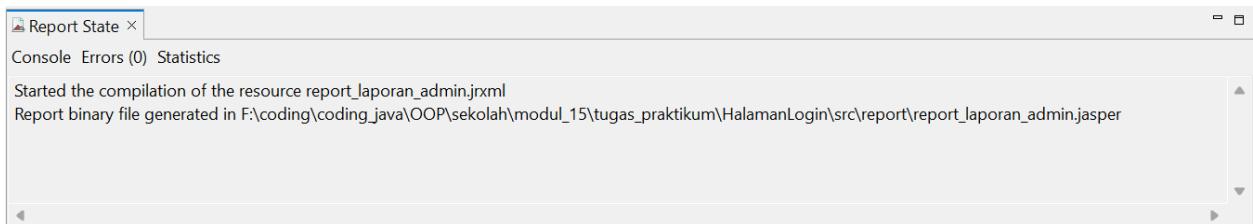
Gambar 2.126 : setting compatibility file dan compiler

7. Jika sudah menyeting compiler, anda sudah bisa mengcompiler file jrxmlnya dengan menekan tombol compile di bagian kanan atas.



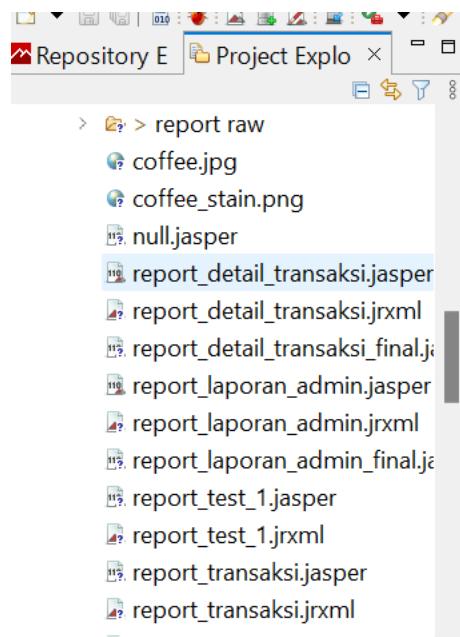
Gambar 2.127 : letak tombol compile file jrxml

8. Jika file sudah tercompile dengan benar, akan muncul tulisan seperti ini di bagian report state.



Gambar 2.128 : tulisan pada bagian report state bahwa file jrxml sudah berhasil di compile

Pada bagian project explorer akan muncul file baru juga yang bernama ‘null.jasper’. Nah file itu yang nantinya direname dan dipanggil di program java.

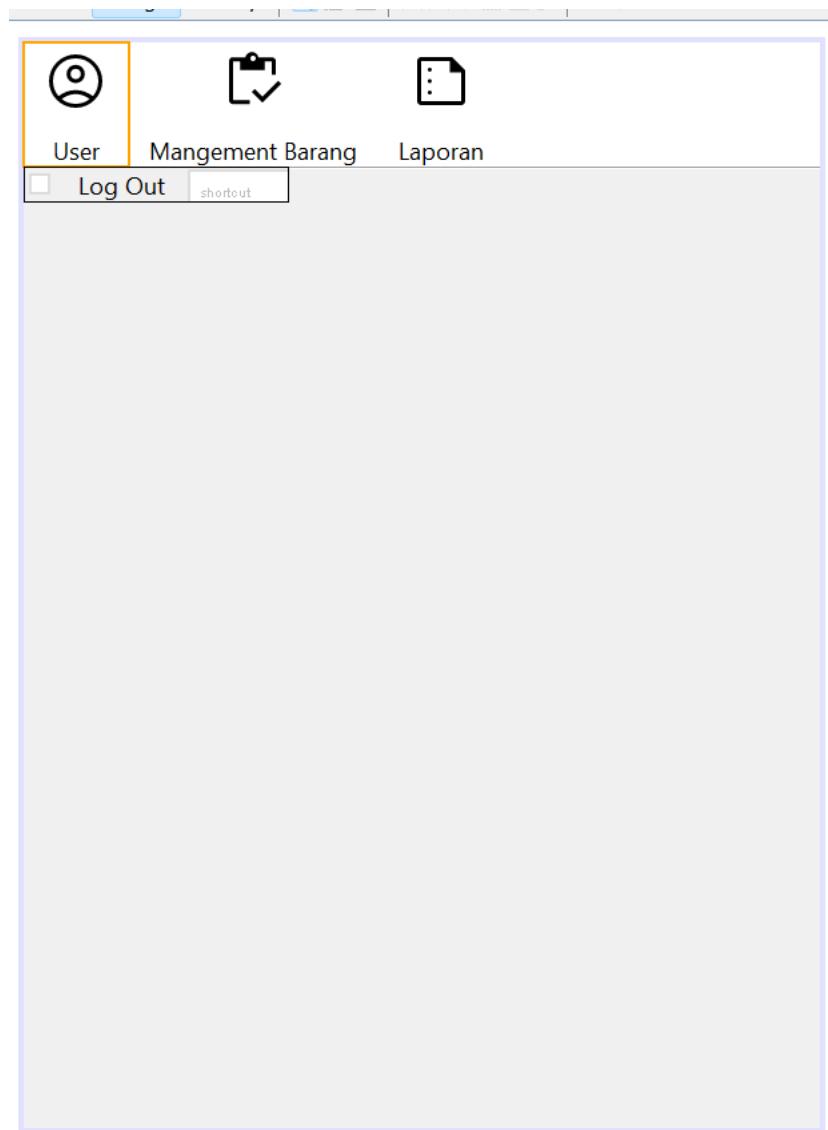


Gambar 2.129 : file null.jasper hasil dari compiling file jrxml

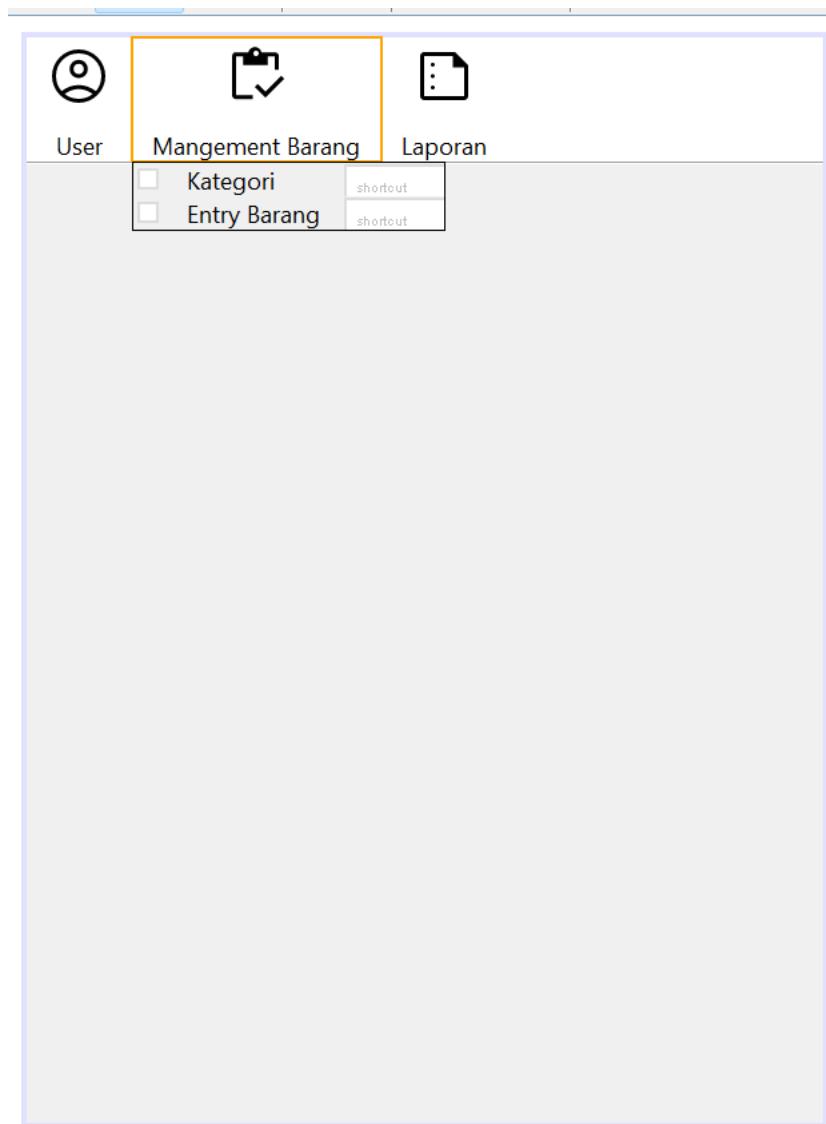
9. Ireport sudah bisa digunakan.

L. Pembuatan Halaman Utama Admin

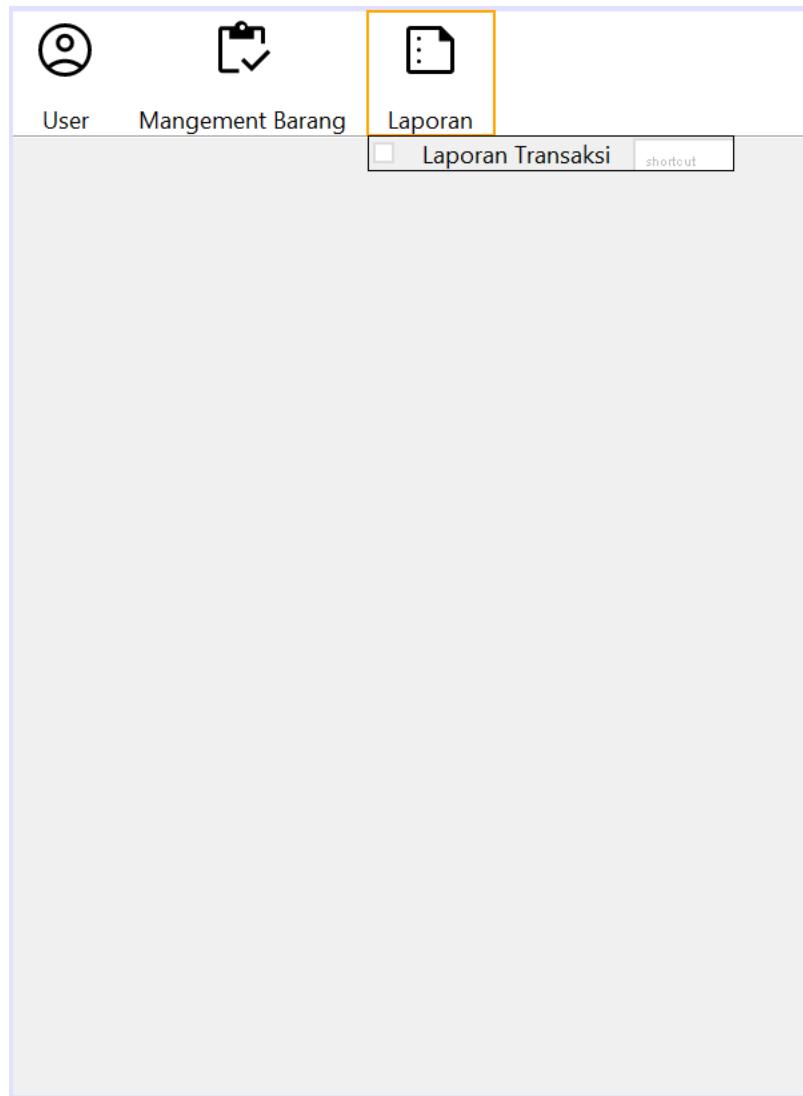
Halaman Utama ini adalah halaman perantara untuk menuju halaman kategori, entry barang, dan laporan transaksi. Halaman utama user kasir ini cuma memiliki 3 menu, menu User dengan submenu logout, menu management barang dengan submenu category dan entry barang, dan menu laporan dengan submenu laporan transaksi. Untuk membuat file jframe, anda bisa melihat pada ‘**Pembuatan Halaman Utama User Kasir**’. Untuk desain yang ada di halaman utama admin ini, anda harus membuat desain seperti gambar 2.127 sampai 2.129 ini.



Gambar 2.130 : Halaman Utama Admin (1)



Gambar 2.131 : Halaman Utama Admin (2)



Gambar 2.132 : Halaman Utama Admin (3)

Jika desain sudah dibuat, jangan lupa untuk mengganti nama variable tiap elemen, agar mempermudah untuk diingat. Setelah itu buka tab Soure pada halaman ini, lalu buat sistemnya agar halaman utama user ini dapat berjalan dengan baik. Berikut cara membuat sistem untuk halaman utama admin:

1. buat constructor sesuai dengan gambar 2.133.

```
    public AdminToko() {
        initComponents();
    }
```

Gambar 2.133 : Isi dari constructor pada halaman utama admin

2. Buat fungsi event. Fungsi event ini digunakan agar element pada halaman jframe dapat berjalan sesuai dengan event yang anda tentukan. Untuk membuat event, anda bisa ke tab design => tekan kanan mouse pada elemen yang ingin kasih event => pilih menu event => pilih event yang anda inginkan. Event yang tersedia untuk elemen itu bermacam – macam, sesuaikan dengan kebutuhan anda. Berikut beberapa contoh event yang digunakan pada halaman form utama admin:



The screenshot shows a Java code editor with several menu item event handlers defined. The code is as follows:

```
177
178     private void logout_menuItemActionPerformed(java.awt.event.ActionEvent evt) {
179         // TODO add your handling code here
180
181         dispose();
182         new Menu_login().setVisible(true);
183     }
184
185     private void EntryBarang_menuItemActionPerformed(java.awt.event.ActionEvent evt) {
186         // TODO add your handling code here:
187
188         dispose();
189         new Entry_Barang().setVisible(true);
190     }
191
192     private void Kategori_menuItemActionPerformed(java.awt.event.ActionEvent evt) {
193         // TODO add your handling code here:
194
195         dispose();
196         new Form_Kategori().setVisible(true);
197     }
198
199     private void Laporan_menuActionPerformed(java.awt.event.ActionEvent evt) {
200         // TODO add your handling code here:
201     }
202
```

Gambar 2.134 : Fungsi event dan isinya pada halaman utama admin (1)

```
202
203     private void LaporanTransaksi_menuItemActionPerformed(java.awt.event.ActionEvent evt) {
204         // TODO add your handling code here:
205
206         dispose();
207         new Laporan_Admin().setVisible(true);
208     }
209
```

Gambar 2.135 : Fungsi event dan isinya pada halaman utama admin (2)

M. Pembuatan Halaman Kategori

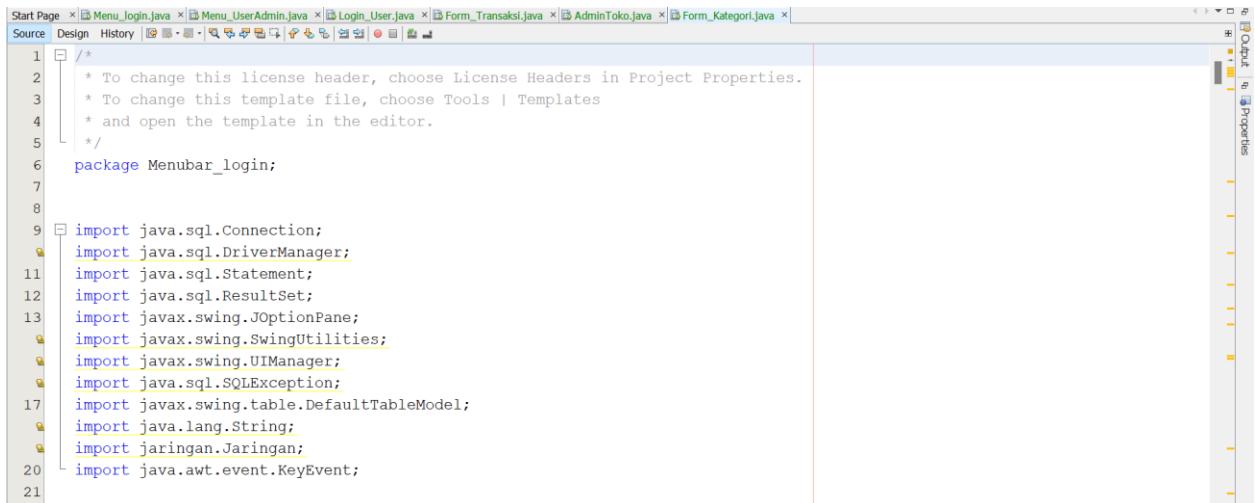
Halaman Kategori ini adalah halaman yang digunakan untuk melakukan CRUD pada tabel kategori dan yang mana kategori ini akan digunakan di halaman entry barang. Halaman utama user kasir ini cuma memiliki banyak menu dan 3 kolom (kolom kode, nama, dan search). Untuk membuat file jframe, anda bisa melihat pada ‘**Pembuatan Halaman Utama User Kasir**’. Untuk desain yang ada di halaman CRUD kategori ini, anda harus membuat desain seperti gambar 2.136 ini.

The diagram shows a window titled "FORM KATEGORI". On the left, there's a sidebar with buttons for "KODE", "NAMA", and "SEARCH". The main area has three input fields: "KODE", "NAMA", and "SEARCH". Below them are buttons for "Baru", "Save", "Edit", "Delete", and "Cancel". At the bottom, there's a table with columns "ID KATEGORI" and "NAMA KATEGORI", containing several rows of data. At the very bottom are buttons for "Refresh" and "Kembali".

Gambar 2.136 : Halaman Kategori

Jika desain sudah dibuat, jangan lupa untuk mengganti nama variable tiap elemen, agar mempermudah untuk diingat. Setelah itu buka tab Soure pada halaman ini, lalu buat sistemnya agar halaman utama user ini dapat berjalan dengan baik. Berikut cara membuat sistem untuk halaman kategori:

1. Import library yang dibutuhkan untuk sistem CRUD kategori.



```
1  /*
2  * To change this license header, choose License Headers in Project Properties.
3  * To change this template file, choose Tools | Templates
4  * and open the template in the editor.
5  */
6 package Menubar_login;
7
8
9 import java.sql.Connection;
10 import java.sql.DriverManager;
11 import java.sql.Statement;
12 import java.sql.ResultSet;
13 import javax.swing.JOptionPane;
14 import javax.swing.SwingUtilities;
15 import javax.swing.UIManager;
16 import java.sql.SQLException;
17 import javax.swing.table.DefaultTableModel;
18 import java.lang.String;
19 import jaringan.Jaringan;
20 import java.awt.event.KeyEvent;
21
```

Gambar 2.137 : Library yang dibutuhkan untuk sistem CRUD kategori

2. Buat beberapa variabel scope global



```
25  * @author adham PSN
26  */
27 public class Form_Kategori extends javax.swing.JFrame {
28
29     private Connection koneksi;
30     private Statement pernyataan;
31     private ResultSet hasil;
32     private String sql;
33     private String hasilPecahanChar;
34     private int cekTombol = 0;
35 }
```

Gambar 2.138 : Variabel scope global yang ada di halaman CRUD kategori

3. Constructor pada halaman ini, isi sesuai dengan gambar 2.139.



```
37 /**
38  * Creates new form Form_Kategori
39  */
40 public Form_Kategori() {
41     initComponents();
42
43     Kosong();
44     tidak_bisa_isi();
45     keyPressed();
46     placeHolder();
47
48     isi_table();
49 }
50
```

Gambar 2.139 : Constructor pada halaman CRUD kategori

4. Buat fungsi dengan nama kosong(), isi fungsi tersebut.

```
Start Page x Menu_Login.java x Menu_UserAdmin.java x Login_User.java x Form_Transaksi.java x AdminToko.java x Form_Kategori.java x
Source Design History 
```

51
52 private void Kosong() {
53
54 txtKode.setText("");
55 txtNama.setText("");
56
57 }
58

Gambar 2.140 : Fungsi Kosong() pada halaman CRUD kategori.

5. Buat fungsi dengan nama bisa_isi(), isi fungsi sesuai dengan gambar 2.138.

```
Start Page × Menu_Login.java × Menu_UserAdmin.java × Login_User.java × Form_Transaksi.java × AdminToko.java × Form_Kategori.java ×
Source Design History 
59
60     private void bisa_isi() {
61         txtKode.setEnabled(false);
62         txtNama.setEnabled(true);
63     }
64
65
```

Gambar 2.141 : Fungsi bisa_isi() pada halaman CRUD kategori

6. Buat fungsi dengan nama tidak_bisa_isi(), isi fungsi tersebut.

```
Start Page × Menu_Login.java × Menu_UserAdmin.java × Login_User.java × Form_Transaksi.java × AdminToko.java × Form_Kategori.java ×
Source Design History 
66
67     private void tidak_bisa_isi() {
68
69         txtKode.setEnabled(false);
70         txtNama.setEnabled(false);
71     }
72 
```

Gambar 2.142 : Fungsi tidak_bisa_isi() pada halaman CRUD kategori

7. Buat fungsi dengan nama tombol_mati(), isi fungsi tersebut.

```
73  
74     private void tombol_mati() {  
75  
76         btnSave.setText("Simpan");  
77         btnSave.setEnabled(false);  
78         btnDelete.setEnabled(false);  
79         btnEdit.setEnabled(false);  
80         btnBaru.setEnabled(false);  
81     }  
82 }
```

Gambar 2.143 : Fungsi tombol_mati() pada halaman CRUD kategori

8. Buat fungsi tombol_hidup(), isi fungsi tersebut.

Gambar 2.144 : Fungsi tombol_hidup() pada halaman CRUD kategori

9. Buat fungsi placeHolder(), dan isi fungsi tersebut.

```
92  
93  
94 □ private void placeHolder() {  
95  
96     txtKode.setToolTipText("TextField Kode");  
97     txtNama.setToolTipText("TextField Nama");  
98     txtSearch.setToolTipText("TextField Search");  
99 }  
100
```

Gambar 2.145 : Fungsi placeholder() di halaman CRUD kategori

10. Buat fungsi keyPressed(), isi fungsi tersebut.

```
Start Page × Menu_Login.java × Menu_UserAdmin.java × Login_User.java × Form_Transaksi.java × AdminToko.java × Form_Kategori.java ×
Source Design History ☰

101
102     private void keyPressed() {
103
104         btnBaru.setMnemonic(KeyEvent.VK_B);
105         btnSearch.setMnemonic(KeyEvent.VK_SLASH);
106         btnKembali.setMnemonic(KeyEvent.VK_F4);
107         btnCancel.setMnemonic(KeyEvent.VK_BACK_SPACE);
108         btnRefresh.setMnemonic(KeyEvent.VK_R);
109         btnDelete.setMnemonic(KeyEvent.VK_D);
110         btnEdit.setMnemonic(KeyEvent.VK_E);
111
112         if(btnSave.getText().equals("Simpan")) {
113
114             btnSave.setMnemonic(KeyEvent.VK_ENTER);
115         } else {
116
117             btnSave.setMnemonic(KeyEvent.VK_U);
118         }
119     }
120 }
```

Gambar 2.146 : Fungsi keyPressed() pada halaman CRUD kategori

11. Buat fungsi dengan nama Simpan(), isi fungsi tersebut.

The screenshot shows a Java IDE interface with multiple tabs at the top: Start Page, Menu_login.java, Menu_UserAdmin.java, Login_User.java, Form_Transaksi.java, AdminToko.java, and Form_Kategori.java. The main editor area displays the following Java code:

```
122     private void Simpan() {
123
124         // Jika tidak ada data yang dipilih
125         if(txtKode.getText().equals("") || txtNama.getText().equals("")) {
126
127             JOptionPane.showMessageDialog(null, "Tidak ada data yang dimasukkan / Data yang dimasukkan kurang lengkap! Silakan memasukkan data lagi");
128
129         } else {
130
131             try {
132
133                 sql = "insert into tb_kategori values"
134                 + "(" + txtKode.getText() + ", '" + txtNama.getText() + "')";
135
136                 koneksi = jaringan.Jaringan.config();
137                 pernyataan = koneksi.createStatement();
138                 pernyataan.execute(sql);
139
140             } catch (Exception e) {
141
142                 JOptionPane.showMessageDialog(null, "Keterangan Error :" + e);
143             }
144
145             isi_table();
146             tidak_bisa_isi();
147 }
```

Gambar 2.147 : Fungsi Simpan() pada halaman CRUD kategori (1)

The screenshot shows a Java IDE interface with multiple tabs at the top: Start Page, Menu_login.java, Menu_UserAdmin.java, Login_User.java, Form_Transaksi.java, AdminToko.java, and Form_Kategori.java. The main editor area displays the following Java code:

```
122     private void Simpan() {
123
124         // Jika tidak ada data yang dipilih
125         if(txtKode.getText().equals("") || txtNama.getText().equals("")) {
126
127             JOptionPane.showMessageDialog(null, "Tidak ada data yang dimasukkan / Data yang dimasukkan kurang lengkap! Silakan memasukkan data lagi");
128
129
130
131             sql = "insert into tb_kategori values"
132             + "(" + txtKode.getText() + ", '" + txtNama.getText() + "')";
133
134             koneksi = jaringan.Jaringan.config();
135             pernyataan = koneksi.createStatement();
136             pernyataan.execute(sql);
137
138
139         } catch (Exception e) {
140
141             JOptionPane.showMessageDialog(null, "Keterangan Error :" + e);
142
143         }
144
145
146
147 }
```

Gambar 2.148 : Fungsi Simpan() pada halaman CRUD kategori (2)

The screenshot shows a Java IDE interface with multiple tabs at the top: Start Page, Menu_Login.java, Menu_UserAdmin.java, Login_User.java, Form_Transaksi.java, AdminToko.java, and Form_Kategori.java. The main area displays the following Java code:

```
132     sql = "insert into tb_kategori values"
133         + "(" + txtKode.getText() + ", " + txtNama.getText() + ")";
134
135     koneksi = jaringan.Jaringan.config();
136     pernyataan = koneksi.createStatement();
137     pernyataan.execute(sql);
138
139
140 } catch (Exception e) {
141     JOptionPane.showMessageDialog(null, "Keterangan Error :" + e);
142 }
143
144
145 isi_table();
146 tidak_bisa_isi();
147
148
149 btnSave.setEnabled(false);
150 btnEdit.setEnabled(true);
151 btnDelete.setEnabled(true);
152 btnKembali.setEnabled(true);
153 btnBaru.setEnabled(true);
154
155
156
157 }
```

Gambar 2.149 : Fungsi Simpan() pada halaman CRUD kategori (3)

12. Buat fungsi dengan nama update(), dan isi fungsi tersebut.

The screenshot shows a Java IDE interface with multiple tabs at the top: Start Page, Menu_Login.java, Menu_UserAdmin.java, Login_User.java, Form_Transaksi.java, AdminToko.java, and Form_Kategori.java. The main area displays the following Java code:

```
156
157
158 private void update() {
159
160     // Jika tidak ada data yang dipilih
161     if(txtKode.getText().equals("") || txtNama.getText().equals("")) {
162
163         JOptionPane.showMessageDialog(null, "Tidak ada data yang dipilih! Silahkan masukkan data lagi");
164
165     } else {
166
167         try {
168
169             sql = "update tb_kategori"
170                 + " set nama_kategori = '" + txtNama.getText() + "'"
171                 + " where id_kategori = '" + txtKode.getText() + "'";
172
173             koneksi = jaringan.Jaringan.config();
174             pernyataan = koneksi.createStatement();
175             pernyataan.executeUpdate(sql);
176
177             JOptionPane.showMessageDialog(null, "Update Berhasil");
178
179             Kosong();
180         } catch (Exception e) {
181     }
```

Gambar 2.150 : Fungsi update() pada halaman CRUD kategori (1)

The screenshot shows a Java code editor with the following code snippet:

```
163     JOptionPane.showMessageDialog(null, "Tidak ada data yang dipilih! Silahkan masukkan data lagi");
164
165 } else {
166
167     try {
168
169         sql = "update tb_kategori"
170         + " set nama_kategori = '" + txtNama.getText() + "'"
171         + " where id_kategori = '" + txtKode.getText() + "'";
172
173         koneksi = jaringan.Jaringan.config();
174         pernyataan = koneksi.createStatement();
175         pernyataan.executeUpdate(sql);
176
177         JOptionPane.showMessageDialog(null, "Update Berhasil");
178
179         Kosong();
180     } catch (Exception e) {
181
182         JOptionPane.showMessageDialog(null, "Keterangan Error :" + e);
183     }
184
185     isi_table();
186 }
187
188 }
```

Gambar 2.151 : Fungsi update() pada halaman CRUD kategori (2)

13. Buat fungsi dengan nama hapus(), isi fungsi tersebut.

The screenshot shows a Java code editor with the following code snippet:

```
190 private void hapus() {
191
192     // Jika tidak ada data yang dipilih
193     if(txtKode.getText().equals("") || txtNama.getText().equals("")) {
194
195         JOptionPane.showMessageDialog(null, "Tidak ada data yang dipilih! Silahkan memilih data lagi");
196
197     } else {
198
199         int dialogButton = JOptionPane.YES_NO_OPTION;
200         int dialogResult = JOptionPane.showConfirmDialog(this, "Data Akan di Hapus", "Hapus Data", dialogButton);
201
202         if(dialogResult != 0) {
203
204             JOptionPane.showMessageDialog(null, "Batal di Hapus");
205         } else {
206
207             try {
208
209                 sql = "DELETE |"
210                 + " FROM tb_kategori"
211                 + " WHERE id_kategori LIKE '%" + txtKode.getText() + "%' AND nama_kategori LIKE '%" + txtNama.g
212
213                 koneksi = jaringan.Jaringan.config();
214                 pernyataan = koneksi.createStatement();
215                 pernyataan.execute(sql);
216
217             }
218
219         }
220
221     }
222 }
```

Gambar 2.152 : Fungsi hapus() pada halaman CRUD kategori (1)

A screenshot of a Java IDE showing the code for a delete operation. The code is part of a method named 'hapus()' in a class. It first checks if the selected item is empty. If it is, it shows a message dialog. If not, it shows a confirmation dialog. If confirmed, it executes a DELETE SQL query on the 'tb_kategori' table where the 'id_kategori' matches the selected value and the 'nama_kategori' contains the search term from the 'txtNama' text field. The code uses JDBC to connect to a database.

```
190
191     a yang dipilih
192     equals("") || txtNama.getText().equals("")) {
193
194     JOptionPane.showMessageDialog(null, "Tidak ada data yang dipilih! Silahkan memilih data lagi");
195
196
197
198
199     = JOptionPane.YES_NO_OPTION;
200     = JOptionPane.showConfirmDialog(this, "Data Akan di Hapus", "Hapus Data", dialogButton);
201
202     = 0) {
203
204     showMessageDialog(null, "Batal di Hapus");
205
206
207
208
209     LETE" |
210     " FROM tb_kategori"
211     " WHERE id_kategori LIKE '%" + txtKode.getText() + "%' AND nama_kategori LIKE '%" + txtNama.getText() + "%'";
212
213     jaringan.Jaringan.config();
214     n = koneksi.createStatement();
215     n.execute(sql);
```

Gambar 2.153 : Fungsi hapus() pada halaman CRUD kategori (2)

A screenshot of a Java IDE showing the continuation of the 'hapus()' method. It handles the case where the user confirms the deletion. It constructs a DELETE SQL query with the same WHERE clause as before. It then connects to the database using a connection object 'koneksi' and a statement object 'pernyataan'. It executes the query and shows a success message dialog. Finally, it calls two helper methods: 'isi_table()' and 'Kosong()'.

```
203
204         JOptionPane.showMessageDialog(null, "Batal di Hapus");
205     } else {
206
207         try {
208
209             sql = "DELETE"
210             + " FROM tb_kategori"
211             + " WHERE id_kategori LIKE '%" + txtKode.getText() + "%' AND nama_kategori LIKE '%" + txtNama.get
212
213             koneksi = jaringan.Jaringan.config();
214             pernyataan = koneksi.createStatement();
215             pernyataan.execute(sql);
216
217             JOptionPane.showMessageDialog(null, "Berhasil di Hapus");
218
219             isi_table();
220             Kosong();
221         } catch(Exception e) {
222
223             JOptionPane.showMessageDialog(null, "Keterangan Error :" + e);
224         }
225     }
226 }
227 }
```

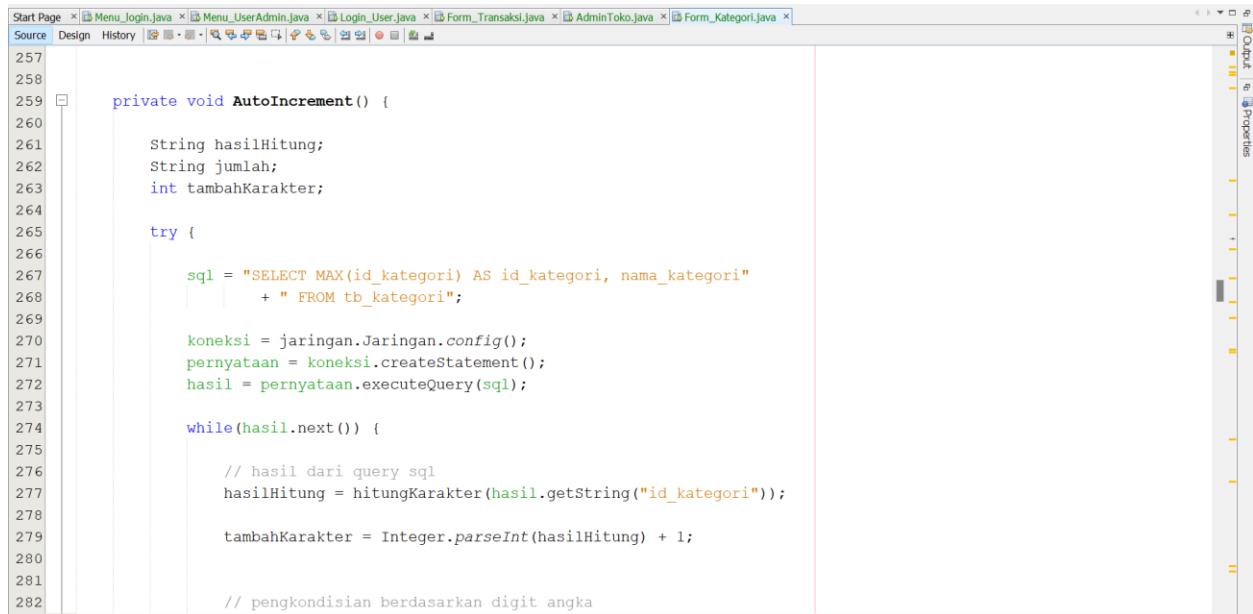
Gambar 2.154 : Fungsi hapus() pada halaman CRUD kategori (3)

14. Buat fungsi dengan nama isi_table(), isi fungsi tersebut.

Gambar 2.155 : Fungsi isi_table() pada halaman CRUD kategori (1)

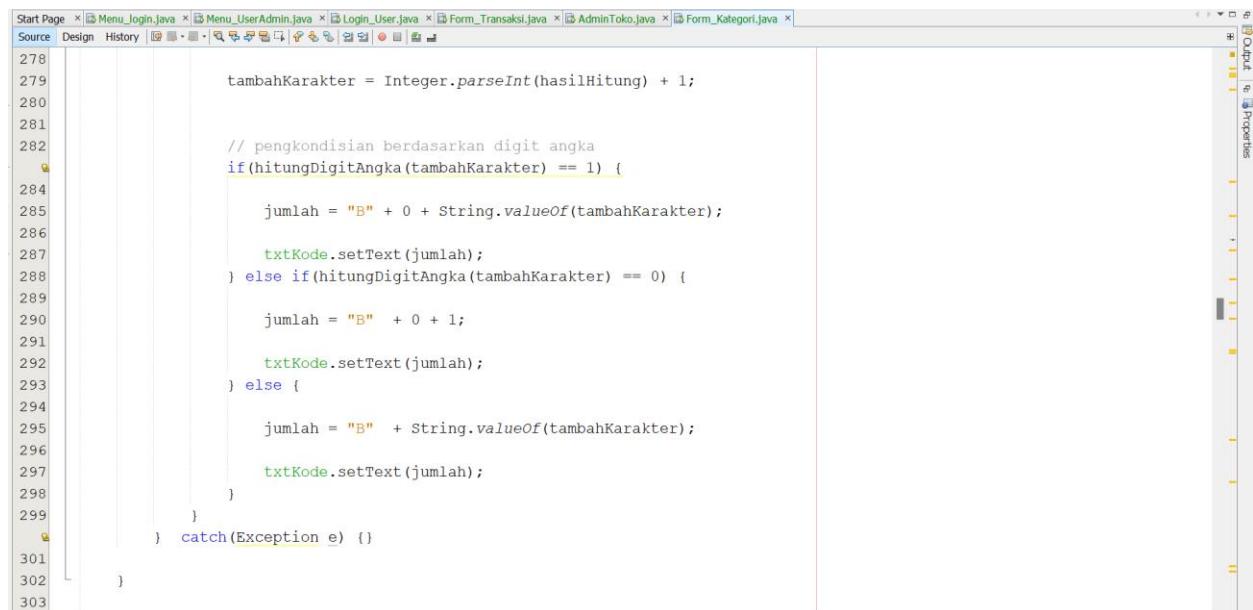
Gambar 2.156 : Fungsi isi_table() pada halaman CRUD kategori (2)

15. Buat fungsi dengan nama AutoIncrement(), isi fungsi tersebut.



```
Start Page | Menu_Login.java | Menu_UserAdmin.java | Login_User.java | Form_Transaksi.java | AdminToko.java | Form_Kategori.java | 
Source Design History | 
257
258
259  private void AutoIncrement() {
260
261     String hasilHitung;
262     String jumlah;
263     int tambahKarakter;
264
265     try {
266
267         sql = "SELECT MAX(id_kategori) AS id_kategori, nama_kategori"
268             + " FROM tb_kategori";
269
270         koneksi = jaringan.Jaringan.config();
271         pernyataan = koneksi.createStatement();
272         hasil = pernyataan.executeQuery(sql);
273
274         while(hasil.next()) {
275
276             // hasil dari query sql
277             hasilHitung = hitungKarakter(hasil.getString("id_kategori"));
278
279             tambahKarakter = Integer.parseInt(hasilHitung) + 1;
280
281             // pengkondision berdasarkan digit angka
282         }
283     } catch (Exception e) {}
284
285     tambahKarakter = Integer.parseInt(hasilHitung) + 1;
286
287     // pengkondision berdasarkan digit angka
288     if(hitungDigitAngka(tambahKarakter) == 1) {
289
290         jumlah = "B" + 0 + String.valueOf(tambahKarakter);
291         txtKode.setText(jumlah);
292     } else if(hitungDigitAngka(tambahKarakter) == 0) {
293
294         jumlah = "B" + 0 + 1;
295         txtKode.setText(jumlah);
296     } else {
297
298         jumlah = "B" + String.valueOf(tambahKarakter);
299         txtKode.setText(jumlah);
300     }
301 }
302 }
```

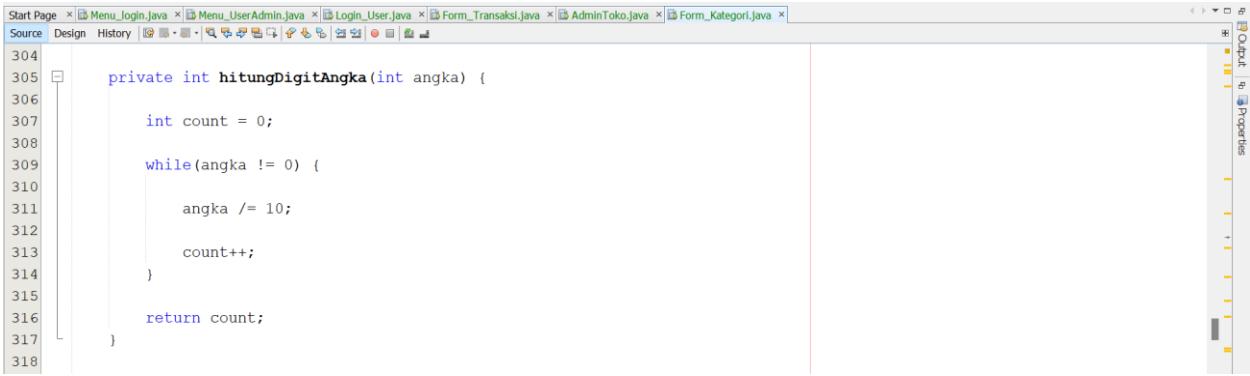
Gambar 2.157 : Fungsi AutoIncrement() pada halaman CRUD kategori (1)



```
Start Page | Menu_Login.java | Menu_UserAdmin.java | Login_User.java | Form_Transaksi.java | AdminToko.java | Form_Kategori.java | 
Source Design History | 
278
279     tambahKarakter = Integer.parseInt(hasilHitung) + 1;
280
281     // pengkondision berdasarkan digit angka
282     if(hitungDigitAngka(tambahKarakter) == 1) {
283
284         jumlah = "B" + 0 + String.valueOf(tambahKarakter);
285         txtKode.setText(jumlah);
286     } else if(hitungDigitAngka(tambahKarakter) == 0) {
287
288         jumlah = "B" + 0 + 1;
289         txtKode.setText(jumlah);
290     } else {
291
292         jumlah = "B" + String.valueOf(tambahKarakter);
293         txtKode.setText(jumlah);
294     }
295 }
296 } catch (Exception e) {}
297 }
```

Gambar 2.158 : Fungsi AutoIncrement() pada halaman CRUD kategori (2)

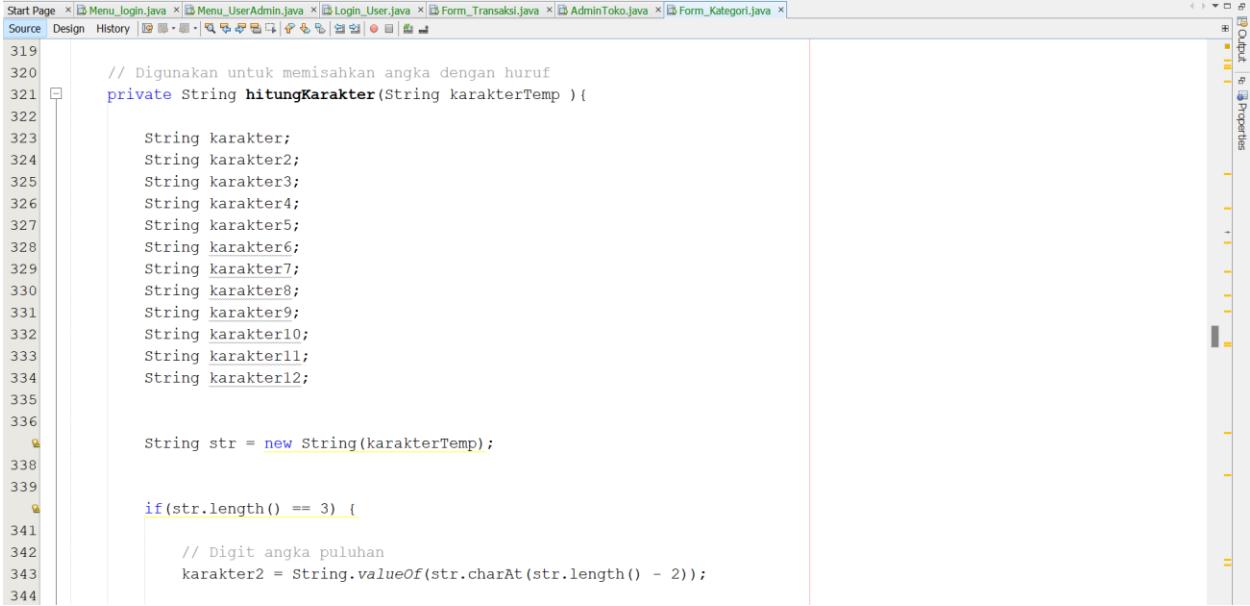
16. Buat fungsi dengan nama hitungDigitAngka(), isi fungsi tersebut.



```
Start Page x Menu_login.java x Menu_UserAdmin.java x Login_User.java x Form_Transaksi.java x AdminToko.java x Form_Kategori.java x
Source Design History
304     private int hitungDigitAngka(int angka) {
305
306         int count = 0;
307
308         while(angka != 0) {
309
310             angka /= 10;
311
312             count++;
313
314         }
315
316         return count;
317     }
318 }
```

Gambar 2.159 : Fungsi hitungDigitAngka() pada halaman CRUD kategori

17. Buat fungsi hitungKarakter(), isi fungsi tersebut.



```
Start Page x Menu_login.java x Menu_UserAdmin.java x Login_User.java x Form_Transaksi.java x AdminToko.java x Form_Kategori.java x
Source Design History
319
320     // Digunakan untuk memisahkan angka dengan huruf
321     private String hitungKarakter(String karakterTemp) {
322
323         String karakter;
324         String karakter2;
325         String karakter3;
326         String karakter4;
327         String karakter5;
328         String karakter6;
329         String karakter7;
330         String karakter8;
331         String karakter9;
332         String karakter10;
333         String karakter11;
334         String karakter12;
335
336
337         String str = new String(karakterTemp);
338
339
340         if(str.length() == 3) {
341
342             // Digit angka puluhan
343             karakter2 = String.valueOf(str.charAt(str.length() - 2));
344         }
345     }
346 }
```

Gambar 2.160 : Fungsi hitungKarakter() pada halaman CRUD kategori (1)

The screenshot shows a Java code editor with several tabs at the top: Start Page, Menu_Login.java, Menu_UserAdmin.java, Login_User.java, Form_Transaksi.java, AdminToko.java, and Form_Kategori.java. The main pane displays Java code for calculating character values based on string length. The code uses if-else statements to handle lengths of 1, 2, 3, 4, and 5. It extracts characters from the end of the string and concatenates them in specific ways to represent digit groups. The code editor interface includes toolbars and panels for code navigation and properties.

```
345     // Digit angka satuan
346     karakter = String.valueOf(str.charAt(str.length() - 1));
347
348     // Penggabungan karakter 1 dan 2
349     this.hasilPecahanChar = karakter2 + karakter;
350 } else if(str.length() == 4) {
351
352     // Digit angka ratusan
353     karakter3 = String.valueOf(str.charAt(str.length() - 3));
354
355     // Digit angka puluhan
356     karakter2 = String.valueOf(str.charAt(str.length() - 2));
357
358     // Digit angka satuan
359     karakter = String.valueOf(str.charAt(str.length() - 1));
360
361     // Penggabungan karakter 1, 2, 3
362     this.hasilPecahanChar = karakter3 + karakter2 + karakter;
363 } else if(str.length() == 5) {
364
365     // Digit angka ribuan
366     karakter4 = String.valueOf(str.charAt(str.length() - 4));
367
368     // Digit angka ratusan
369     karakter3 = String.valueOf(str.charAt(str.length() - 3));
370 }
```

Gambar 2.161 : Fungsi hitungKarakter() pada halaman CRUD kategori (2)

The screenshot shows a Java code editor with several tabs at the top: Start Page, Menu_Login.java, Menu_UserAdmin.java, Login_User.java, Form_Transaksi.java, AdminToko.java, and Form_Kategori.java. The main pane displays Java code for calculating character values based on string length. The code uses if-else statements to handle lengths of 1, 2, 3, 4, and 5. It extracts characters from the end of the string and concatenates them in specific ways to represent digit groups. The code editor interface includes toolbars and panels for code navigation and properties.

```
371     // Digit angka puluhan
372     karakter2 = String.valueOf(str.charAt(str.length() - 2));
373
374     // Digit angka satuan
375     karakter = String.valueOf(str.charAt(str.length() - 1));
376
377     // Penggabungan karakter 1, 2, 3, 4
378     this.hasilPecahanChar = karakter4 + karakter3 + karakter2 + karakter;
379 } else if(str.length() == 6) {
380
381     // Digit angka puluhan ribu
382     karakter5 = String.valueOf(str.charAt(str.length() - 5));
383
384     // Digit angka ribuan
385     karakter4 = String.valueOf(str.charAt(str.length() - 4));
386
387     // Digit angka ratusan
388     karakter3 = String.valueOf(str.charAt(str.length() - 3));
389
390     // Digit angka puluhan
391     karakter2 = String.valueOf(str.charAt(str.length() - 2));
392
393     // Digit angka satuan
394     karakter = String.valueOf(str.charAt(str.length() - 1));
395
396     // Penggabungan karakter 1, 2, 3, 4, 5
```

Gambar 2.162 : Fungsi hitungKarakter() pada halaman CRUD kategori (3)

The screenshot shows a Java code editor with the following code:

```
377     // Penggabungan karakter 1, 2, 3, 4
378     this.hasilPecahanChar = karakter4 + karakter3 + karakter2 + karakter;
379 } else if(str.length() == 6) {
380
381     // Digit angka puluhan ribu
382     karakter5 = String.valueOf(str.charAt(str.length() - 5));
383
384     // Digit angka ribuan
385     karakter4 = String.valueOf(str.charAt(str.length() - 4));
386
387     // Digit angka ratusan
388     karakter3 = String.valueOf(str.charAt(str.length() - 3));
389
390     // Digit angka puluhan
391     karakter2 = String.valueOf(str.charAt(str.length() - 2));
392
393     // Digit angka satuan
394     karakter = String.valueOf(str.charAt(str.length() - 1));
395
396     // Penggabungan karakter 1, 2, 3, 4, 5
397     this.hasilPecahanChar = karakter5 + karakter4 + karakter3 + karakter2 + karakter;
398 }
399
400     return hasilPecahanChar;
401 }
402 }
```

Gambar 2.163 : Fungsi hitungKarakter() pada halaman CRUD kategori (4)

18. Buat fungsi Search(), isi fungsi sesuai tersebut.

The screenshot shows a Java code editor with the following code:

```
404     private void Search() {
405
406         // Jika TextField search kosong
407         if(txtSearch.getText().equals("")){
408
409             txtSearch.requestFocus();
410
411         } else {
412
413             DefaultTableModel tbl = new DefaultTableModel();
414             tbl.addColumn("id_kategori");
415             tbl.addColumn("nama_kategori");
416             tabelKategori.setModel(tbl);
417
418             try {
419
420                 sql = "select *"
421                     + " from tb_kategori"
422                     + " where nama_kategori like '%" + txtSearch.getText() + "%' || id_kategori = '" + txtSearch.getText();
423
424                 koneksi = jaringan.Jaringan.config();
425                 pernyataan = koneksi.createStatement();
426                 hasil = pernyataan.executeQuery(sql);
427
428                 while (hasil.next()) {
429
430
431             }
432
433         }
434     }
435 }
```

Gambar 2.164 : Fungsi Search() pada halaman CRUD kategori (1)

```
Start Page × Menu_Login.java × Menu_UserAdmin.java × Login_User.java × Form_Transaksi.java × AdminToko.java × Form_Kategori.java ×
Source Design History | Back Forward Find Replace Properties
419
420     sql = "select *"
421         + " from tb_kategori"
422         + " where nama_kategori like '%" + txtSearch.getText() + "%' || id_kategori = '" + txtSearch.getText()
423
424     koneksi = jaringan.Jaringan.config();
425     pernyataan = koneksi.createStatement();
426     hasil = pernyataan.executeQuery(sql);
427
428     while (hasil.next()) {
429
430         tb1.addRow(new Object[] {
431
432             hasil.getString("id_kategori"),
433             hasil.getString("nama_kategori"),
434         });
435
436         tabelKategori.setModel(tb1);
437
438     }
439
440     txtKode.setEnabled(false);
441
442 } catch(Exception e) {}
443
444
```

Gambar 2.165 : Fungsi Search() pada halaman CRUD kategori (2)

19. Buat fungsi FunctionEdit(), isi fungsi tersebut.

```
Projects Navigator
445
446 private void functionEdit() {
447
448     if(txtKode.getText().equals("") || txtNama.getText().equals("")) {
449
450         JOptionPane.showMessageDialog(null, "Tidak ada data yang dipilih! Silahkan memilih data lagi");
451     } else {
452
453         bisa_isi();
454         tombol_mati();
455
456         txtNama.requestFocus();
457         btnSave.setText("Update");
458         btnSave.setEnabled(true);
459         txtKode.setEnabled(false);
460     }
461
462 }
463
```

Gambar 2.166 : Fungsi FunctionEdit() pada halaman CRUD kategori

20. Buat fungsi dengan nama RefreshHalaman(), isi fungsi tersebut.

```
Start Page × Menu_Login.java × Menu_UserAdmin.java × Login_User.java × Form_Transaksi.java × AdminToko.java × Form_Kategori.java ×
Source Design History | Back Forward Find Replace Properties
464
465 public void refreshHalaman() {
466
467     dispose();
468
469     new Form_Kategori().setVisible(true);
470 }
471
472
```

Gambar 2.167 : Fungsi Refresh Halaman pada halaman CRUD Kategori

21. Buat fungsi event. Fungsi event ini digunakan agar element pada halaman jframe dapat berjalan sesuai dengan event yang anda tentukan. Untuk membuat event, anda bisa ke tab design => tekan kanan mouse pada elemen yang ingin kasih event => pilih menu event => pilih event yang anda inginkan. Event yang tersedia untuk elemen itu bermacam – macam, sesuaikan dengan kebutuhan anda. Berikut beberapa contoh event yang digunakan pada halaman form CRUD kategori:

Gambar 2.168 : Fungsi Event dan isinya pada halaman CRUD Kategori (1)

Gambar 2.169 : Fungsi Event dan isinya pada halaman CRUD Kategori (2)

The screenshot shows a Java IDE interface with multiple tabs at the top: Start Page, Menu_login.java, Menu_UserAdmin.java, Login_User.java, Form_Transaksi.java, AdminToko.java, and Form_Kategori.java. The main area displays Java code for the Form_Kategori.java file. The code includes several event handlers:

```
748  
749     private void btnCancelActionPerformed(java.awt.event.ActionEvent evt) {  
750         // TODO add your handling code here:  
751  
752         JOptionPane.showMessageDialog(null, "Batal");  
753  
754         Kosong();  
755         tidak_bisa_isi();  
756         tombol_hidup();  
757  
758         btnKembali.setEnabled(true);  
759     }  
760  
761     private void txtKodeActionPerformed(java.awt.event.ActionEvent evt) {  
762         // TODO add your handling code here:  
763     }  
764  
765     private void btnKembaliActionPerformed(java.awt.event.ActionEvent evt) {  
766         // TODO add your handling code here:  
767  
768         dispose();  
769  
770         new AdminToko().setVisible(true);  
771     }  
772  
773 }
```

Gambar 2 170 : Fungsi Event dan isinya pada halaman CRUD Kategori (3)

The screenshot shows a Java IDE interface with multiple tabs at the top: Start Page, Menu_login.java, Menu_UserAdmin.java, Login_User.java, Form_Transaksi.java, AdminToko.java, and Form_Kategori.java. The main area displays Java code for the Form_Kategori.java file. The code includes an event handler for the btnBaru button:

```
774  
775     private void btnBaruActionPerformed(java.awt.event.ActionEvent evt) {  
776         // TODO add your handling code here:  
777  
778         bisa_isi();  
779         Kosong();  
780  
781         txtKode.setEnabled(false);  
782         txtNama.requestFocus();  
783  
784         btnEdit.setEnabled(false);  
785         btnDelete.setEnabled(false);  
786         btnKembali.setEnabled(false);  
787         btnBaru.setEnabled(false);  
788  
789         btnSave.setText("Simpan");  
790         btnSave.setEnabled(true);  
791  
792         AutoIncrement();  
793         keyPressed();  
794     }  
795 }
```

Gambar 2.171 : Fungsi Event dan isinya pada halaman CRUD Kategori (4)

The screenshot shows a Java IDE interface with multiple tabs at the top: Start Page, Menu_login.java, Menu_UserAdmin.java, Login_User.java, Form_Transaksi.java, AdminToko.java, and Form_Kategori.java. The main area displays Java code for the Form_Kategori.java file. The code includes an event handler for the btnRefresh button:

```
796  
797     private void btnRefreshActionPerformed(java.awt.event.ActionEvent evt) {  
798         // TODO add your handling code here:  
799  
800         refreshHalaman();  
801     }  
802 }
```

Gambar 2.172 : Fungsi Event dan isinya pada halaman CRUD Kategori (5)

N. Pembuatan Halaman Entry Barang

Halaman Entry Barang ini adalah halaman yang digunakan untuk melakukan CRUD pada tabel data barang dan yang mana data barang ini akan digunakan di halaman transaksi. Halaman input barang ini memiliki beberapa menu dan kolom untuk kebutuhan CRUD barang. Untuk membuat file jframe, anda bisa melihat pada ‘**Pembuatan Halaman Utama User Kasir**’. Untuk desain yang ada di halaman entry barang ini, anda harus membuat desain seperti gambar 2.173 ini.

The image shows a software interface titled 'Design Preview [Entry_Barang]'. The main title bar is purple with the text 'FORM INPUT BARANG' in white. Below the title, there are five input fields labeled 'KODE', 'NAMA', 'HARGA', 'STOCK', and 'KATEGORI'. Each label is followed by a white input box. Below these fields is a grey toolbar containing buttons for 'BARU', 'Save', 'EDIT', 'HAPUS', and 'RESET'. Underneath the toolbar is a search section with a 'SEARCH' input field and a 'SEARCH' button. At the bottom of the window is a table with columns 'KODE', 'NAMA', 'HARGA', 'STOCK', and 'KATEGORI'. The table has several empty rows. At the very bottom are three buttons: 'REFRESH', 'TAMBAH STOCK', and 'BACK'.

Gambar 2.173 : Halaman Entry Barang

Jika desain sudah dibuat, jangan lupa untuk mengganti nama variable tiap elemen, agar mempermudah untuk diingat. Setelah itu buka tab Soure pada halaman ini, lalu buat

sistemnya agar halaman utama user ini dapat berjalan dengan baik. Berikut cara membuat sistem untuk halaman entry barang:

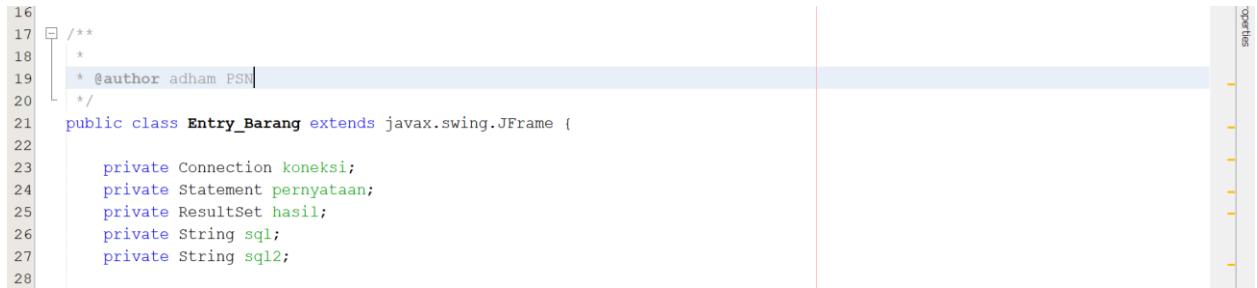
1. Import library yang dibutuhkan untuk sistem CRUD entry barang.



```
6 package Menubar_login;
7
8 import java.sql.Connection;
9 import java.sql.Statement;
10 import java.sql.ResultSet;
11 import javax.swing.JOptionPane;
12 import javax.swing.table.DefaultTableModel;
13 import javax.swing.DefaultComboBoxModel;
14 import java.awt.event.KeyEvent;
15 import jaringan.Jaringan;
16
```

Gambar 2.174 : Library yang dibutuhkan untuk sistem CRUD entry barang

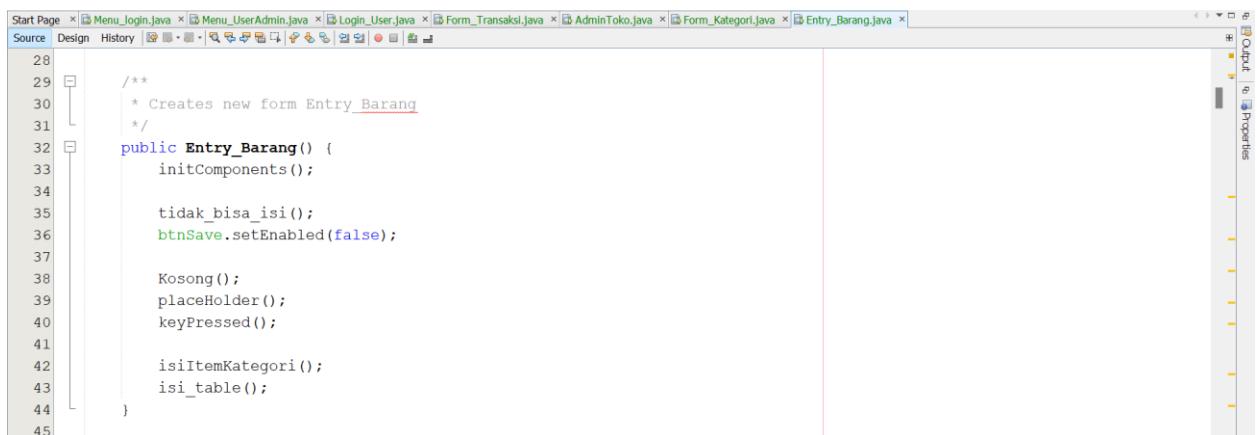
2. Buat beberapa variabel scope global



```
16
17 /**
18 * @author adham PSN
19 */
20 public class Entry_Barang extends javax.swing.JFrame {
21
22     private Connection koneksi;
23     private Statement pernyataan;
24     private ResultSet hasil;
25     private String sql;
26     private String sql2;
27
28 }
```

Gambar 2.175 : Beberapa variabel scope global pada halaman entry barang

3. Isi constructor, sesuai dengan gambar 2.176.



```
Start Page | Menu_Login.java | Menu_UserAdmin.java | Login_User.java | Form_Transaksi.java | AdminToko.java | Form_Kategori.java | Entry_Barang.java | 
Source Design History | 
28
29 /**
30 * Creates new form Entry_Barang
31 */
32 public Entry_Barang() {
33     initComponents();
34
35     tidak_bisa_isi();
36     btnSave.setEnabled(false);
37
38     Kosong();
39     placeHolder();
40     keyPressed();
41
42     isiItemKategori();
43     isi_table();
44 }
45
```

Gambar 2.176 : Isi dari Constructor pada halaman entry barang

4. Buat fungsi Kosong(), isi fungsi tersebut.

```
Start Page x Menu_login.java x Menu_UserAdmin.java x Login_User.java x Form_Transaksi.java x AdminToko.java x Form_Kategori.java x Entry_Barang.java x
Source Design History 
46     private void Kosong() {
47
48         txtSearch.setText("");
49         txtKode.setText("");
50         txtNama.setText("");
51         txtHarga.setText("");
52         txtStock.setText("");
53         cmbKategori.setSelectedItem(null);
54     }

```

Gambar 2.177 : Fungsi Kosong() pada halaman entry barang

5. Buat fungsi bisa_isi(), isi fungsi tersebut.

```
56     private void bisa_isi() {
57
58         txtSearch.setEnabled(true);
59         txtKode.setEnabled(false);
60         txtNama.setEnabled(true);
61         txtHarga.setEnabled(true);
62         txtStock.setEnabled(true);
63         cmbKategori.setEnabled(true);
64     }
65 }
```

Gambar 2.178 : Fungsi bisa_isi() pada halaman entry barang

6. Buat fungsi tidak_bisi_isi(), isi fungsi tersebut.

```
Start Page × Menu_Login.java × Menu_UserAdmin.java × Login_User.java × Form_Transaksi.java × AdminToko.java × Form_Kategori.java × Entry_Barang.java ×
Source Design History 


```

65
66 private void tidak_bisa_isi() {
67 txtKode.setEnabled(false);
68 txtNama.setEnabled(false);
69 txtHarga.setEnabled(false);
70 txtStock.setEnabled(false);
71 cmbKategori.setEnabled(false);
72 }
73

```


```

Gambar 2.179 : Fungsi tidak_bisa_isi() di halaman entry barang

7. Buat fungsi tombol_mati(), isi fungsi tersebut.

```
74     private void tombol_mati() {
75
76         btnSave.setText("Simpan");
77         btnSave.setEnabled(false);
78         btnEdit.setEnabled(false);
79         btnHapus.setEnabled(false);
80         btnSearch.setEnabled(true);
81     }
82 }
```

Gambar 2.180 : Fungsi tombol_mati() di halaman entry barang

8. Buat fungsi tombol_hidup(), isi fungsi tersebut.

```
Start Page | Menu_login.java | Menu_UserAdmin.java | Login_User.java | Form_Transaksi.java | AdminToko.java | Form_Kategori.java | Entry Barang.java
Source Design History
83     private void tombol_hidup() {
84
85         btnSave.setText("Simpan");
86         btnBaru.setEnabled(true);
87         btnSave.setEnabled(false);
88         btnEdit.setEnabled(true);
89         btnHapus.setEnabled(true);
90         btnSearch.setEnabled(true);
91         btnReset.setEnabled(true);
92         btnBack.setEnabled(true);
93         btnTambahStock.setEnabled(true);
94     }
```

Gambar 2.181 : Fungsi tombol_hidup() pada halaman entry barang

9. Buat fungsi placeHolder(), isi fungsi tersebut.

```
Start Page | Menu_login.java | Menu_UserAdmin.java | Login_User.java | Form_Transaksi.java | AdminToko.java | Form_Kategori.java | Entry Barang.java
Source Design History
96     private void placeHolder() {
97
98         txtKode.setToolTipText("TextField Kode");
99         txtNama.setToolTipText("TextField Nama");
100        txtHarga.setToolTipText("TextField Harga");
101        txtStock.setToolTipText("TextField Stock");
102        cmbKategori.setToolTipText("ComboBox Kategori");
103        txtSearch.setToolTipText("TextField Search");
104    }
105}
```

Gambar 2.182 : Fungsi placeHolder() pada halaman entry barang

10. Buat fungsi keyPressed(), isi fungsi tersebut.

```
Start Page | Menu_login.java | Menu_UserAdmin.java | Login_User.java | Form_Transaksi.java | AdminToko.java | Form_Kategori.java | Entry Barang.java
Source Design History
105
106     private void keyPressed() {
107
108         btnBaru.setMnemonic(KeyEvent.VK_B);
109         btnSearch.setMnemonic(KeyEvent.VK_SLASH);
110         btnRefresh.setMnemonic(KeyEvent.VK_R);
111         btnBack.setMnemonic(KeyEvent.VK_F4);
112         btnReset.setMnemonic(KeyEvent.VK_BACK_SPACE);
113         btnHapus.setMnemonic(KeyEvent.VK_D);
114         btnEdit.setMnemonic(KeyEvent.VK_E);
115         btnTambahStock.setMnemonic(KeyEvent.VK_T);
116
117         if (btnSave.getText().equals("Simpan")) {
118
119             btnSave.setMnemonic(KeyEvent.VK_S);
120         } else {
121
122             btnSave.setMnemonic(KeyEvent.VK_U);
123         }
124     }
125
126 }
```

Gambar 2.183 : Fungsi keyPressed() pada halaman entry barang

11. Buat fungsi Simpan, isi fungsi tersebut.

```
126
127     private void Simpan() {
128
129         // Jika tidak ada data yang dimasukkan
130         if (txtKode.getText().equals("") || txtNama.getText().equals(""))
131             || txtHarga.getText().equals("") || txtStock.getText().equals("")
132             || cmbKategori.getSelectedItem().equals("")) {
133
134             JOptionPane.showMessageDialog(null, "Tidak ada data yang dimasukkan / Data yang dimasukkan tidak sesuai dengan"
135
136             Kosong();
137             tidak_bisa_isi();
138             tombol_hidup();
139         } else {
140
141             if ((txtHarga.getText().equals("-") || txtHarga.getText().equals("."))
142                 || txtHarga.getText().equals(",") || txtHarga.getText().equals("_")
143                 || txtHarga.getText().equals("k") || txtHarga.getText().equals("K"))
144                 && (txtStock.getText().equals("-") || txtStock.getText().equals("."))
145                 || txtStock.getText().equals(",") || txtStock.getText().equals("_")
146                 || txtStock.getText().equals("k") || txtStock.getText().equals("K")) {
147
148                 JOptionPane.showMessageDialog(null, "Format Angka yang Anda Masukkan Kurang Tepat!");
149             } else {
150
151                 try {
```

Gambar 2.184 : Fungsi Simpan() di halaman entry barang (1)

```
Start Page x Menu_Login.java x Menu_UserAdmin.java x Login_User.java x Form_Transaksi.java x AdminToko.java x Form_Kategori.java x Entry_Barang.java x
Source Design History 
126
127
128
129
130     als("")  
    ext().equals("")  

132
133
134     ang dimasukkan / Data yang dimasukkan tidak sesuai dengan format yang dibutuhkan! Silahkan masukkan data lagi");  

135
136
137
138
139
140
141     xt().equals(".")  
    .getText().equals(" ")  
    .getText().equals("K")))  

142     K.getText().equals(".")  
    .getText().equals(" ")  
    .getText().equals("K")))  

143
144
145
146
147
148     yang Anda Masukkan Kurang Tepat!");  

149
150
151
```

Gambar 2.185 : Fungsi Simpan() di halaman entry barang (2)

```

Start Page | Menu_Login.java | Menu_UserAdmin.java | Login_User.java | Form_Transaksi.java | AdminToko.java | Form_Kategori.java | Entry_Barang.java
Source Design History | Back Forward Stop Refresh Properties

150
151
152
153     try {
154
155         sql = "insert into tb_barang(kode, nama, harga, stok, kategori) values"
156             + "(" + txtKode.getText() + ", '" + txtNama.getText() + "', " + txtHarga.getText() + ", " + txt
157
158         koneksi = jaringan.Jaringan.config();
159         pernyataan = koneksi.createStatement();
160         pernyataan.execute(sql);
161
162         tidak_bisa_isi();
163         isi_table();
164
165         btnSave.setEnabled(false);
166     } catch (Exception e) {
167
168         JOptionPane.showMessageDialog(null, "Keterangan Error :" + e);
169
170         Kosong();
171         tombol_hidup();
172
173     }
174
175

```

Gambar 2.186 : Fungsi Simpan() di halaman entry barang (3)

12. Buat fungsi Update(), isi fungsi tersebut.

```

Start Page | Menu_Login.java | Menu_UserAdmin.java | Login_User.java | Form_Transaksi.java | AdminToko.java | Form_Kategori.java | Entry_Barang.java
Source Design History | Back Forward Stop Refresh Properties

175
176     private void update() {
177
178         // Jika tidak ada data yang dimasukkan
179         if (txtKode.getText().equals(""))
180             || txtNama.getText().equals("") || txtHarga.getText().equals("")
181             || txtStock.getText().equals("") || cmbKategori.getSelectedItem().equals(""))
182
183             JOptionPane.showMessageDialog(null, "Tidak ada data yang dimasukkan / Data yang dimasukkan tidak sesuai dengan");
184
185         } else {
186
187             if ((txtHarga.getText().equals("-") || txtHarga.getText().equals("."))
188                 || txtHarga.getText().equals(",") || txtHarga.getText().equals(" "))
189                 || txtHarga.getText().equals("k") || txtHarga.getText().equals("R"))
190                 && (txtStock.getText().equals("-") || txtStock.getText().equals("."))
191                 || txtStock.getText().equals(",") || txtStock.getText().equals(" "))
192                 || txtStock.getText().equals("k") || txtStock.getText().equals("R"))
193
194             JOptionPane.showMessageDialog(null, "Format Angka yang Anda Masukkan Kurang Tepat!");
195
196         } else {
197
198             try {
199
200                 sql = "UPDATE tb_barang"
201                     + " SET nama = '" + txtNama.getText() + "', harga = " + txtHarga.getText() + ", stok = " + txt
202
203             }
204
205         }
206
207

```

Gambar 2.187 : Fungsi Update() di halaman entry barang (1)

The screenshot shows a Java IDE interface with multiple tabs at the top: Start Page, Menu_login.java, Menu_UserAdmin.java, Login_User.java, Form_Transaksi.java, AdminToko.java, Form_Kategori.java, and Entry_Barang.java. The main area displays Java code for the Entry_Barang.java class. The code includes several conditional statements (als("...").equals("...")) and a try-catch block. A red vertical line highlights a portion of the code between lines 176 and 200.

```
175
176
177
178
179
180     als("")  
181     edItem().equals("")) {  
182  
183         sukkan / Data yang dimasukkan tidak sesuai dengan format yang dibutuhkan! Silahkan masukkan data lagi");  
184  
185
186
187     als(".")  
188     ().equals(".")  
189     ().equals("K"))  
190     t().equals(".")  
191     ().equals(".")  
192     ().equals("K")) {  
193  
194         da Masukkan Kurang Tepat!");  
195
196
197
198
199
200     arga = " + txtHarga.getText() + ", stok = " + txtStock.getText() + ", kategori = '" + cmbKategori.getSelectedItem() + "'";
```

Gambar 2.188 : Fungsi Update() di halaman entry barang (2)

The screenshot shows a Java IDE interface with multiple tabs at the top: Start Page, Menu_login.java, Menu_UserAdmin.java, Login_User.java, Form_Transaksi.java, AdminToko.java, Form_Kategori.java, and Entry_Barang.java. The main area displays Java code for the Entry_Barang.java class. The code includes a try-catch block. The catch block contains a JOptionPane.showMessageDialog call with a message about an error. A red vertical line highlights the entire try-catch block.

```
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221     try {
        sql = "UPDATE tb_barang"
        + " SET nama = '" + txtNama.getText() + "', harga = " + txtHarga.getText() + ", stok = " + txtStock.getText()
        + " WHERE kode = " + txtKode.getText();

        koneksi = jaringan.Jaringan.config();
        pernyataan = koneksi.createStatement();
        pernyataan.executeUpdate(sql);

        JOptionPane.showMessageDialog(null, "Update Berhasil");

        Kosong();
        isi_table();
        tidak_bisa_isi();
        tombol_hidup();

        btnSave.setEnabled(false);
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, "Keterangan Error :" + e);
    }
}
```

Gambar 2.189 : Fungsi Update() di halaman entry barang (3)

13. Buat fungsi hapus(), isi fungsi tersebut.

```
223     private void hapus() {
224
225         // Jika tidak ada data yang dimasukkan
226         if (txtKode.getText().equals("") || txtNama.getText().equals("") ||
227             || txtHarga.getText().equals("") || txtStock.getText().equals("") ||
228             cmbKategori.getSelectedItem().equals("")) {
229
230             JOptionPane.showMessageDialog(null, "Tidak ada data yang dipilih! Silahkan memilih data lagi");
231
232         } else {
233
234             int dialogButton = JOptionPane.YES_NO_OPTION;
235             int dialogResult = JOptionPane.showConfirmDialog(this, "Data Akan di Hapus", "Hapus Data", dialogButton);
236
237             if (dialogResult != 0) {
238
239                 JOptionPane.showMessageDialog(null, "Batal di Hapus");
240             } else {
241
242                 try {
243
244                     sql = "DELETE"
245                         + " FROM tb_barang"
246                         + " WHERE kode = " + txtKode.getText();
247
248                     koneksi = jaringan.Jaringan.config();
249
250                     Statement statement = koneksi.createStatement();
251
252                     statement.executeUpdate(sql);
253
254                     JOptionPane.showMessageDialog(null, "Data Berhasil di Hapus");
255
256                     txtKode.setText("");
257                     txtNama.setText("");
258                     txtHarga.setText("");
259                     txtStock.setText("");
260                     cmbKategori.setSelectedIndex(0);
261
262                     tabelBarang.setModel(model);
263
264                     tabelBarang.setRowSelectionInterval(0, -1);
265
266                     tabelBarang.clearSelection();
267
268                     tabelBarang.repaint();
269
270                     tabelBarang.revalidate();
271
272                     tabelBarang.updateUI();
273
274                     tabelBarang.repaint();
275
276                     tabelBarang.revalidate();
277
278                     tabelBarang.updateUI();
279
280                     tabelBarang.repaint();
281
282                     tabelBarang.revalidate();
283
284                     tabelBarang.updateUI();
285
286                     tabelBarang.repaint();
287
288                     tabelBarang.revalidate();
289
290                     tabelBarang.updateUI();
291
292                     tabelBarang.repaint();
293
294                     tabelBarang.revalidate();
295
296                     tabelBarang.updateUI();
297
298                     tabelBarang.repaint();
299
299                 }
300             }
301         }
302     }
303 }
```

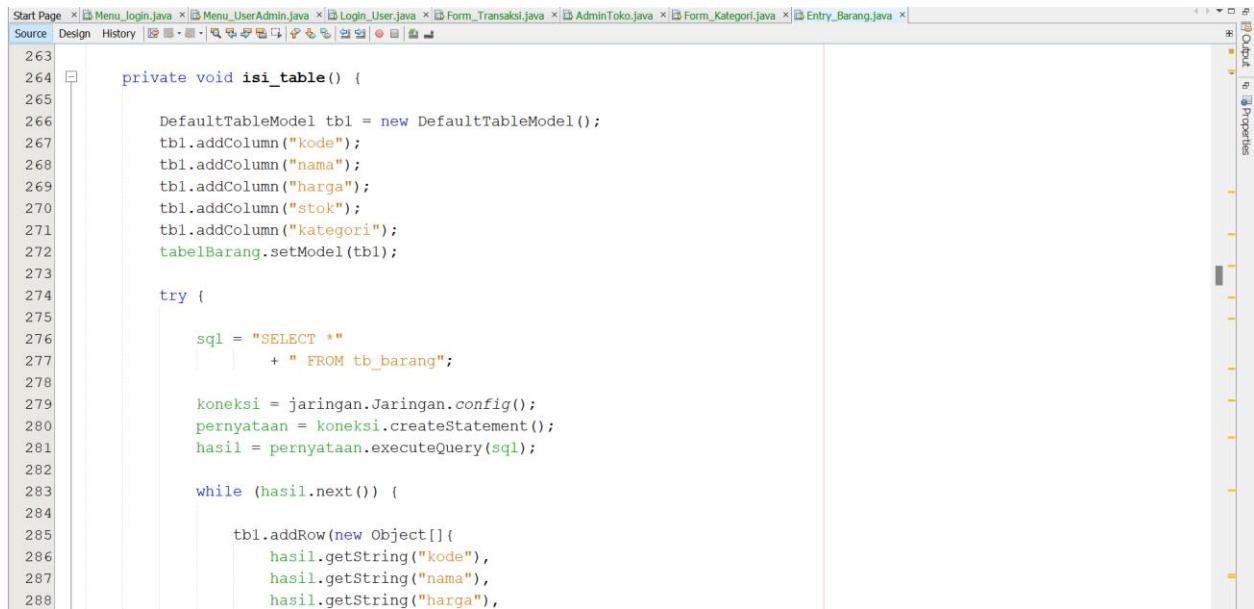
Gambar 2.190 : Fungsi hapus() di halaman entry barang(1)

```
Start Page x Menu_login.java x Menu_UserAdmin.java x Login_User.java x Form_Transaksi.java x AdminToko.java x Form_Kategori.java x Entry_Barang.java x
Source Design History 
```

238 JOptionPane.showMessageDialog(null, "Batal di Hapus");
239 } else {
240
241 try {
242
243 sql = "DELETE"
244 + " FROM tb_barang"
245 + " WHERE kode = " + txtKode.getText();
246
247 koneksi = jaringan.Jaringan.config();
248 pernyataan = koneksi.createStatement();
249 pernyataan.execute(sql);
250
251 JOptionPane.showMessageDialog(null, "Berhasil di Hapus");
252
253 Kosong();
254 isi_table();
255 } catch (Exception e) {
256
257 JOptionPane.showMessageDialog(null, "Keterangan Error :" + e);
258 }
259 }
260 }
261 }
262 }
263 }

Gambar 2.191 : Fungsi hapus() di halaman entry barang(2)

14. Buat fungsi isi_table(), isi fungsi tersebut.



```
Start Page | Menu_Login.java | Menu_UserAdmin.java | Login_User.java | Form_Transaksi.java | AdminToko.java | Form_Kategori.java | Entry_Barang.java | 
Source Design History | 
263 | 
264 |     private void isi_table() { 
265 | 
266 |         DefaultTableModel tbl = new DefaultTableModel(); 
267 |         tbl.addColumn("kode"); 
268 |         tbl.addColumn("nama"); 
269 |         tbl.addColumn("harga"); 
270 |         tbl.addColumn("stok"); 
271 |         tbl.addColumn("kategori"); 
272 |         tabelBarang.setModel(tbl); 
273 | 
274 |         try { 
275 | 
276 |             sql = "SELECT * 
277 |                 + " FROM tb_barang"; 
278 | 
279 |             koneksi = jaringan.Jaringan.config(); 
280 |             pernyataan = koneksi.createStatement(); 
281 |             hasil = pernyataan.executeQuery(sql); 
282 | 
283 |             while (hasil.next()) { 
284 | 
285 |                 tbl.addRow(new Object[] { 
286 |                     hasil.getString("kode"), 
287 |                     hasil.getString("nama"), 
288 |                     hasil.getString("harga"), 
289 |                     hasil.getString("stok"), 
290 |                     hasil.getString("kategori")}); 
291 |             } 
292 | 
293 |             tabelBarang.setModel(tbl); 
294 |         } catch (Exception e) { 
295 |         } 
296 |     } 
297 | 
298 | } 
299 | 
```

Gambar 2.192 : Fungsi isi_table() di halaman entry barang(1)



```
Start Page | Menu_Login.java | Menu_UserAdmin.java | Login_User.java | Form_Transaksi.java | AdminToko.java | Form_Kategori.java | Entry_Barang.java | 
Source Design History | 
274 | 
275 |     try { 
276 | 
277 |         sql = "SELECT * 
278 |                 + " FROM tb_barang"; 
279 | 
280 |         koneksi = jaringan.Jaringan.config(); 
281 |         pernyataan = koneksi.createStatement(); 
282 |         hasil = pernyataan.executeQuery(sql); 
283 | 
284 |         while (hasil.next()) { 
285 | 
286 |             tbl.addRow(new Object[] { 
287 |                 hasil.getString("kode"), 
288 |                 hasil.getString("nama"), 
289 |                 hasil.getString("harga"), 
290 |                 hasil.getString("stok"), 
291 |                 hasil.getString("kategori")}); 
292 |         } 
293 | 
294 |         tabelBarang.setModel(tbl); 
295 |     } catch (Exception e) { 
296 |     } 
297 | 
298 | } 
299 | 
```

Gambar 2.193 : Fungsi isi_table() di halaman entry barang(2)

15. Buat fungsi dengan nama isiItemKategori(), isi fungsi tersebut.

The screenshot shows a Java code editor with the 'Source' tab selected. The code is part of a class and defines a method named 'isiItemKategori'. The method uses a try-catch block to execute a SQL query against a database connection. It adds items from the result set to a JComboBox model. If an exception occurs, it prints an error message. Finally, it calls a 'Kosong()' method. The code is numbered from 299 to 324.

```
299  
300     private void isiItemKategori() {  
301         DefaultComboBoxModel cmb1 = new DefaultComboBoxModel();  
302  
303         try {  
304             sql = "SELECT *  
305                 + " FROM tb_kategori";  
306  
307             koneksi = jaringan.Jaringan.config();  
308             pernyataan = koneksi.createStatement();  
309             hasil = pernyataan.executeQuery(sql);  
310  
311             while (hasil.next()) {  
312                 cmb1.addElement(hasil.getString("nama_kategori"));  
313             }  
314             cmbKategori.setModel(cmb1);  
315         } catch (Exception e) {  
316             e.printStackTrace();  
317         }  
318         Kosong();  
319     }  
320  
321 }
```

Gambar 2.194 : Fungsi isiItemKategori() pada halaman entry barang

16. Buat fungsi tambahStock(), isi fungsi tersebut.

The screenshot shows a Java code editor with the 'Source' tab selected. The code is part of a class and defines a method named 'tambahStock'. It first checks if no data is selected in four input fields (txtKode, txtNama, txtHarga, txtStock). If none are selected, it shows a message dialog. Otherwise, it initializes variables for conversion and input. It then shows an input dialog for updating stock. The stock value is parsed as an integer. If the input is not a valid number, it prints an error message. The code is numbered from 324 to 349.

```
324  
325     private void tambahStock() {  
326  
327         // Jika tidak ada data yang dipilih  
328         if (txtKode.getText().equals("") || txtNama.getText().equals("") ||  
329             txtHarga.getText().equals("") || txtStock.getText().equals("") ||  
330             cmbKategori.getSelectedItem().equals("")) {  
331             JOptionPane.showMessageDialog(null, "Tidak ada data yang dipilih! Silahkan memilih data lagi");  
332         } else {  
333             int ConvJumlah;  
334             int jumlah;  
335             int ConvStok;  
336             String ConvHasil;  
337  
338             String stok = JOptionPane.showInputDialog(  
339                     "Update Stock: "  
340             );  
341  
342             ConvStok = Integer.parseInt(stok);  
343  
344             if (!stok.equals("")) {  
345                 if (stok.equals("-") || stok.equals(".") || stok.equals(",") ||  
346                     stok.equals(" ")) {  
347                         JOptionPane.showMessageDialog(null, "Masukkan angka saja");  
348                     }  
349             }  
350         }  
351     }  
352 }
```

Gambar 2.195 : Fungsi tambahStock() pada halaman entry barang(1)

The screenshot shows a Java code editor with the following code snippet:

```
349     if (stok.equals("-") || stok.equals(".")) || stok.equals(",") ||  
350         stok.equals(" ") || stok.equals("K") || stok.equals("K")) {  
351         JOptionPane.showMessageDialog(null, "Format Angka yang Anda Masukkan Kurang Tepat!");  
352     } else {  
353         try {  
354             sql = "SELECT *"  
355                 + " FROM tb_barang"  
356                 + " WHERE kode = " + txtKode.getText();  
357  
358             koneksi = jaringan.Jaringan.config();  
359             pernyataan = koneksi.createStatement();  
360             hasil = pernyataan.executeQuery(sql);  
361  
362             while (hasil.next()) {  
363                 ConvJumlah = Integer.parseInt(hasil.getString("stok"));  
364                 jumlah = ConvStok + ConvJumlah;  
365  
366                 ConvHasil = String.valueOf(jumlah);  
367  
368                 sq12 = "UPDATE tb_barang"  
369                     + " SET stok = " + ConvHasil  
370                     + " WHERE kode = " + txtKode.getText();  
371  
372             }  
373         }  
374     }  
375     jumlah = ConvStok + ConvJumlah;  
376  
377     ConvHasil = String.valueOf(jumlah);  
378  
379     sq12 = "UPDATE tb_barang"  
380         + " SET stok = " + ConvHasil  
381         + " WHERE kode = " + txtKode.getText();  
382  
383     koneksi = jaringan.Jaringan.config();  
384     pernyataan = koneksi.createStatement();  
385     pernyataan.executeUpdate(sq12);  
386  
387     JOptionPane.showMessageDialog(null, "Tambah Stock Berhasil");  
388 } catch (Exception e) {  
389     JOptionPane.showMessageDialog(null, "Keterangan Error :" + e);  
390 }  
391  
392     isi_table();  
393     Kosong();  
394 }
```

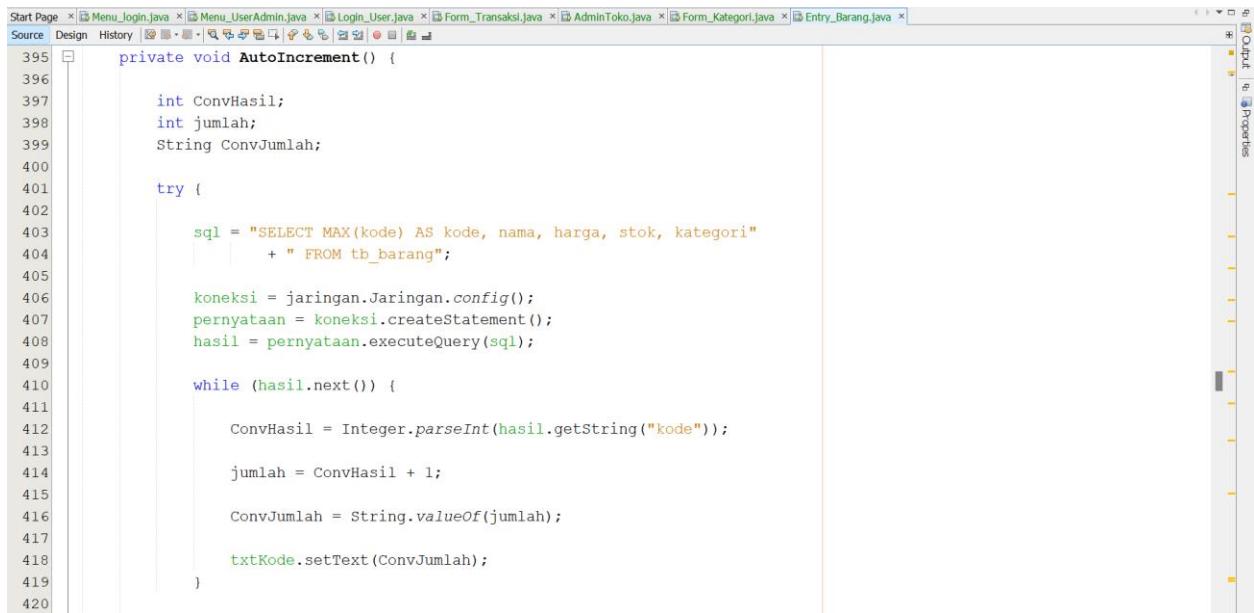
Gambar 2.196 : Fungsi tambahStock() pada halaman entry barang(2)

The screenshot shows the continuation of the Java code editor with the following code snippet:

```
368     jumlah = ConvStok + ConvJumlah;  
369  
370     ConvHasil = String.valueOf(jumlah);  
371  
372     sq12 = "UPDATE tb_barang"  
373         + " SET stok = " + ConvHasil  
374         + " WHERE kode = " + txtKode.getText();  
375  
376     koneksi = jaringan.Jaringan.config();  
377     pernyataan = koneksi.createStatement();  
378     pernyataan.executeUpdate(sq12);  
379  
380 }  
381  
382 JOptionPane.showMessageDialog(null, "Tambah Stock Berhasil");  
383 } catch (Exception e) {  
384     JOptionPane.showMessageDialog(null, "Keterangan Error :" + e);  
385 }  
386  
387 isi_table();  
388 Kosong();  
389 }  
390  
391 }  
392 }  
393 }
```

Gambar 2.197 : Fungsi tambahStock() pada halaman entry barang(3)

17. Buat fungsi AutoIncrement(), isi fungsi tersebut.



The screenshot shows a Java code editor with the following code:

```
private void AutoIncrement() {
    int ConvHasil;
    int jumlah;
    String ConvJumlah;

    try {
        sql = "SELECT MAX(kode) AS kode, nama, harga, stok, kategori"
              + " FROM tb_barang";

        koneksi = jaringan.Jaringan.config();
        pernyataan = koneksi.createStatement();
        hasil = pernyataan.executeQuery(sql);

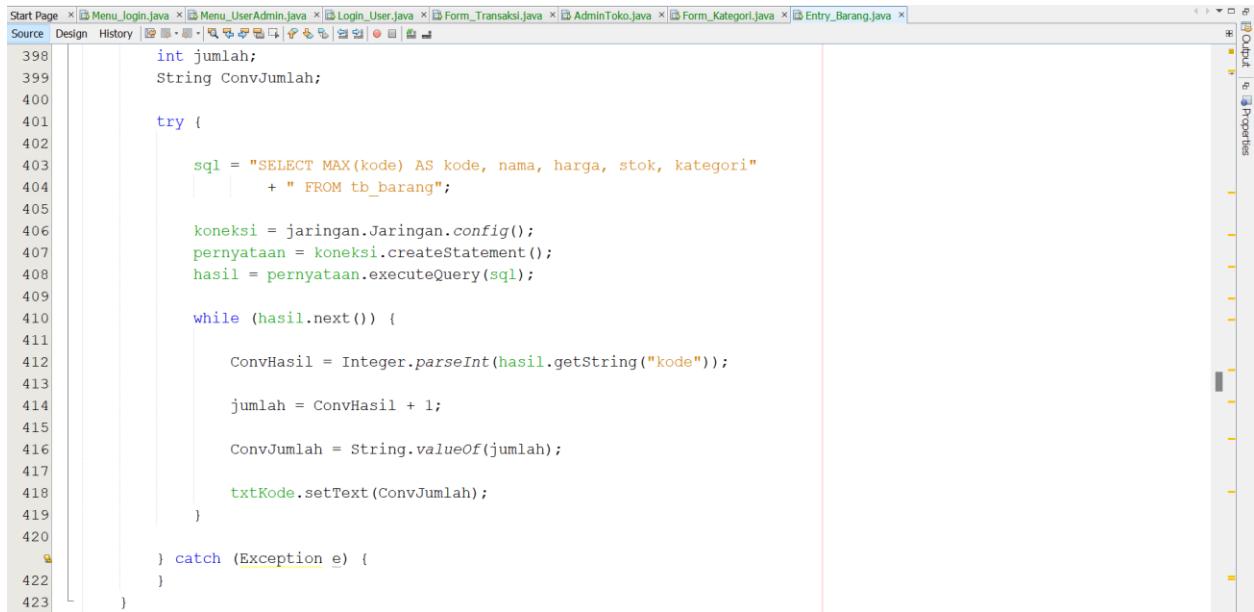
        while (hasil.next()) {
            ConvHasil = Integer.parseInt(hasil.getString("kode"));

            jumlah = ConvHasil + 1;

            ConvJumlah = String.valueOf(jumlah);

            txtKode.setText(ConvJumlah);
        }
    }
}
```

Gambar 2.198 : Fungsi AutoIncrement() pada halaman entry barang(1)



The screenshot shows a Java code editor with the following code:

```
int jumlah;
String ConvJumlah;

try {
    sql = "SELECT MAX(kode) AS kode, nama, harga, stok, kategori"
          + " FROM tb_barang";

    koneksi = jaringan.Jaringan.config();
    pernyataan = koneksi.createStatement();
    hasil = pernyataan.executeQuery(sql);

    while (hasil.next()) {
        ConvHasil = Integer.parseInt(hasil.getString("kode"));

        jumlah = ConvHasil + 1;

        ConvJumlah = String.valueOf(jumlah);

        txtKode.setText(ConvJumlah);
    }
} catch (Exception e) {
}
}
```

Gambar 2.199 : Fungsi AutoIncrement() pada halaman entry barang(2)

18. Buat fungsi Search(), isi fungsi tersebut.

```
Start Page | Menu_Login.java | Menu_UserAdmin.java | Login_User.java | Form_Transaksi.java | AdminToko.java | Form_Kategori.java | Entry_Barang.java
Source Design History | Back Forward Stop Refresh Properties
425 private void Search() {
426
427     // Jika TextField search kosong
428     if (txtSearch.getText().equals("")) {
429
430         txtSearch.requestFocus();
431
432     } else {
433
434         DefaultTableModel tbl = new DefaultTableModel();
435         tbl.addColumn("kode");
436         tbl.addColumn("nama");
437         tbl.addColumn("harga");
438         tbl.addColumn("stok");
439         tbl.addColumn("kategori");
440         tabelBarang.setModel(tbl);
441
442         try {
443
444             sql = "SELECT *"
445             + " FROM tb_barang"
446             + " WHERE kode = '" + txtSearch.getText() + "' || nama LIKE '%" + txtSearch.getText() + "%' || kat
447
448             koneksi = jaringan.Jaringan.config();
449             pernyataan = koneksi.createStatement();
450             hasil = pernyataan.executeQuery(sql);
451
452         } catch (Exception e) {
453             JOptionPane.showMessageDialog(null, "Error : " + e.getMessage());
454         }
455
456         tabelBarang.setModel(tbl);
457
458     }
459
460 }
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500 }
```

Gambar 2.200 : Fungsi Search() pada halaman entry barang(1)

```
Start Page | Menu_Login.java | Menu_UserAdmin.java | Login_User.java | Form_Transaksi.java | AdminToko.java | Form_Kategori.java | Entry_Barang.java
Source Design History | Back Forward Stop Refresh Properties
425
426
427
428
429
430
431
432
433
434     litTableModel();
435
436
437
438
439
440
441
442
443
444
445
446     txtSearch.getText() + "' || nama LIKE '%" + txtSearch.getText() + "%' || kategori LIKE '%" + txtSearch.getText() + "%'";
447
448     onfig();
449     atement();
450     ry(sq1);
```

Gambar 2.201 : Fungsi Search() pada halaman entry barang(2)

```

443     sql = "SELECT *"
444         + " FROM tb_barang"
445         + " WHERE kode = '" + txtSearch.getText() + "' || nama LIKE '%" + txtSearch.getText() + "%' || kat
446
447     koneksi = jaringan.Jaringan.config();
448     pernyataan = koneksi.createStatement();
449     hasil = pernyataan.executeQuery(sql);
450
451     while (hasil.next()) {
452
453         tabelBarang.addRow(new Object[]{
454             hasil.getString("kode"),
455             hasil.getString("nama"),
456             hasil.getString("harga"),
457             hasil.getString("stok"),
458             hasil.getString("kategori")});
459
460     }
461
462     tabelBarang.setModel(tbl);
463
464     txtKode.setEnabled(true);
465 } catch (Exception e) {
466 }
467
468 }

```

Gambar 2.202 : Fungsi Search() pada halaman entry barang(3)

19. Buat fungsi functionEdit(), isi fungsi tersebut.

```

470     private void functionEdit() {
471
472         if (txtKode.getText().equals("") || txtNama.getText().equals("") ||
473             txtHarga.getText().equals("") || txtStock.getText().equals("") ||
474             cmbKategori.getSelectedItem().equals("")) {
475
476             JOptionPane.showMessageDialog(null, "Tidak ada data yang dipilih! Silahkan memilih data lagi");
477         } else {
478
479             bisa_isi();
480             tombol_mati();
481
482             txtNama.requestFocus();
483             btnSave.setText("Update");
484             btnSave.setEnabled(true);
485             txtKode.setEnabled(false);
486
487             btnBaru.setEnabled(false);
488             btnEdit.setEnabled(false);
489             btnHapus.setEnabled(false);
490             btnBack.setEnabled(false);
491             btnTambahStock.setEnabled(false);
492
493         }
494     }

```

Gambar 2.203 : Fungsi functionEdit() pada halaman entry barang

20. Buat fungsi event. Fungsi event ini digunakan agar element pada halaman jframe dapat berjalan sesuai dengan event yang anda tentukan. Untuk membuat event, anda bisa ke tab design => tekan kanan mouse pada elemen yang ingin kasih event => pilih menu event =>

pilih event yang anda inginkan. Event yang tersedia untuk elemen itu bermacam – macam, sesuaikan dengan kebutuhan anda. Berikut beberapa contoh event yang digunakan pada halaman form CRUD entry barang:

Gambar 2.204 : Fungsi Event dan isinya pada halaman entry barang (1)

```
865
866  □ private void btnSaveActionPerformed(java.awt.event.ActionEvent evt) {
867      // TODO add your handling code here:
868
869      if (btnSave.getText() == "Simpan") {
870
871          Simpan();
872      } else {
873
874          update();
875      }
876  }
877
878  □ private void btnEditActionPerformed(java.awt.event.ActionEvent evt) {
879      // TODO add your handling code here:
880
881      functionEdit();
882  }
883
884  □ private void btnHapusActionPerformed(java.awt.event.ActionEvent evt) {
885      // TODO add your handling code here:
886
887      hapus();
888  }
889
```

Gambar 2.205 : Fungsi Event dan isinya pada halaman entry barang (2)

The screenshot shows a Java IDE interface with multiple tabs at the top: Start Page, Menu_login.java, Menu_UserAdmin.java, Login_User.java, Form_Transaksi.java, AdminToko.java, Form_Kategori.java, and Entry_Barang.java. The main code editor displays Java code for the Entry_Barang.java class. The code includes three event handlers: btnResetActionPerformed, btnTambahStockActionPerformed, and txtHargaActionPerformed. The code is mostly placeholder comments, with some specific logic like JOptionPane.showMessageDialog and Kosong() methods.

```
889  
890  private void btnResetActionPerformed(java.awt.event.ActionEvent evt) {  
891      // TODO add your handling code here:  
892  
893      JOptionPane.showMessageDialog(null, "Reset");  
894  
895      Kosong();  
896      tombol_hidup();  
897  }  
898  
899  private void btnTambahStockActionPerformed(java.awt.event.ActionEvent evt) {  
900      // TODO add your handling code here:  
901  
902      tidak_bisa_isi();  
903  
904      tambahStock();  
905  }  
906  
907  private void txtHargaActionPerformed(java.awt.event.ActionEvent evt) {  
908      // TODO add your handling code here:  
909  }
```

Gambar 2.206 : Fungsi Event dan isinya pada halaman entry barang (3)

The screenshot shows a Java IDE interface with multiple tabs at the top: Start Page, Menu_login.java, Menu_UserAdmin.java, Login_User.java, Form_Transaksi.java, AdminToko.java, Form_Kategori.java, and Entry_Barang.java. The main code editor displays Java code for the Entry_Barang.java class. The code includes two event handlers: tabelBarangMouseClicked and txtKodeActionPerformed. The tabelBarangMouseClicked handler retrieves data from a selected table row and sets it into various text fields and a combobox. The txtKodeActionPerformed handler is a placeholder comment.

```
910  
911  private void tabelBarangMouseClicked(java.awt.event.MouseEvent evt) {  
912      // TODO add your handling code here:  
913  
914      // untuk mendisable editing column pada tabel  
915      tabelBarang.setDefaultEditor(Object.class, null);  
916  
917      int tabelData = tabelBarang.getSelectedRow();  
918  
919      txtKode.setText(tabelBarang.getValueAt(tabelData, 0).toString());  
920      txtNama.setText(tabelBarang.getValueAt(tabelData, 1).toString());  
921      txtHarga.setText(tabelBarang.getValueAt(tabelData, 2).toString());  
922      txtStock.setText(tabelBarang.getValueAt(tabelData, 3).toString());  
923      cmbKategori.setSelectedItem(tabelBarang.getValueAt(tabelData, 4).toString());  
924  
925      tabelBarang.isCellEditable(tabelData, 0);  
926      tabelBarang.isCellEditable(tabelData, 1);  
927      tabelBarang.isCellEditable(tabelData, 2);  
928      tabelBarang.isCellEditable(tabelData, 3);  
929      tabelBarang.isCellEditable(tabelData, 4);  
930  }  
931  
932  private void txtKodeActionPerformed(java.awt.event.ActionEvent evt) {  
933      // TODO add your handling code here:  
934  }
```

Gambar 2.207 : Fungsi Event dan isinya pada halaman entry barang (4)

```
Start Page × Menu_login.java × Menu_UserAdmin.java × Login_User.java × Form_Transaksi.java × AdminToko.java × Form_Kategori.java × Entry_Barang.java ×
Source Design History 
931
932     private void txtKodeActionPerformed(java.awt.event.ActionEvent evt) {
933         // TODO add your handling code here:
934     }
935
936     private void btnSearchActionPerformed(java.awt.event.ActionEvent evt) {
937         // TODO add your handling code here:
938
939         Search();
940         bisa_isi();
941     }
942
943     private void cmbKategoriActionPerformed(java.awt.event.ActionEvent evt) {
944         // TODO add your handling code here:
945     }
946
947     private void btnRefreshActionPerformed(java.awt.event.ActionEvent evt) {
948         // TODO add your handling code here:
949
950         dispose();
951
952         new Entry_Barang().setVisible(true);
953     }

```

Gambar 2.208 : Fungsi Event dan isinya pada halaman entry barang (5)

```
Start Page x Menu_Login.java x Menu_UserAdmin.java x Login_User.java x Form_Transaksi.java x AdminToko.java x Form_Kategori.java x Entry_Barang.java x
Source Design History
940     bisa_isi();
941 }
942
943 private void cmbKategoriActionPerformed(java.awt.event.ActionEvent evt) {
944     // TODO add your handling code here:
945 }
946
947 private void btnRefreshActionPerformed(java.awt.event.ActionEvent evt) {
948     // TODO add your handling code here:
949
950     dispose();
951
952     new Entry_Barang().setVisible(true);
953 }
954
955 private void btnBackActionPerformed(java.awt.event.ActionEvent evt) {
956     // TODO add your handling code here:
957
958     dispose();
959
960     new AdminToko().setVisible(true);
961 }
962
963 private void txtSearchActionPerformed(java.awt.event.ActionEvent evt) {
964     // TODO add your handling code here:
965 }
```

Gambar 2.209 : Fungsi Event dan isinya pada halaman entry barang (6)

O. Pembuatan Halaman Laporan Transaksi

Halaman Laporan Transaksi ini adalah halaman yang digunakan untuk membuat laporan tentang transaksi. Halaman laporan admin ini memiliki 2 kolom (tanggal_awal dan tanggal_akhir) dan 2 menu. Pada halaman laporan ini bisa mencetak laporan transaksi berdasarkan tanggal yang sudah ditentukan. Untuk membuat file jframe, anda bisa melihat

pada ‘**Pembuatan Halaman Utama User Kasir**’. Untuk desain yang ada di halaman entry barang ini, anda harus membuat desain seperti gambar 2.210 ini.

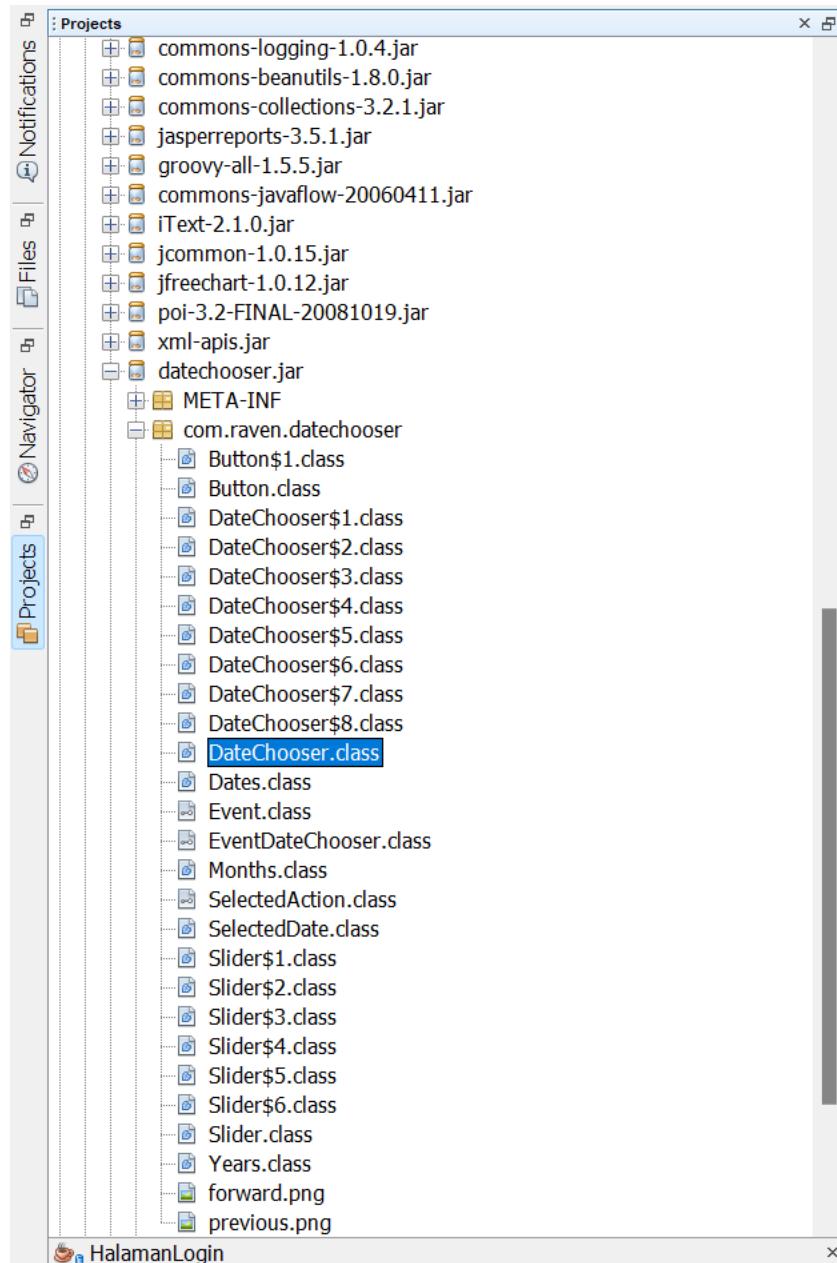
The screenshot shows a user interface for a transaction report. At the top, a teal header bar contains the title 'Laporan Transaksi'. Below this, a yellow section features a 'Range Tanggal' input field with two date pickers separated by a '-' sign, and buttons for 'Cari' (Search) and 'Print'. A caption 'Tabel Data Transaksi' is positioned above a table with four columns: 'id_transaksi', 'total_harga', 'tanggal', and 'status_pembayaran'. The table has four rows, all of which are currently empty. At the bottom of the yellow section is a large, empty light-gray rectangular area. The footer is a teal bar containing three buttons: 'Refresh', 'Back', and 'Keluar'.

Gambar 2.210 : Halaman laporan transaksi

Jika desain sudah dibuat, jangan lupa untuk mengganti nama variable tiap elemen, agar mempermudah untuk diingat. Setelah mengganti variabel, anda perlu mengaitkan textfield tanggal awal dan tanggal akhir dengan class ‘Datechooser’, agar textfield ini bisa berfungsi sebagai tempat pemilihan tanggal. Untuk mengaitkan texfield dengan class ‘Datechooser’, ada beberapa langkah yang harus dilakukan:

1. Pastikan library ‘Datechooser’ sudah terimport di project.

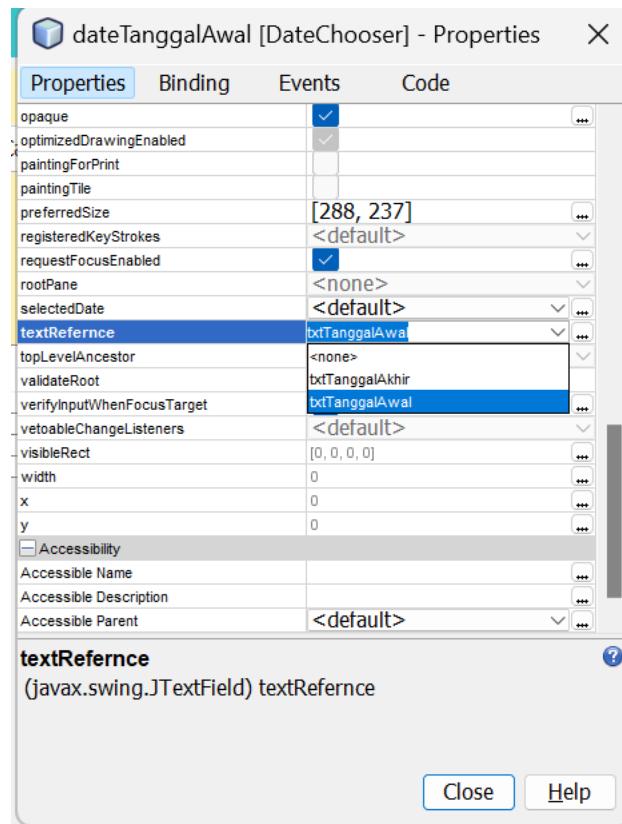
2. Buka library ‘datechooser’ pada libraries project => buka package ‘com.raven.datechooser’ => Drag and drop class ‘Datechooser’ ke bagian luar canvas design halaman laporan. Lakukan drag and drop class ‘Datechooser’ 2 kali.



Gambar 2.211 : Class Datechooser pada library datechooser

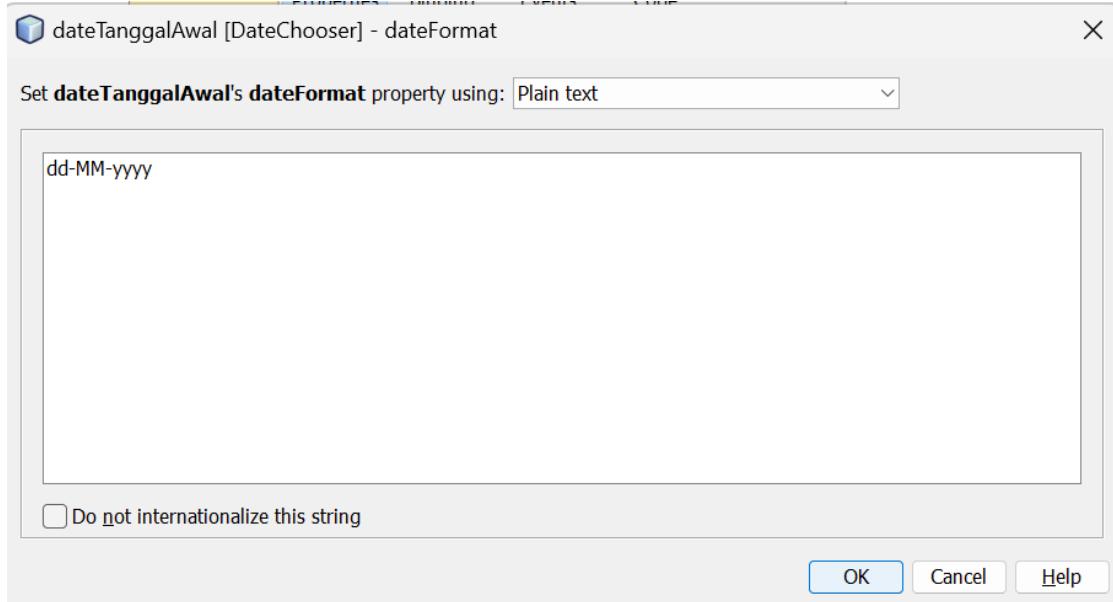
3. Buka tab navigator, akan ada 2 komponen di bagian other komponens. Komponen tersebut memiliki nama ‘datechooser 1’ dan ‘datechooser 2’. Nah, anda harus mengganti nama variabel tiap komponen menjadi ‘dateTanggalAwal’ sama ‘dateTanggalAkhir’.

4. Jika nama variabel sudah diubah, maka tiap komponen tadi kita hubungkan dengan textfield pada design halaman laporan sesuai peruntukannya. Untuk mengganti textReference: tekan kanan mouse pada komponen datechooser => pilih properties => cari properti textReference => ganti text reference menjadi sesuai peruntukannya masing – masing.



Gambar 2.212 : Properti textReference tiap komponen datechooser

5. Jangan lupa untuk mengganti dateformat tiap komponennya. Untuk mengganti dateFormat: tekan kanan mouse pada komponen datechooser => pilih properties => cari properti dateFormat => klik tombol titik tiga di pojok kanan => isi format seperti ini ‘dd-MM-yyyy’.



Gambar 2.213 : DateFormat untuk datechooser

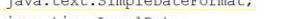
Jika sudah menhubungkan datechooser dengan textfield, jangan lupa tiap textfield diberi tooltipstext. Setelah itu buka tab Soure pada halaman ini, lalu buat sistemnya agar halaman utama user ini dapat berjalan dengan baik. Berikut cara membuat sistem untuk halaman laporan transaksi:

1. Import library yang dibutuhkan untuk sistem laporan

```
Start Page | Menu_Login.java | Menu_UserAdmin.java | Login_User.java | Form_Transaksi.java | AdminToko.java | Form_Kategori.java | Entry_Barang.java | Laporan_Admin.java |  
Source Design History |           
```

```
1  /*  
2   * To change this license header, choose License Headers in Project Properties.  
3   * To change this template file, choose Tools | Templates  
4   * and open the template in the editor.  
5   */  
6  package Menubar_login;  
7  
8  import com.raven.datechooser.EventDateChooser;  
9  import com.raven.datechooser.SelectedAction;  
10 import com.raven.datechooser.SelectedDate;  
11 import java.io.File;  
12 import java.io.InputStream;  
13 import java.sql.Connection;  
14 import java.sql.ResultSet;  
15 import java.sql.Statement;  
16 import java.text.SimpleDateFormat;  
17 import java.time.LocalDate;  
18 import java.time.Period;  
19 import java.time.format.DateTimeFormatter;  
20 import java.time.temporal.ChronoUnit;  
21 import java.util.HashMap;  
22 import java.util.Locale;  
23 import java.util.Map;  
24 import javax.swing.JOptionPane;  
25 import javax.swing.table.DefaultTableModel;  
import javax.swing.table.TableModel;
```

Gambar 2.214 : Library yang dibutuhkan untuk sistem laporan transaksi (1)

```
Start Page × Menu_login.java × Menu_UserAdmin.java × Login_User.java × Form_Transaksi.java × AdminToko.java × Form_Kategori.java × Entry_Barang.java × Laporan_Admin.java ×
Source Design History 
14 import java.sql.ResultSet;
15 import java.sql.Statement;
16 import java.text.SimpleDateFormat;
17 import java.time.LocalDate;
18 import java.time.Period;
19 import java.time.format.DateTimeFormatter;
20 import java.time.temporal.ChronoUnit;
21 import java.util.HashMap;
22 import java.util.Locale;
23 import java.util.Map;
24 import javax.swing.JOptionPane;
25 import javax.swing.table.DefaultTableModel;
26 import javax.swing.table.TableModel;
27 import net.sf.jasperreports.engine.JasperCompileManager;
28 import net.sf.jasperreports.engine.JasperFillManager;
29 import net.sf.jasperreports.engine.JasperPrint;
30 import net.sf.jasperreports.engine.JasperReport;
31 import net.sf.jasperreports.engine.design.JasperDesign;
32 import net.sf.jasperreports.engine.util.JRLoader;
33 import net.sf.jasperreports.engine.xml.JRXmlLoader;
34 import net.sf.jasperreports.view.JasperViewer;
35
36 /**
37 *
38 * @author adham
39 */
```

Gambar 2.215 : Library yang dibutuhkan untuk sistem laporan transaksi (2)

2. Buat beberapa variable scope global

The screenshot shows a Java IDE interface with the following details:

- Title Bar:** The title bar displays the project name "Start Page" and several open files: "Menu_Login.java", "Menu_UserAdmin.java", "Login_User.java", "Form_Transaksi.java", "AdminToko.java", "Form_Kategori.java", "Entry_Barang.java", and "Laporan_Admin.java".
- Toolbar:** The toolbar includes icons for Source, Design, History, and various search and navigation functions.
- Code Editor:** The main area shows the code for the "Laporan_Admin" class. The code uses JDBC to connect to a database and generate a JasperReport. It includes imports for javax.swing.JFrame, java.sql.Connection, java.sql.Statement, java.sql.ResultSet, and java.util.HashMap.
- Sidebar:** On the left, there's a "Projects" view showing the project structure. On the right, there's a "Notifications" panel.

```
40 public class Laporan_Admin extends javax.swing.JFrame {  
41  
42     private Connection koneksi;  
43     private Statement pernyataan;  
44     private ResultSet hasil;  
45     private String sql;  
46     private String sql2;  
47  
48     //ireport  
49     JasperReport JasRep;  
50     JasperPrint JasPri;  
51     Map param = new HashMap();  
52     JasperDesign JasDes;  
53     InputStream jasperStream;  
54 }
```

Gambar 2.216 : Variable scope global pada sistem halaman laporan transaksi

3. Isi constructor, sesuai dengan gambar 2.217

```
Start Page × Menu_Login.java × Menu_UserAdmin.java × Login_User.java × Form_Transaksi.java × AdminToko.java × Form_Kategori.java × Entry_Barang.java × Laporan_Admin.java ×
Source Design History 

```

55 /**
56 * Creates new form Laporan Admin
57 */
58 public Laporan_Admin() {
59 initComponents();
60
61 Kosong();
62 sistemTabelTransaksi();
63 isi_tableTransaksiSemua();
64 }

```


```

Gambar 2.217 : Constructor pada halaman laporan transaksi

4. Buat fungsi dengan nama Kosong(), isi fungsi tersebut.

```
65
66     public void Kosong() {
67         txtTanggalAwal.setText("");
68         txtTanggalAkhir.setText("");
69     }

```

Gambar 2.218 : Fungsi Kosong() di halaman laporan transaksi

5. Buat fungsi sistemTabelTransaksi(), isi fungsi tersebut.

```
70  
71     private void sistemTabelTransaksi() {  
72         // untuk mendisable editing column pada tabel  
73         tabelTransaksi.setDefaultEditor(Object.class, null);  
74     }  
75  
76
```

Gambar 2.219 : Fungsi sistemTabelTransaksi() pada halaman laporan transaksi

6. Buat fungsi isi_tableTransaksi(), isi fungsi tersebut.

Gambar 2.220 : Fungsi isi tableTransaksi() pada halaman laporan transaksi (1)

The screenshot shows a Java code editor with several tabs at the top: Start Page, Menu_login.java, Menu_UserAdmin.java, Login_User.java, Form_Transaksi.java, AdminToko.java, Form_Kategori.java, Entry_Barang.java, and Laporan_Admin.java. The main code area contains the following Java code:

```
101 // Pengecekan Jarak hari
102 long jarakHari = ChronoUnit.DAYS.between(tanggalAwalUbahDate, tanggalAkhirUbahDate);
103
104 if (jarakHari < 0) {
105     JOptionPane.showMessageDialog(null,
106         "Tanggal yang dipilih tidak tepat!");
107 } else {
108
109     try {
110
111         DefaultTableModel tbl = new DefaultTableModel();
112         tbl.addColumn("id_transaksi");
113         tbl.addColumn("total_harga");
114         tbl.addColumn("tanggal_transaksi");
115         tbl.addColumn("status_pembayaran");
116         tabelTransaksi.setModel(tbl);
117
118         sql = "SELECT *"
119             + " FROM tb_transaksi"
120             + " WHERE tanggal_pembelian BETWEEN '" + tanggal_awal + "' AND '" + tanggal_akhir + "'";
121
122         koneksi = jaringan.Jaringan.config();
123         pernyataan = koneksi.createStatement();
124         hasil = pernyataan.executeQuery(sql);
125
126     }
127 }
```

Gambar 2.221 : Fungsi isi_tableTransaksi() pada halaman laporan transaksi (2)

The screenshot shows the continuation of the Java code from the previous screenshot. The code is enclosed in a try-catch block:

```
122
123         koneksi = jaringan.Jaringan.config();
124         pernyataan = koneksi.createStatement();
125         hasil = pernyataan.executeQuery(sql);
126
127     while (hasil.next()) {
128
129         LocalDate tanggalTransaksiRaw = LocalDate.parse(hasil.getString("tanggal_pembelian"), formatterInput);
130         String tanggalTransaksiUbahFormat = tanggalTransaksiRaw.format(formatterOutput);
131
132         tbl.addRow(new Object[]{
133             hasil.getString("id_transaksi"),
134             hasil.getString("total_harga"),
135             tanggalTransaksiUbahFormat,
136             hasil.getString("status_pembayaran"),
137         });
138
139         tabelTransaksi.setModel(tbl);
140
141     } catch (Exception e) {
142         JOptionPane.showMessageDialog(null, "Keterangan Error : " + e);
143     }
144
145 }
146
147 }
```

Gambar 2.222 : Fungsi isi_tableTransaksi() pada halaman laporan transaksi (3)

7. Buat fungsi isi_tableTransaksiSemua(), isi fungsi tersebut.

Gambar 2.223 : Fungsi isi_tableTransaksiSemua() pada halaman laporan transaksi (1)

```
Start Page × Menu_login.java × Menu_UserAdmin.java × Login_User.java × Form_Transaksi.java × AdminToko.java × Form_Kategori.java × Entry_Barang.java × Laporan_Admin.java ×
Source Design History
163         + " FROM tb_transaksi";
164
165     koneksi = jaringan.Jaringan.config();
166     pernyataan = koneksi.createStatement();
167     hasil = pernyataan.executeQuery(sql);
168
169     while (hasil.next()) {
170
171         LocalDate tanggalTransaksiRaw = LocalDate.parse(hasil.getString("tanggal_pembelian"), formatterInput);
172         String tanggalTransaksiUbahFormat = tanggalTransaksiRaw.format(formatterOutput);
173
174         tbl.addRow(new Object[]{
175             hasil.getString("id_transaksi"),
176             hasil.getString("total_harga"),
177             tanggalTransaksiUbahFormat,
178             hasil.getString("status_pembayaran"),});
179     }
180
181     tabelTransaksi.setModel(tbl);
182
183     } catch (Exception e) {
184
185         JOptionPane.showMessageDialog(null, "Keterangan Error :" + e);
186     }
187 }
```

Gambar 2.224 : Fungsi isi_tableTransaksiSemua() pada halaman laporan transaksi (2)

8. Buat fungsi ambilJarakTanggal(), isi fungsi tersebut.

```
188
189     public String[] ambilJarakTanggal() {
190
191         SelectedDate tanggal_awal = dateTanggalAwal.getSelectedDate();
192         SelectedDate tanggal_akhir = dateTanggalAkhir.getSelectedDate();
193
194         String tanggal1 = String.valueOf(tanggal_awal.getDay());
195         String bulan1 = String.valueOf(tanggal_awal.getMonth());
196         String tahun1 = String.valueOf(tanggal_awal.getYear());
197
198         String tanggal2 = String.valueOf(tanggal_akhir.getDay());
199         String bulan2 = String.valueOf(tanggal_akhir.getMonth());
200         String tahun2 = String.valueOf(tanggal_akhir.getYear());
201
202         // pengecekan tanggal
203         if (hitungDigitAngka(Integer.parseInt(tanggal1)) == 1) {
204
205             tanggal1 = "0" + tanggal1;
206         }
207
208         if (hitungDigitAngka(Integer.parseInt(tanggal2)) == 1) {
209
210             tanggal2 = "0" + tanggal2;
211         }
212
213     }
214
215     return new String[] {bulan1, tanggal1, tahun1, bulan2, tanggal2, tahun2};
216 }
```

Gambar 2.225 : Fungsi ambilJarakTanggal() pada halaman laporan transaksi (1)

Gambar 2.226 : Fungsi ambilJarakTanggal() pada halaman laporan transaksi (2)

9. Buat fungsi hitungDigitAngka(), isi fungsi tersebut.

```
Start Page x Menu_Login.java x Menu_UserAdmin.java x Login_User.java x Form_Transaksi.java x AdminToko.java x Form_Kategori.java x Entry_Barang.java x Laporan_Admin.java x
Source Design History 
231
232     private int hitungDigitAngka(int angka) {
233
234         int count = 0;
235
236         while (angka != 0) {
237
238             angka /= 10;
239
240             count++;
241         }
242
243         return count;
244     }

```

Gambar 2.227 : Fungsi hitungDigitAngka() pada halaman laporan transaksi

10. Buat fungsi printData(), isi fungsi tersebut.

Gambar 2.228 : Fungsi printData() pada halaman laporan transaksi (1)

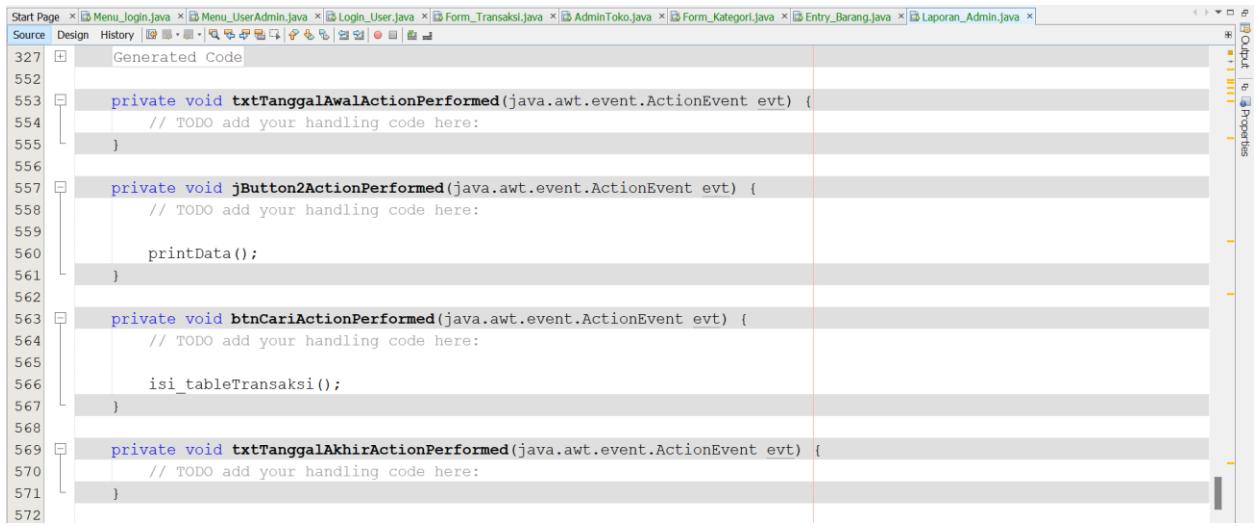
Gambar 2.229 : Fungsi printData() pada halaman laporan transaksi (2)

```
294         JOptionPane.showMessageDialog(null,
295             "Tidak ada data transaksi dari tanggal "
296             + tanggalAwalUbahDate.format(formatterOutput) + " sampai dengan tanggal "
297             + tanggalAkhirUbahDate.format(formatterOutput));
298     } else {
299
300         File report = new File("src/report/report_laporan_admin_final.jasper");
301
302         JasRep = (JasperReport) JRLoader.loadObject(report);
303
304         param.clear();
305         param.put("tanggal_awal", tanggal_awal);
306         param.put("tanggal_akhir", tanggal_akhir);
307         param.put("tanggal_awal_tampil", tanggalAwalUbahDate.format(formatterOutput));
308         param.put("tanggal_akhir_tampil", tanggalAkhirUbahDate.format(formatterOutput));
309
310         JasPri = JasperFillManager.fillReport(JasRep, param, koneksi);
311         JasperViewer.viewReport(JasPri, false);
312         JasperViewer.setLookAndFeelDecorated(true);
313     }
314 } catch (Exception e) {
315
316     JOptionPane.showMessageDialog(null, "Keterangan Error :" + e);
317 }
318 }
```

Gambar 2 230 : Fungsi printData() pada halaman laporan transaksi (3)

11. Buat fungsi event. Fungsi event ini digunakan agar element pada halaman jframe dapat berjalan sesuai dengan event yang anda tentukan. Untuk membuat event, anda bisa ke tab design => tekan kanan mouse pada elemen yang ingin kasih event => pilih menu event => pilih event yang anda inginkan. Event yang tersedia untuk elemen itu bermacam – macam,

sesuaikan dengan kebutuhan anda. Berikut beberapa contoh event yang digunakan pada halaman laporan transaksi:



The screenshot shows the NetBeans IDE interface with the code editor open. The code is part of a Java class and contains several event handlers:

```
327 Generated Code
552
553     private void txtTanggalAwalActionPerformed(java.awt.event.ActionEvent evt) {
554         // TODO add your handling code here:
555     }
556
557     private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
558         // TODO add your handling code here:
559         printData();
560     }
561
562     private void btnCariActionPerformed(java.awt.event.ActionEvent evt) {
563         // TODO add your handling code here:
564
565         isi_tableTransaksi();
566     }
567
568     private void txtTanggalAkhirActionPerformed(java.awt.event.ActionEvent evt) {
569         // TODO add your handling code here:
570     }
571
572
```

Gambar 2.231 : Fungsi Event dan isinya pada halaman laporan transaksi (1)



The screenshot shows the NetBeans IDE interface with the code editor open, continuing from the previous code block. It includes more event handlers:

```
572
573     private void tabelTransaksiMouseClicked(java.awt.event.MouseEvent evt) {
574         // TODO add your handling code here:
575
576         sistemTabelTransaksi();
577     }
578
579     private void btnKeluarActionPerformed(java.awt.event.ActionEvent evt) {
580         // TODO add your handling code here:
581
582         dispose();
583     }
584
585     private void btnRefreshActionPerformed(java.awt.event.ActionEvent evt) {
586         // TODO add your handling code here:
587
588         dispose();
589         new Laporan_Admin().setVisible(true);
590     }
591
592     private void btnBackActionPerformed(java.awt.event.ActionEvent evt) {
593         // TODO add your handling code here:
594
595         dispose();
596         new AdminToko().setVisible(true);
597     }
```

Gambar 2.232 : Fungsi Event dan isinya pada halaman laporan transaksi (2)

P. Pembuatan Exe File

Jika program yang sudah anda buat, pastinya anda ingin program itu berjalan tanpa netbeans kan, untuk melakukan itu anda harus membuat suatu software atau exe file. Untuk membuat

exe file anda membutuhkan suatu software yang namanya Advance Installer, software ini digunakan untuk membuat file installer, yang nantinya memungkinkan kita untuk menjalankan program yang sudah anda buat selayaknya menjalankan software seperti biasanya. Untuk membuat exe file dari program yang sudah anda buat, ada beberapa langkah yang harus dilakukan, antara lain:

1. Clean & Build project Halaman Login. Untuk mengebuild project anda bisa mengunjungi tool yang ada di gambar 2.233.



Gambar 2.233 : Tool untuk clean & build

Tools untuk clean & build ada di sebelah kiri dari ikon tombol play.

2. Jika build sudah berhasil, di bagian output akan muncul tulisan ‘Build Successful’ seperti gambar 2.234.

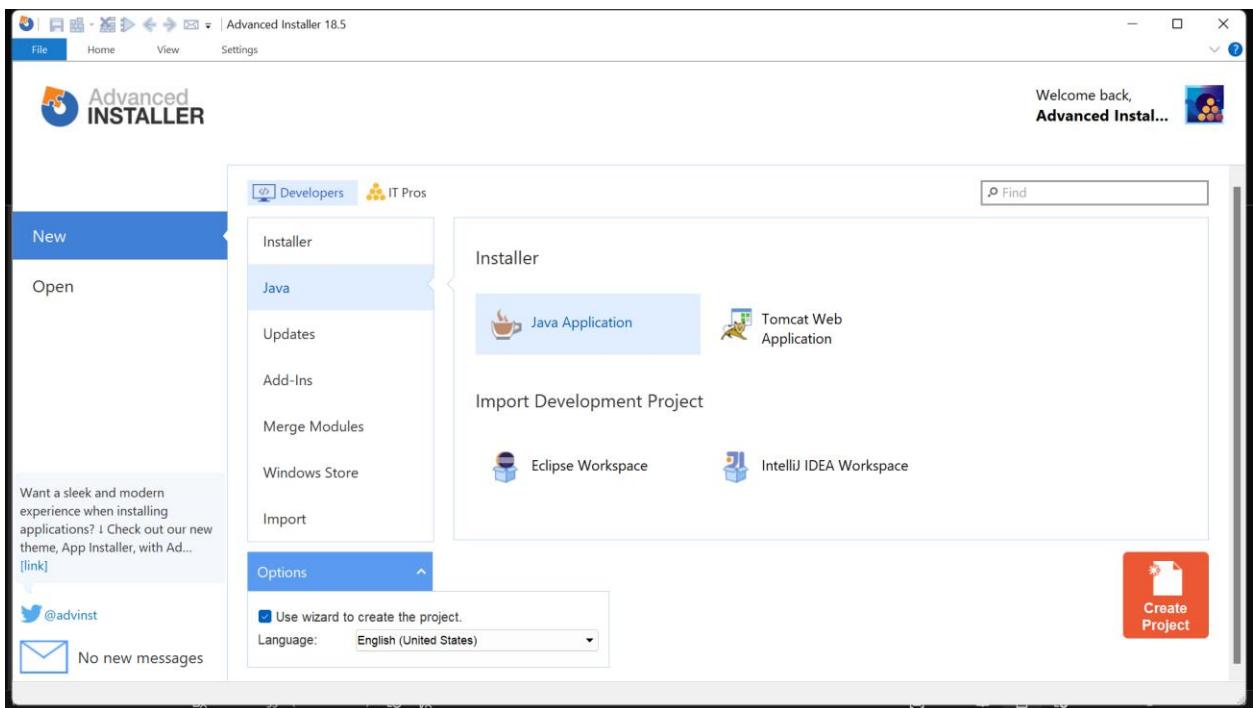
The screenshot shows the Eclipse IDE's Output view with the title 'Output - HalamanLogin (clean,jar)'. The log output is as follows:

```
Created dir: F:\coding\coding_java\OOP\sekolah\modul_15\tugas_praktikum\HalamanLo
Compiling 18 source files to F:\coding\coding_java\OOP\sekolah\modul_15\tugas_pra
warning: [options] bootstrap class path not set in conjunction with -source 1.7
Note: Some input files use or override a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
Note: Some input files use unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.
1 warning
Copying 179 files to F:\coding\coding_java\OOP\sekolah\modul_15\tugas_praktikum\H
Copied 20 empty directories to 8 empty directories under F:\coding\coding_java\OO
compile:
Created dir: F:\coding\coding_java\OOP\sekolah\modul_15\tugas_praktikum\HalamanLo
Copying 1 file to F:\coding\coding_java\OOP\sekolah\modul_15\tugas_praktikum\Hala
Not copying library C:\Program Files\Java\jdk1.7.0_80 , it's a directory.
Copy libraries to F:\coding\coding_java\OOP\sekolah\modul_15\tugas_praktikum\Hal
Not copying library C:\Program Files\Java\jdk1.8.0_111\jre\lib\javaws.jar, due to
Not copying library C:\Program Files\Java\jdk1.8.0_111\jre\lib\deploy.jar, due to
Not copying library C:\Program Files\Java\jdk1.8.0_111\jre\lib\plugin.jar, due to
Building jar: F:\coding\coding_java\OOP\sekolah\modul_15\tugas_praktikum\HalamanL
To run this application from the command line without Ant, try:
java -jar "F:\coding\coding_java\OOP\sekolah\modul_15\tugas_praktikum\HalamanLogi
jar:
BUILD SUCCESSFUL (total time: 5 seconds)
```

Gambar 2.234 : Build project berhasil dilakukan

Setelah project berhasil di build akan muncul folder dist pada project halaman_login. Yang mana folder tersebut menyimpan file .jar yang akan dikonversi ke format exe dan beberapa library yang kita gunakan di project.

3. Buka software Advance Installer.
4. Cari Tab java, dan centang use wizard to create project => jika sudah tekan tombol ‘create project’.



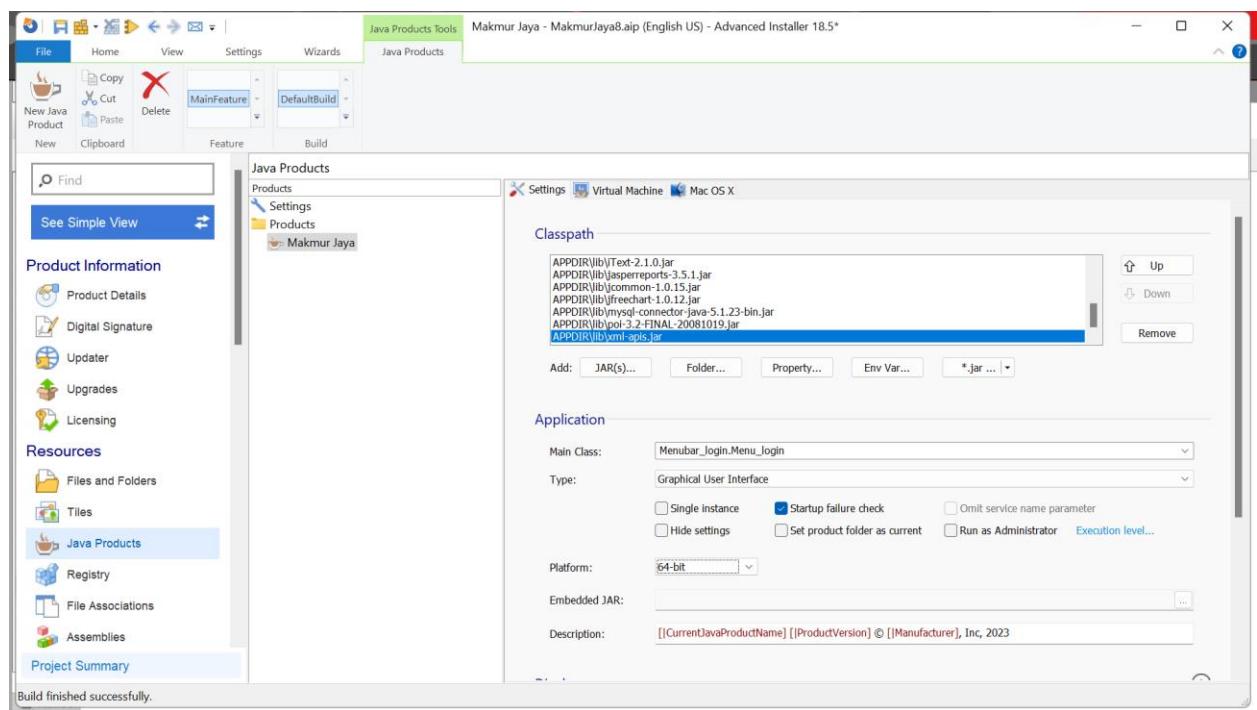
Gambar 2.235 : Halaman home screen software advance installer

5. Jika wizard sudah muncul, isi nama produk dan company name, tekan next => pilih exe setup file => pada bagian installer analytics, hilangkan centang pada track “nama aplikasi” install => pilih tempat penyimpanan project advance installer dan tempat penyimpanan exe file => tambahkan folder dist dari project yang sudah anda buat, jangan lupa untuk mencentang synchronized folder => Jika semua library sudah tercentang dan nama main class sudah sesuai, tekan next => pilih GUI => minimum java version taruh di 1.4, untuk maximum taruh di 1.8. preferred taruh any saja => centang create shortcut di start menu dan desktop => hilangkan centang pada Create a Mac OS package => lompati => Pilih Tema installer sesuai dengan keinginan anda => Untuk bahasa penulis menyarakannya hanya

menggunakan bahasa inggris saja untuk bahsa instalasi => kalau anda mempunyai file rtf untuk licence and agreement anda bisa menambahkan, tapi untuk disini tidak terlebih dahulu => hilangkan centang pada ‘build the project now’, tekan finish.

6. Untuk menu product details tambahkan icon untuk icon yang muncul di control panel.
Untuk icon yang bisa menggunakan file gambar berformat .ico saja. Tambahkan centang pada pilihan Disable repair.

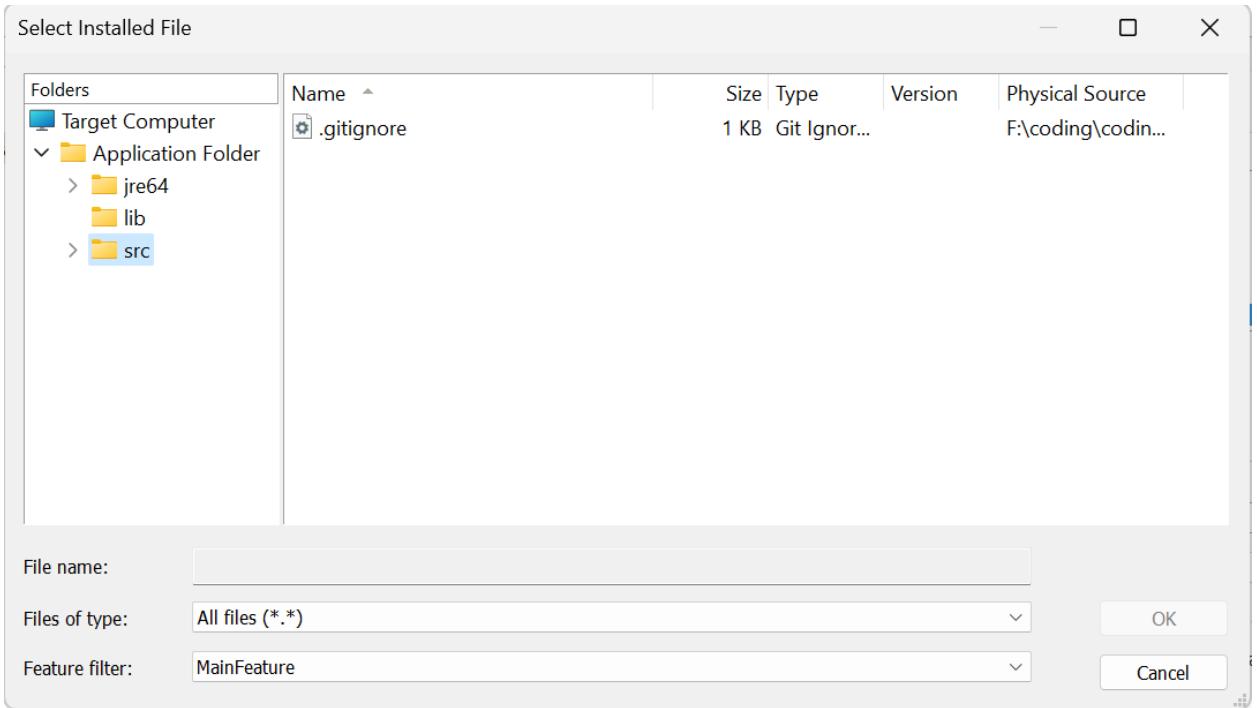
7. Pada menu Java product, dibagian Makmur jaya, pilih platform nya 64 bit.



Gambar 2.236 : Platform instalasi produk berubah menjadi 64 bit

Masuk pada bagian setting ambil jre versi 1.8 yang sudah terinstal di sistem operasi komputer anda. Pada bagian setting ini pilih add JRE, pilih platform 64 bit dan pilih jre 1.8.0. Jika sudah tekan oke.

Pada menu Makmur jaya, tekan tombol add jar => tekan kanan mouse, pilih add folder tambahkan folder src pada folder project halaman login. Maka folder src akan masuk pada list folder installed file.

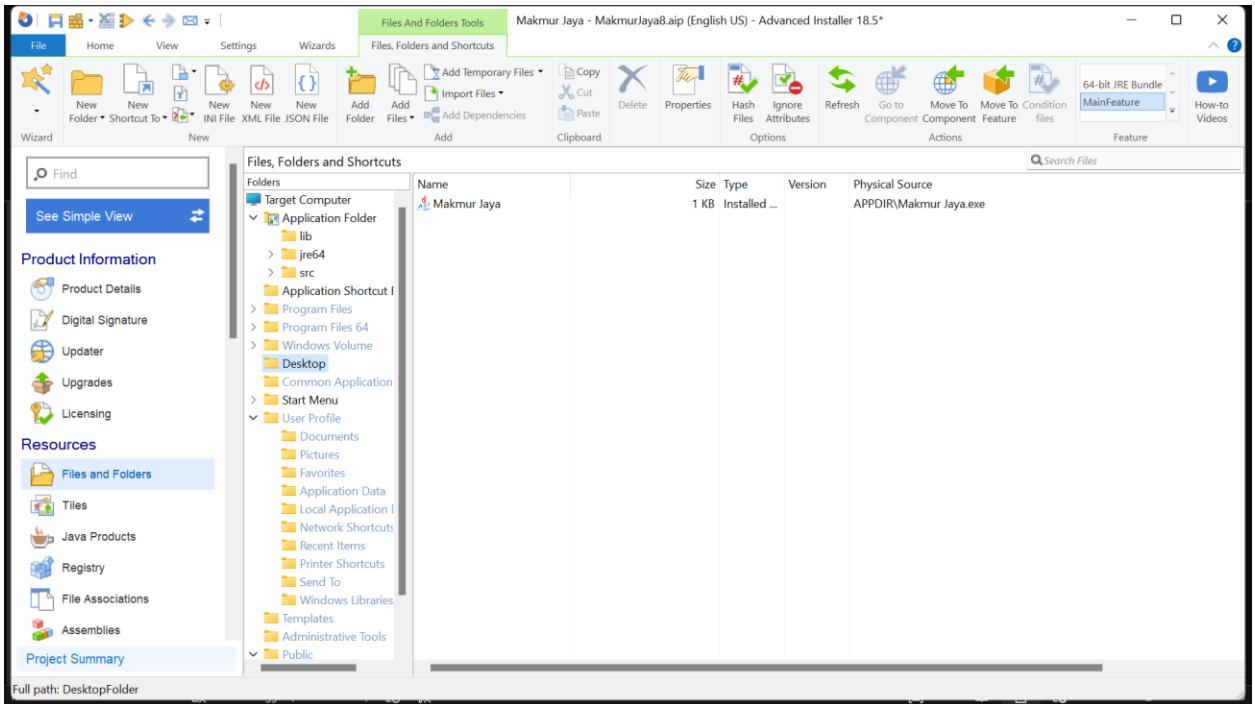


Gambar 2.237 : Folder src sudah masuk pada list installed file.

Buka folder src => report => pilih file format jasper untuk dimasukkan ke classpath, jika sudah klik ok. Ulangi langkah ini sampai semua file yang diperlukan masuk ke classpath.

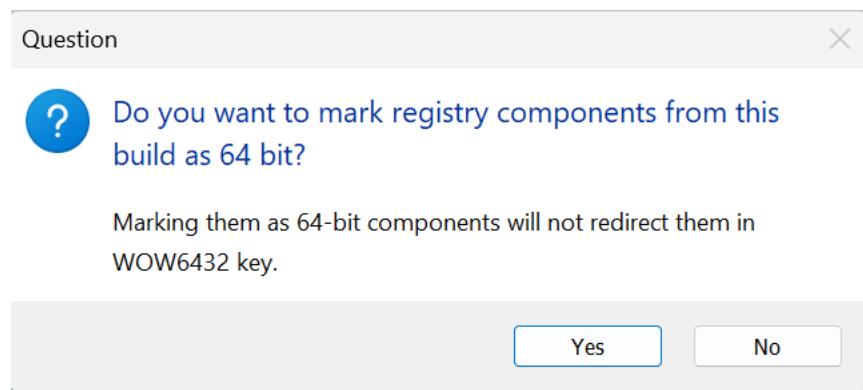
Di bagian display, tambahkan icon file lagi.

8. Pada bagian menu ‘Files and Folder’, copy paste shortcut dari bagian application shortcut ke folder start menu dan desktop.



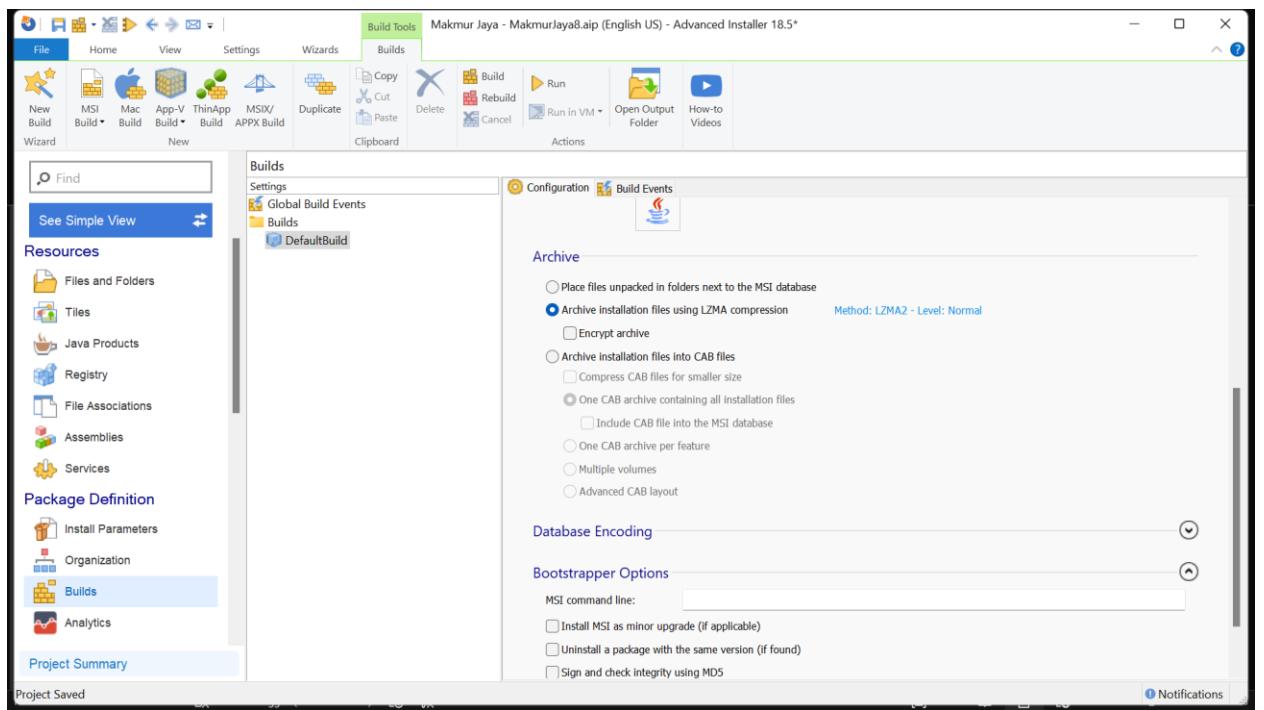
Gambar 2.238 : Shortcut aplikasi Makmur jaya

9. Pada bagian menu install parameter, ganti package type ke '64 bit package for x64 processors'. Jika muncul prompt seperti gambar 2.239, tekan no saja.



Gambar 2.239 : Prompt konfirmasi perubahan package type

10. Pada bagian menu Builds, tambahkan exe name dan tambahkan juga icon exenya. Ubah kompresinya menjadi LZMA.



Gambar 2.240 : Kompresi diubah menjadi LZMA

11. Pada bagian User Interface, anda bisa mengkostumisasi tampilan installernya.
12. Jika sudah semua, kembali ke menu product details, dan tekan tombol build.
13. Exe file sudah dibuat, program sudah bisa digunakan.