LAPORAN TUGAS AKHIR

SISTEM INFORMASI PENGOLAHAN TRANSAKSI LAUNDRY BERBASIS WEB MENGGUNAKAN PHP DAN MYSQL

(Studi Kasus: *IBIE LAUNDRY & WET CLEANING*)

WEB BASED INFORMATION SYSTEM FOR LAUNDRY TRANSACTION USING PHP AND MYSQL

(Case Study : IBIE LAUNDRY & WET CLEANING)

ELOK NADIA ALIFAH

14/370039/SV/07546

PROGRAM STUDI D3 ILMU KOMPUTER DAN SISTEM INFORMASI

DEPARTEMEN TEKNIK ELEKTRO DAN INFORMATIKA

SEKOLAH VOKASI

UNIVERSITAS GADJAH MADA

YOGYAKARTA

2017

LAPORAN TUGAS AKHIR

SISTEM INFORMASI PENGOLAHAN TRANSAKSI LAUNDRY BERBASIS WEB MENGGUNAKAN PHP DAN MYSQL

(Studi Kasus: *IBIE LAUNDRY & WET CLEANING*)

WEB BASED INFORMATION SYSTEM FOR LAUNDRY TRANSACTION USING PHP AND MYSQL

(Case Study : IBIE LAUNDRY & WET CLEANING)

Diajukan untuk memenuhi salah satu syarat memperoleh derajat

Ahli Madya Komputer dan Sistem Informasi

ELOK NADIA ALIFAH

14/370039/SV/07546

PROGRAM STUDI D3 ILMU KOMPUTER DAN SISTEM INFORMASI

DEPARTEMEN TEKNIK ELEKTRO DAN INFORMATIKA

SEKOLAH VOKASI

UNIVERSITAS GADJAH MADA

YOGYAKARTA

2017

HALAMAN PENGESAHAN
LAPORAN TUGAS AKHIR

SISTEM INFORMASI PENGOLAHAN TRANSAKSI LAUNDRY BERBASIS WEB MENGGUNAKAN PHP DAN MYSQL (STUDI KASUS: IBIE LAUNDRY & WET CLEANING)

disusun oleh:

Elok Nadia Alifah
14/370039/SV/07546

Diajukan untuk memenuhi salah satu syarat memperoleh gelar Ahli Madya
Pada Program Studi Diploma III Ilmu Komputer dan Sistem Informasi,
Departemen Teknik Elektro dan Informatika,
Sekolah Vokasi,
Universitas Gadjah Mada

Diterima dan disetujui oleh,

Ketua Penguji,

Umar Taufiq, M. Cs
NIP. 1120160190

Sekretaris Penguji,

Muhammad Rifqi A. M. Sc
NIP. 1120160176

Anggota Penguji/Dosen Pembimbing

Imam Fahrurrozi, S.T., M.Cs

Mengetahui,

Ketua Departemen
Teknik Elektro dan Informatika

Ir. Lukman Subekti, M.T.
NIP. 196210301993031002

Ketua Program Studi
Diploma III Ilmu Komputer dan Sistem Informasi

Dr. Sigit Priyanta, S.Si., M.Kom
NIP. 197704012002121002

PERNYATAAN

Dengan ini saya menyatakan bahwa Laporan Tugas Akhir ini tidak pernah diajukan untuk memperoleh gelar Ahli Madya di suatu Perguruan Tinggi, dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka.

Yogyakarta, 6 Februari 2017

ELOK NADIA ALIFAH

KATA PENGANTAR

Alhamdulillah, puji dan syukur penulis panjatkan kepada Allah SWT yang telah melimpahkan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan Laporan Tugas Akhir yang berjudul "Sistem Informasi Pengolahan Transaksi Laundry Berbasis Web Menggunakan PHP dan MySql (Studi Kasus: *Ibie Laundry & Wet Cleaning*)" ini sebagai salah satu syarat untuk memperoleh gelar Ahli Madya pada jenjang pendidikan Diploma 3 Komputer dan Sistem Informasi Sekolah Vokasi Universitas Gadjah Mada.

Laporan ini disusun sebagai syarat untuk memperoleh gelar Ahli Madya pada jenjang pendidikan Diploma 3 Komputer dan Sistem Informasi Sekolah Vokasi Universitas Gadjah Mada ini dapat terselesaikan berkat dorongan, dukungan, bimbingan, serta bantuan dari berbagai pihak, baik secara langsung maupun secara tidak langsung. Untuk itu perkenankanlah penulis mengucapkan terima kasih kepada:

1. Ayah, bunda, adik Ayu, adik Cantika dan eyang tercinta yang selalu memberikan semangat, dukungan, cinta, kasih sayang dan doa untuk kesuksesan dalam mengerjakan tugas akhir ini.
2. Bapak Sigit Priyanta, S.Si., M.Kom., selaku Ketua Pengelola Program Diploma 3 Komputer dan Sistem Informasi Sekolah Vokasi UGM.
3. Bapak Imam Fahrurrozi M.Cs selaku dosen pembimbing TA yang tidak bosan-bosan dalam membimbing, mengarahkan, memberi petunjuk, dan juga memberi semangat.
4. Om Aphit Dedy N selaku pemilik *Ibie Laundry & Wet Cleaning*.
5. Randi Aulia Ramadhan yang selalu memberikan semangat, dukungan dan doa untuk segera menyelesaikan tugas akhir.
6. Dwi Rengga Prasetyo yang sudah banyak membantu dalam pengerjaan tugas akhir ini.

7. Teman-teman seperjuangan dalam pengerjaan tugas akhir ini yang senantiasa memberikan semangat dan ajakan untuk segera menyelesaikan tugas akhir ini.
8. Keluarga KOMSI C 2014 UGM yang selalu memberikan rasa kenyamanan dan ketenangan.
9. Keluarga Himakomsi, EPS dan Kewirush yang memberikan keceriaan dan supportnya.
10. Semua pihak yang telah turut andil membantu penulisan Laporan Tugas Akhir ini yang tidak bisa penulis sebutkan satu-persatu.

Penulis menyadari bahwa Laporan Tugas Akhir ini masih jauh dari sempurna, hal ini disebabkan murni karena keterbatasan pengetahuan penulis. Oleh karena itu penulis mengharapkan kritik dan saran yang bersifat membangun. Semoga laporan ini dapat meningkatkan pengetahuan kita serta memberikan manfaat bagi penulis, pembaca dan dunia ilmu komputer pada umunya.

Yogayakarta, 6 Februari 2017

ELOK NADIA ALIFAH

DAFTAR ISI

LAPORAN TUGAS AKHIR i

LAPORAN TUGAS AKHIR ii

HALAMAN PENGESAHAN.....**Error! Bookmark not defined.**

PERNYATAAN iv

KATA PENGANTAR v

DAFTAR ISI vii

DAFTAR GAMBAR xi

DAFTAR TABEL xvii

INTISARI xviii

ABSTRACK xix

BAB I 1

PENDAHULUAN 1

1.1 Latar Belakang 1

1.2 Rumusan Masalah 2

1.3 Batasan Masalah 2

1.4 Tujuan 2

1.5 Manfaat 2

1.6 Metodologi Penelitian 3

1.7 Sistematika Penulisan 4

BAB II 7

TINJAUAN PUSTAKA 7

BAB III 9

LANDASAN TEORI 9

3.1 Konsep Dasar Sistem Informasi 9

3.1.1 Konsep Dasar Sistem 9
3.1.2 Konsep Dasar Data dan Informasi 11
3.1.3 Sistem Informasi 12
3.2 Daur Hidup Sistem 14
3.2.1 Analisis Sistem 16
3.2.2 Perancangan Sistem 17
3.2.3 Implementasi Sistem 18
3.2.4 Pembaruan Sistem 18
3.3 Basis Data 19
3.3.1 Konsep Dasar Basis Data 19
3.3.2 Model Basis Data 19
3.3.3 Structured query language (SQL) 22
3.4 World Wide Web (WWW) 23
3.4.1 Web Browser 24
3.4.2 Web Server 24
3.5 Perangkat Pengembangan Sistem 25
3.6.1 HTML (Hyper Text Markup Language) 25
3.6.2 PHP (PHP Hypertext Preprocessor) 25
3.6.3 Model View Controller (MVC) 26
3.6.4 Framework Code Igniter 27
3.6.5 Javascript 28
3.6.6 Jquery 29
3.6.7 MySQL 30
3.6 UML *(Unifed Modeling Language)* 31
3.6.1 *Use Case* Diagram 32
3.6.2 *Activity* Diagram 33
3.6.3 *Class* Diagram 34
3.7 Profil Umum 35
BAB IV 37
ANALISI DAN PERANCANGAN SISTEM 37

4.1 Analisis Sistem 37
4.1.1 Analisis sistem yang ada 37
4.1.2 Analisis pengguna sistem 37
4.1.3 Analisis kebutuhan sistem 38
4.2 Perancangan Sistem 39
4.2.1 Perancangan Proses 39
4.2.2 Perancangan basis data 90
4.2.3 Perancangan struktur menu 97
4.2.4 Perancangan antarmuka pengguna 97
BAB V 125
IMPLEMENTASI SISTEM 125
5.1 Spesifikasi Sistem 125
5.2 Implementasi Database 125
5.3 Implementasi Antarmuka Proses 129
5.3.1 Koneksi basis data 129
5.3.2 Halaman login 129
5.3.3 Halaman dashboard 131
5.3.4 Halaman transaksi 132
5.3.5 Halaman barang 139
5.3.6 Halaman perlengkapan 145
5.3.7 Halaman data perlengkapan 150
5.3.8 Halaman *customer* 155
5.3.9 Halaman laporan 161
BAB VI 171
PENGUJIAN DAN PEMBAHASAN 171
6.1 Uji Coba Kemampuan Sistem 171
6.1.1 Halaman Login 171
6.1.2 Halaman Dashboard 172
6.1.3 Halaman Transaksi 173
6.1.4 Halaman Barang 175

6.1.5 Halaman Perlengkapan 177
6.1.6 Halaman Data Perlengkapan 179
6.1.7 Halaman *Customer* 181
6.1.8 Halaman Laporan *Omzet* 183
6.1.9 Halaman Laporan *Profit* 185
6.2 Pengujian Oleh Calon Pengguna Sistem 186
6.3 Hasil Pengujian Sistem 187
BAB VII 188
PENUTUP 188
7.1 Kesimpulan 188
DAFTAR PUSTAKA 189
LAMPIRAN 191

DAFTAR GAMBAR

Gambar 3. 1 Siklus Informasi (Jogiyanto, 2005) 12
Gambar 3. 2 Daur Hidup Sistem 16
Gambar 3. 3 Komponen MVC 26
Gambar 3. 4 Contoh penggunaan javascript (www.w3schools.com) 29
Gambar 3. 5 Contoh kode JQuery 29
Gambar 3. 6 Stuktur organisasi 36

Gambar 4. 1 *Use case* diagram 40
Gambar 4. 2 *Activity* diagram *admin* tambah transaksi 41
Gambar 4. 3 *Activity* diagram *admin* hapus transaksi 42
Gambar 4. 4 *Activity* diagram *admin* lihat *detail* transaksi 43
Gambar 4. 5 *Activity* diagram *admin* lihat *detail* transaksi 44
Gambar 4. 6 *Activity* diagram *admin* kirim *detail* transaksi 45
Gambar 4. 7 *Activity* diagram *admin* update transaksi 46
Gambar 4. 8 *Activity* diagram *admin* tambah barang 47
Gambar 4. 9 *Activity* diagram *admin* edit barang 48
Gambar 4. 10 *Activity* diagram *admin* hapus barang 49
Gambar 4. 11 *Activity* diagram *admin* lihat *detail* barang 50
Gambar 4. 12 *Activity* diagram *admin* cetak data barang 51
Gambar 4. 13 *Activity* diagram *admin* tambah data perlengkapan 52
Gambar 4. 14 *Activity* diagram *admin* edit data perlengkapan 53
Gambar 4. 15 *Activity* diagram *admin* hapus data perlengkapan 54
Gambar 4. 16 *Activity* diagram *admin* lihat *detail* data perlengkapan 55
Gambar 4. 17 *Activity* diagram *admin* cetak data perlengkapan 56
Gambar 4. 18 *Activity* diagram *admin* tambah perlengkapan 57
Gambar 4. 19 *Activity* diagram *admin* edit perlengkapan 58
Gambar 4. 20 *Activity* diagram *admin* hapus perlengkapan 59
Gambar 4. 21 *Activity* diagram *admin* lihat *detail* perlengkapan 60
Gambar 4. 22 *Activity* diagram *admin* cetak perlengkapan 61
Gambar 4. 23 *Activity* diagram *admin* tambah *customer* 62
Gambar 4. 24 *Activity* diagram *admin* edit *customer* 63
Gambar 4. 25 *Activity* diagram *admin* hapus *customer* 64
Gambar 4. 26 *Activity* diagram *admin* lihat *detail customer* 65
Gambar 4. 27 *Activity* diagram *admin* cetak data *customer* 66
Gambar 4. 28 *Activity* diagram *manager* lihat *detail* transaksi 67
Gambar 4. 29 *Activity* diagram *manager* lihat *detail* transaksi 68
Gambar 4. 30 *Activity* diagram *manager* lihat *detail* barang 69
Gambar 4. 31 *Activity* diagram *manager* cetak data barang 70
Gambar 4. 32 *Activity* diagram *manager* tambah data perlengkapan 71

Gambar 4. 33 *Activity* diagram *manager* edit data perlengkapan 72
Gambar 4. 34 *Activity* diagram *manager* hapus data perlengkapan..... 73
Gambar 4. 35 *Activity* diagram *manager* lihat *detail* data perlengkapan..... 74
Gambar 4. 36 *Activity* diagram *manager* cetak data perlengkapan..... 75
Gambar 4. 37 *Activity* diagram *manager* tambah perlengkapan 76
Gambar 4. 38 *Activity* diagram *manager* edit perlengkapan..... 77
Gambar 4. 39 *Activity* diagram *manager* hapus perlengkapan 78
Gambar 4. 40 *Activity* diagram *manager* lihat *detail* perlengkapan..... 79
Gambar 4. 41 *Activity* diagram *manager* cetak perlengkapan 80
Gambar 4. 42 *Activity* diagram *manager* lihat *detail customer* 81
Gambar 4. 43 *Activity* diagram *manager* cetak data *customer*..... 82
Gambar 4. 44 *Activity* diagram *manager* pilih laporan *omzet*..... 83
Gambar 4. 45 *Activity* diagram *manager* cetak laporan *omzet*..... 84
Gambar 4. 46 *Activity* diagram *manager* pilih laporan pengeluaran 85
Gambar 4. 47 *Activity* diagram *manager* cetak laporan pengeluaran 86
Gambar 4. 48 *Activity* diagram *manager* pilih laporan *profit* 87
Gambar 4. 49 *Activity* diagram *manager* cetak laporan *profit* 88
Gambar 4. 50 *Class* Diagram 89
Gambar 4. 51 Entity Relationship Diagram (ERD) 91
Gambar 4. 52 Rancangan Relasi Antar Tabel 96
Gambar 4. 53 Struktur menu *admin* 97
Gambar 4. 54 Struktur menu *manager*..... 97
Gambar 4. 55 Rancangan halaman login 98
Gambar 4. 56 Rancangan halaman dashboard *admin* 99
Gambar 4. 57 Rancangan halaman dashboard *manager*..... 100
Gambar 4. 58 Rancangan halaman transaksi *admin* 101
Gambar 4. 59 Rancangan halaman transaksi *manager* 101
Gambar 4. 60 Rancangan halaman tambah transaksi..... 102
Gambar 4. 61 Rancangan halaman *detail* transaksi *admin* 102
Gambar 4. 62 Rancangan halaman *detail* transaksi *manager* 103
Gambar 4. 63 Rancangan halaman barang *admin*..... 104
Gambar 4. 64 Rancangan halaman barang *manager* 105
Gambar 4. 65 Rancangan halaman tambah barang..... 105
Gambar 4. 66 Rancangan halaman edit barang..... 106
Gambar 4. 67 Rancangan halaman *detail* barang *admin*..... 106
Gambar 4. 68 Rancangan halaman *detail* barang *manager* 107
Gambar 4. 69 Rancangan halaman perlengkapan *admin* 108
Gambar 4. 70 Rancangan halaman perlengkapan *manager*..... 109
Gambar 4. 71 Rancangan halaman tambah perlengkapan *admin* 109
Gambar 4. 72 Rancangan halaman tambah perlengkapan *manager* 110

Gambar 4. 73 Rancangan halaman edit perlengkapan *admin* 110
Gambar 4. 74 Rancangan halaman edit perlengkapan *manager* 111
Gambar 4. 75 Rancangan halaman *detail* perlengkapan *admin* 111
Gambar 4. 76 Rancangan halaman *detail* perlengkapan *manager* 112
Gambar 4. 77 Rancangan halaman data perlengkapan *admin* 113
Gambar 4. 78 Rancangan halaman data perlengkapan *manager* 114
Gambar 4. 79 Rancangan halaman tambah data perlengkapan *admin* 114
Gambar 4. 80 Rancangan halaman tambah data perlengkapan manage 115
Gambar 4. 81 Rancangan halaman edit data perlengkapan *admin* 115
Gambar 4. 82 Rancangan halaman edit data perlengkapan *manager* 116
Gambar 4. 83 Rancangan halaman *detail* data perlengkapan admin 116
Gambar 4. 84 Rancangan halaman *detail* data perlengkapan *manager* 117
Gambar 4. 85 Rancangan halaman *customer admin* 118
Gambar 4. 86 Rancangan halaman *customer manager* 119
Gambar 4. 87 Rancangan halaman tambah *customer* 119
Gambar 4. 88 Rancangan halaman edit *customer* 120
Gambar 4. 89 Rancangan halaman *detail customer admin* 120
Gambar 4. 90 Rancangan halaman *detail customer manager* 121
Gambar 4. 91 Rancangan halaman jangka waktu laporan 122
Gambar 4. 92 Rancangan halaman laporan *omzet* 123
Gambar 4. 93 Rancangan halaman laporan pengeluaran 123
Gambar 4. 94 Rancangan halaman laporan *profit* 124

Gambar 5. 1 Implementasi Database laundry 125
Gambar 5. 2 Implementasi tabel user 126
Gambar 5. 3 Implementasi tabel barang 126
Gambar 5. 4 Implementasi tabel *customer* 127
Gambar 5. 5 Implementasi tabel data perlengkapan 127
Gambar 5. 6 Implementasi tabel *detail* perlengkapan 127
Gambar 5. 7 Implementasi tabel perlengkapan 128
Gambar 5. 8 Implementasi tabel *detail* transaksi 128
Gambar 5. 9 Implementasi tabel transaksi 128
Gambar 5. 10 Source code koneksi database 129
Gambar 5. 11 Implementasi halaman login 130
Gambar 5. 12 Potongan kode login 130
Gambar 5. 13 Implementasi halaman dashboard *admin* 131
Gambar 5. 14 Implementasi halaman dashboard *manager* 131
Gambar 5. 15 Potongan kode halaman dashboard *admin* 131
Gambar 5. 16 Potongan kode halaman dashboard *manager* 132
Gambar 5. 17 Implementasi halaman transaksi *admin* 132

Gambar 5. 18 Implementasi halaman transaksi *manager* 133
Gambar 5. 19 Implementasi halaman tambah transaksi..... 133
Gambar 5. 20 Implementasi halaman *detail* transaksi *admin* 133
Gambar 5. 21 Implementasi halaman *detail* transaksi *manager* 134
Gambar 5. 22 Implementasi halaman cetak transaksi 134
Gambar 5. 23 Implementasi halaman kirim transaksi..... 134
Gambar 5. 24 Potongan kode halaman transaksi *admin* 135
Gambar 5. 25 Potongan kode halaman transaksi *manager* 135
Gambar 5. 26 Potongan kode halaman tambah transaksi 136
Gambar 5. 27 Potongan kode halaman *detail* transaksi *admin* 136
Gambar 5. 28 Potongan kode halaman *detail* transaksi *manager* 137
Gambar 5. 29 Potongan kode halaman cetak transaksi *admin*..... 137
Gambar 5. 30 Potongan kode halaman cetak transaksi *manager* 137
Gambar 5. 31 Potongan kode halaman kirim transaksi..... 138
Gambar 5. 32 Potongan kode halaman kirim transaksi (lanjutan) 139
Gambar 5. 33 Potongan kode halaman kirim transaksi (lanjutan) 139
Gambar 5. 34 Implementasi halaman barang *admin* 140
Gambar 5. 35 Implementasi halaman barang *manager* 140
Gambar 5. 36 Implementasi halaman tambah barang 141
Gambar 5. 37 Implementasi halaman ubah barang 141
Gambar 5. 38 Implementasi halaman *detail* barang *admin*..... 141
Gambar 5. 39 Implementasi halaman *detail* barang *manager* 142
Gambar 5. 40 Implementasi halaman cetak barang 142
Gambar 5. 41 Potongan kode halaman barang *admin*..... 142
Gambar 5. 42 Potongan kode halaman barang *manager* 143
Gambar 5. 43 Potongan kode halaman tambah barang..... 143
Gambar 5. 44 Potongan kode halaman ubah barang..... 144
Gambar 5. 45 Potongan kode halaman *detail* barang *admin* 144
Gambar 5. 46 Potongan kode halaman *detail* barang *manager* 144
Gambar 5. 47 Potongan kode halaman cetak barang *admin* 145
Gambar 5. 48 Potongan kode halaman cetak barang *manager* 145
Gambar 5. 49 Implementasi halaman perlengkapan 146
Gambar 5. 50 Implementasi halaman tambah perlengkapan 146
Gambar 5. 51 Implementasi halaman ubah perlengkapan 147
Gambar 5. 52 Implementasi halaman *detail* perlengkapan 147
Gambar 5. 53 Implementasi halaman cetak perlengkapan..... 147
Gambar 5. 54 Potongan kode halaman perlengkapan 148
Gambar 5. 55 Potongan kode halaman tambah perlengkapan 149
Gambar 5. 56 Potongan kode halaman ubah perlengkapan 149
Gambar 5. 57 Potongan kode halaman *detail* perlengkapan..... 150

Gambar 5. 58 Potongan kode halaman cetak perlengkapan 150
Gambar 5. 59 Implementasi halaman data perlengkapan 151
Gambar 5. 60 Implementasi halaman tambah data perlengkapan 151
Gambar 5. 61 Implementasi halaman ubah data perlengkapan 151
Gambar 5. 62 Implementasi halaman *detail* data perlengkapan 152
Gambar 5. 63 Implementasi halaman cetak data perlengkapan..... 152
Gambar 5. 64 Potongan kode halaman data perlengkapan 152
Gambar 5. 65 Potongan kode halaman tambah data perlengkapan 153
Gambar 5. 66 Potongan kode halaman ubah data perlengkapan 153
Gambar 5. 67 Potongan kode halaman *detail* data perlengkapan 154
Gambar 5. 68 Potongan kode halaman cetak data perlengkapan..... 154
Gambar 5. 69 Implementasi halaman *customer admin*..... 155
Gambar 5. 70 Implementasi halaman *customer manager*..... 156
Gambar 5. 71 Implementasi halaman tambah *customer* 156
Gambar 5. 72 Implementasi halaman ubah *customer* 156
Gambar 5. 73 Implementasi halaman *detail customer admin*..... 157
Gambar 5. 74 Implementasi halaman *detail customer manager*..... 157
Gambar 5. 75 Implementasi halaman cetak *customer* 157
Gambar 5. 76 Potongan kode halaman *customer admin*..... 158
Gambar 5. 77 Potongan kode halaman *customer manager* 158
Gambar 5. 78 Potongan kode halaman tambah *customer*..... 159
Gambar 5. 79 Potongan kode halaman ubah *customer*..... 159
Gambar 5. 80 Potongan kode halaman *detail customer admin*..... 160
Gambar 5. 81 Potongan kode halaman *detail customer manager* 160
Gambar 5. 82 Potongan kode halaman cetak *customer admin* 160
Gambar 5. 83 Potongan kode halaman cetak *customer manager* 161
Gambar 5. 84 Implementasi halaman jangka waktu laporan 162
Gambar 5. 85 Implementasi halaman laporan *omzet* 162
Gambar 5. 86 Implementasi halaman cetak laporan *omzet*..... 162
Gambar 5. 87 Implementasi halaman laporan pengeluaran 163
Gambar 5. 88 Implementasi halaman cetak laporan pengeluaran..... 163
Gambar 5. 89 Implementasi halaman laporan *profit*..... 163
Gambar 5. 90 Implementasi halaman cetak laporan *profit* 164
Gambar 5. 91 Potongan kode halaman jangka waktu laporan 164
Gambar 5. 92 Potongan kode halaman laporan *omzet* 165
Gambar 5. 93 Potongan kode halaman cetak laporan *omzet*..... 166
Gambar 5. 94 Potongan kode halaman laporan pengeluaran 167
Gambar 5. 95 Potongan kode halaman cetak laporan pengeluaran 168
Gambar 5. 96 Potongan kode halaman laporan *profit*..... 169
Gambar 5. 97 Potongan kode halaman cetak laporan *profit* 170

Gambar 6. 1 Tampilan halaman login 171
Gambar 6. 2 Tampilan halaman gagal login 172
Gambar 6. 3 Tampilan halaman dashboard *admin* 172
Gambar 6. 4 Tampilan halaman dashboard *manager* 173
Gambar 6. 5 Tampilan halaman transaksi 173
Gambar 6. 6 Tampilan halaman tambah transaksi 174
Gambar 6. 7 Tampilan halaman *detail* transaksi 175
Gambar 6. 8 Tampilan halaman form kirim *email* 175
Gambar 6. 9 Tampilan halaman barang 176
Gambar 6. 10 Tampilan halaman tambah barang 176
Gambar 6. 11 Tampilan halaman ubah barang 177
Gambar 6. 12 Tampilan halaman *detail* barang 177
Gambar 6. 13 Tampilan halaman perlengkapan 178
Gambar 6. 14 Tampilan halaman tambah perlengkapan 178
Gambar 6. 15 Tampilan halaman ubah perlengkapan 179
Gambar 6. 16 Tampilan halaman *detail* perlengkapan 179
Gambar 6. 17 Tampilan halaman data perlengkapan 180
Gambar 6. 18 Tampilan halaman tambah data perlengkapan 180
Gambar 6. 19 Tampilan halaman ubah data perlengkapan 181
Gambar 6. 20 Tampilan halaman *detail* data perlengkapan 181
Gambar 6. 21 Tampilan halaman *customer* 182
Gambar 6. 22 Tampilan halaman tambah *customer* 182
Gambar 6. 23 Tampilan halaman ubah *customer* 183
Gambar 6. 24 Tampilan halaman *detail customer* 183
Gambar 6. 25 Tampilan halaman jangka waktu laporan *omzet* 184
Gambar 6. 26 Halaman Laporan *Omzet* 184
Gambar 6. 27 Tampilan halaman jangka waktu laporan pengeluaran 185
Gambar 6. 28 Tampilan halaman laporan pengeluaran 185
Gambar 6. 29 Tampilan halaman jangka waktu laporan *profit* 186
Gambar 6. 30 Tampilan halaman laporan *profit* 186

DAFTAR TABEL

Tabel 2. 1 Tabel Perbandingan 8

Tabel 3. 1 Simbol dalam ER-Diagram 20
Tabel 3. 2 Simbol *Use Case* Diagram 33
Tabel 3. 3 Simbol *Activity* Diagram 34
Tabel 3. 4 Simbol *Class* Diagram 35

Tabel 4. 1 Rancangan Tabel User 92
Tabel 4. 2 Rancangan Tabel Barang 92
Tabel 4. 3 Rancangan Tabel *Customer* 93
Tabel 4. 4 Rancangan Tabel Data Perlengkapan 93
Tabel 4. 5 Rancangan Tabel *Detail* Perlengkapan 93
Tabel 4. 6 Rancangan Tabel Perlengkapan 94
Tabel 4. 7 Rancangan Tabel *Detail* Transaksi 94
Tabel 4. 8 Rancangan Tabel Transaksi 95

INTISARI

SISTEM INFORMASI PENGOLAHAN TRANSAKSI LAUNDRY BERBASIS WEB MENGGUNAKAN PHP DAN MYSQL

(Studi Kasus: *Ibie Laundry & Wet Cleaning*)

Oleh :

ELOK NADIA ALIFAH
14/37009/SV/07546

Ibie Laundry & Wet Cleaning merupakan salah satu laundry di Yogyakarta. Pencatatan data transaksi dan penghitungan *omzet*, *profit* masih menggunakan cara manual. Pencatatan data transaksi pada sistem sebelumnya terdapat bagian yang tidak lengkap sehingga terdapat banyak kesalahan, sehingga masih harus menggunakan cara manual untuk pengecekan transaksi. Kemudian, tidak ada fitur untuk penghitungan *omzet* dan *profit* secara otomatis.

Berdasarkan permasalah tersebut perlu dibangun sistem yang mampu menangani proses pencatatan transaksi, pengitungan *omzet* dan *profit*. Sistem ini mampu menangani proses penghitungan *profit* dengan menyinkronkan jumlah omzet dengan pembelian perlengkapan.

Sistem Informasi Pengolahan Transaksi Laundry Berbasis Web Menggunakan Php dan MySql (Studi Kasus: *Ibie Laundry & Wet Cleaning*) ini dibangun dengan menggunakan *framework Code Igniter* dan basis data MySql. Manfaat yang diharapkan dari penelitian ini yaitu sistem dapat melakukan pencatatan data transaksi, penghitungan *omzet* dan *profit* yang lebih terorganisir ketika sistem ini mulai dioperasikan.

Kata kunci : sistem, data, transaksi, *omzet*, *profit*.

ABSTRACK

WEB BASED INFORMATION SYSTEM FOR LAUNDRY TRANSACTION USING PHP AND MYSQL

(Case Study : Ibie Laundry & Wet Cleaning)

By :

ELOK NADIA ALIFAH
14/370039/SV/07546

Ibie Laundry & Wet Cleaning is one of laundry in yogyakarta. The records of data transaction and the profit are still manually calculated. Old data transaction system is still incomplete and there are many mistakes, so the consequences are to use manual way to check the transactions. Then, it lacks feature for calculating the revenue and profit automatically.

Based on this problems, the system needs to be capable of handling the process of the transaction, counting the revenue and profit. This system is able to handle process of calculating the revenue by synchronize the numbers of purchased equipments.

Laundry Processing Transaction Information System is web based using php and mysql (case study: Ibie Laundry & Wet Cleaning) built by using framework code igniter and mysql database. The benefits are expected from this research that the system can do data transaction, revenue and profit calculation more organized.

Keywords : system, data, transaction, revenue, profit.

BAB I
PENDAHULUAN

1.1 Latar Belakang

Ibie Laundry & Wet Cleaning merupakan salah satu laundry yang lebih fokus untuk hotel yang berada di Yogyakarta, letak yang strategis dengan wilayah hotel yaitu di Ruko Rafflesia 2 Blok E, Babarsari, Depok, Sleman, Yogyakarta, membuat banyak pemilik hotel yang menggunakan jasa laundry ini. Usaha ini masih tergolong usaha yang baru, karena berdiri pada tahun 2015. Seiring dengan banyaknya usaha laundry yang ada di Yogyakarta, *Ibie Laundry & Wet Cleaning* terus berbenah agar mampu bersaing dan berkembang.

Dalam hal pengolahan data, laundry ini menggunakan sistem yang sederhana seperti transaksi penyerahan barang, transaksi pengambilan barang, data pelanggan dan lain sebagainya. Pencatatan data menggunakan sistem lama ternyata masih sering terjadi banyak kesalahan karena ada bagian yang terpotong dari sistem tersebut. Penggunaan sistem lama sangat menghambat bagian keuangan untuk mengumpulkan data yang digunakan untuk menghitung *omzet, profit,* dan lain sebagainya sehingga pencatatan tersebut harus dilakukan secaara manual.

Pencatatan data yang dilakukan secara manual menyebabkan sering terjadi kekeliruan yang disebabkan oleh *human error* dan tidak jarang ada keterangan yang hilang serta data yang sama dicatat beberapa kali yang menyebabkan redundansi data. Selain itu, tentu saja waktu yang dibutuhkan untuk pencatatan data relatif lebih lama. Jika hal ini tidak dapat diatasi, maka pemilik laundry akan kewalahan dalam mengambil kebijakan untuk kemajuan usaha karena data yang ada tidak disajikan dengan cepat, tepat, dan akurat.

Berdasarkan latar belakang inilah, pemilik laundry berkeinginan untuk membangun sistem baru yang dapat membantu kegiatan proses usahanya, sehingga dapat menyelesaikan permasalahan-permasalahan yang dihadapi usaha laundry ini. Untuk itu, dalam tugas akhir ini akan dibangun sistem informasi

pengolahan transaksi laundry pada *Ibie Laundry & Wet Cleaning*, sehingga dapat meningkatkan efektivitas dan efisiensi kinerja usaha laundry ini.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dikemukakan di atas, maka rumusan masalah penelitian adalah bagaimana memudahkan pengolahan data untuk menentukan *omzet* dan *profit* pada *Ibie Laundry & Wet Cleaning*. Sehingga membutuhkan suatu sistem yang berfokus pada pengolahan data transaksi dan keuangan yang sesuai dengan kebutuhan *Ibie Laundry & Wet Cleaning.*

1.3 Batasan Masalah

Batasan masalah yang ada pada penelitian adalah sebagai berikut:

a. Studi kasus dilakukan di *Ibie Laundry & Wet Cleaning* Ruko Rafflesia 2 Blok E, Babarsari, Depok, Sleman, Yogyakarta.
b. Sistem yang dibangun tidak menggunakan *interface mobile*.
c. Pengiriman data transaksi *customer* menggunakan *email*.

1.4 Tujuan

Tujuan adanya penelitian ini adalah membangun sistem informasi pengolahan transaksi laundry pada *Ibie Laundry & Wet Cleaning* yang mampu mengelola data transaksi dan keuangan di *Ibie Laundry & Wet Cleaning*.

1.5 Manfaat

Manfaat dari penelitian ini adalah sebagai berikut :

a. Bagi *Ibie Laundry & Wet Cleaning*
    1. Memberi kemudahan dalam melakukan pencatatan dan pengelolaan transaksi.
    2. Memberi kemudahan dalam melakukan pengelolaan keuangan untuk penghitungan *omzet* dan *profit*.
    3. Mempermudah mengirimkan data transaksi kepada *customer* melalui *email* yang terintegrasi didalam sistem.
    4.

b. Bagi Peneliti

1. Dapat menerapkan ilmu yang diperoleh di bangku kuliah ke dalam kasus nyata di masyarakat.
2. Menambah pengetahuan serta kemampuan dalam pengembangan sistem informasi.

1.6 Metodologi Penelitian

Dalam mengembangkan sistem laundry ini, metode yang digunakan untuk memperoleh data-data antara lain :

1. Pengumpulan data

Pengumpulan data adalah bagian penting dalam pembuatan suatu tugas akhir. Dengan data yang lengkap akan menghasilkan sistem yang diinginkan. Adapun metode pengumpulan data dilakukan dengan cara :

a. Studi literature

Metode ini dilakukan dengan studi kepustakaan dengan mencari referensi yang berkaitan dengan tema yang dibahas.

b. Observasi

Metode ini dilakukan dengan pengamatan secara langsung terhadap data-data yang tertulis manual dalam buku data transaksi, data pengambilan barang, data *customer* dan data keuangan.

c. Wawancara

Dengan metode wawancara, diperoleh informasi lebih banyak dan bisa melengkapi observasi yang telah dilakukan sebelumnya. Wawancara dilakukan dengan tanya jawab dengan pihak yang berkaitan dengan kebutuhan penelitian, seperti gambaran sistem yang diharapkan.

2. Pengembangan aplikasi

Metode yang digunakan dalam proses pengembangan sistem ini adalah metode waterfall, berikut adalah metode waterfall menurut Jogiyanto (2005) :

a. Communication

Menganalisis kebutuhan sistem baik kebutuhan fungsional dan kebutuhan non fungsional sistem. Proses analisis diawali dengan mempelajari

informasi-informasi yang berasal dari data tertulis atau arsip yang terdapat di *Ibie Laundry & Wet Cleaning*. Proses selanjutnya adalah melakukan analisis proses bisnis terjadi mulai dari data *customer*, data pengambilan barang, data stok bahan serta transaksi.

b. Planning

Mengestimasi tahapan dan proses untuk pembuatan sistem, kemudian dilanjutkan untuk penjadwalan dari tiap tahapan dan proses agar terdapat acuan untuk menyelesaikan tahapan tepat waktu.

c. Modelling

Merancang proses-proses yang dapat dikerjakan sistem kemudian merancang data yang dibutuhkan dalam proses tersebut kedalam sistem. Desain yang dibuat meliputi desain basis data menggunakan Entity Relationship Diagram (ERD), desain proses yang ada dalam sistem dan perancangan model sistem dengan menggunakan UML, struktur menu dan desain rancangan antarmuka yang terdapat dalam sistem informasi pengolahan transaksi laundry pada *Ibie Laundry & Wet Cleaning.*

d. Construction

Mengimplementasikan hasil perancangan sistem menjadi sebuah aplikasi dengan cara coding dan testing. Metode ini dilakukan dengan mengimplementasikan dan menguji aplikasi secara langsung dan mencari kesalahan atau kekurangan yang ada, sehingga dapat disempurnakan lagi agar sesuai dengan kebutuhan.

e. Deployment

Tahapan ini merupakan tahapan penyerahan aplikasi kepada pemilik *Ibie Laundry & Wet Cleaning* bukti bahwa sistem sudah sesuai dengan keinginan pemilik laundry.

1.7 Sistematika Penulisan

Sistematika penulisan yang digunakan dalam penyusunan tugas akhir ini adalah sebagai berikut :

BAB I PENDAHULUAN

Bab ini menjelaskan tentang latar belakang, perumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, yang digunakan sebagai referensi dalam penelitian tugas akhir, metode penelitian yang akan digunakan dan sistematikan penulisan tugas akhir.

BAB II TINJAUAN PUSTAKA

Bab ini menguraikan tentang informasi beberapa hasil penelitian sudah dilakukan sebelumnya. Tinjauan pustaka berupa tugas akhir, skripsi, makalah, jurnal dan tesis dari beberapa universitas. Kemudian informasi tersebut dijadikan sebagai bahan pembanding dalam pembuatan sistem yang sedang diteliti saat ini.

BAB III DASAR TEORI

Berisi tentang materi – materi, teori ataupun pendapat dari sumber lain yang dapat digunakan sebagai acuan dalam penelitian. Dasar teori ini harus dapat dipertanggungjawabkan secara ilmiah dengan menyertakan sumber – sumber referensi yang sudah diakui kebenarannya.

BAB IV ANALISIS DAN PERANCANGAN SISTEM

Bab ini menjelaskan tentang analisis terhadap sistem dan proses dalam pembuatan sistem. Analisa terhadap sistem yang berjalan saat ini dan sistem yang akan dibuat, perancangan database sistem, pembuatan *Entity Relationship Diagram* (*ERD*) dan *Unifed Modeling Language* (*UML*) sistem, perancangan struktur menu dan perancangan antarmuka pengguna.

BAB V IMPLEMENTASI

Bab ini menguraikan tentang penerapan bab sebelumnya yaitu analisis dan perancangan sistem yang dibuat pada pembangunan sistem dengan menampilkan antarmuka dan potongan kode disertai penjelasan cara kerja sistem.

BAB VI PENGUJIAN DAN PEMBAHASAN

Membahas tentang bagaimana sistem dapat berjalan sesuai dengan yang diharapkan.

BAB VII PENUTUP

Bab penutup berisi tentang kesimpulan hasil penelitian yang diperoleh sesuai dengan tujuan penelitian. Selain itu berisi pula saran sebagai acuan untuk penelitian berikutnya agar dapat menghasilkan aplikasi yang lebih baik.

BAB II
TINJAUAN PUSTAKA

Dalam pembuatan tugas akhir ini, dilakukan tinjauan pada beberapa sumber pustaka dan berikut adalah beberapa sumber pustaka yang berkaitan dengan Sistem Informasi Pengolahan Transaksi Laundry yang digunakan sebagai acuan oleh penulis:

Nugroho (2011) mengembangkan Sistem Informasi pada Jasa Usaha Laundry dengan tujuan untuk membantu pelayanan *customer* melakukan pencucian di laundry dngan memakai sistem *computerized*. Sistem ini berbasis *desktop* dan menggunakan Bahasa Pemrograman Visual Basic 0.6.

Bahasa Pemrograman Visual Basic 0.6 juga digunakan oleh Rukmanasari (2013) untuk mengembangkan Sistem Informasi Jasa Laundry pada D&A Laundry. Sistem Informasi ini bertujuan untuk meningkatkan pelayanan yang lebih baik lagi serta dapat mengolah data dan keuangan dengan mudah, cepat, tepat dan akurat.

Velnita (2012) mengembangkan Sistem Informasi Jasa Laundry dengan Menggunakan Java Swing dan Peristence API di Laundry Nareswari. Sistem ini dirancang untuk mengganti proses pelayanan manual menjadi pelayanan secara terkomputerisasi.

Berdasarkan tinjauan di atas, penulis membuat Sistem Informasi Pengolahan Transaksi Laundry Berbasis Web Menggunakan PHP dan MySql (Studi Kasus: *Ibie Laundry & Wet Cleaning*) dengan mempertimbangkan tinjaun pustaka yang telah dibahas. Sistem Informasi Pengolahan Transaksi Laundry pada *Ibie Laundry & Wet Cleaning* di bangun dengan menggunakan framework code igniter. Sistem ini selain untuk mengolah data transaksi, sistem juga dapat menampilkan laporan *omzet* dan *profit*.

Perbandingan penelitian sebelumnya dengan penelitian yang telah dilakukan dapat dilihat pada Tabel 2.1 .

Tabel 2. 1 Tabel Perbandingan

Penulis	Nugroho (2011)	Velnita (2012)	Rukmanasari (2013)	Penulis (2016)
Judul	Aplikasi Sistem Informasi pada Jasa Usaha Laundry.	Perancangan Sistem Informasi Jasa Laundry dengan Menggunakan Java Swing dan Persistenc API di Laundry Nareswari.	Analisis dan Perancangan Sistem Informasi Jasa Laundry pada D&A Laundry dengan Menggunakan Visual Basic 6.0.	Sistem Informasi Pengolahan Transaksi Laundry Berbasis Web Menggunakan PHP dan MySql (Studi Kasus: *Ibie Laundry & Wet Cleaning*).
Tujuan	Pelayanan *customer* yang melakukan pencucian di laundry dengan menggunakan sistem computerized.	Sarana untuk mengganti proses pelayanan manual menjaadi pelayanan secarakomputerisasi.	Meningkatkan pelayanan yang lebih baik lagi serta dapat mengolah data dan keuangan dengan mudah, cepat, tepat dan akurat.	Mengolah data transaksi, mengolah laporan *omzet* dan *profit*, mengirimkan data transaksi langsung kepada *email customer*.
Development Tools	Visual Basic 6.0.	Java.	Visual Basic 6.0.	PHP Framework Code Igniter.
Basis Data	MySql	MySql	MySql	MySql

BAB III
LANDASAN TEORI

3.1 Konsep Dasar Sistem Informasi

3.1.1 Konsep Dasar Sistem

Sistem pada dasarnya adalah sekelompok komponen dan elemen yang digabungkan menjadi satu untuk mencapai tujuan tertentu. Secara sederhana sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur, komponen, atau variabel-variabel yang terorganisasi, saling berinteraksi, saling tergantung satu sama lain dan terpadu. Sistem bisa berupa abstraksi atau fisis (Gordon B. Davis, 2002). Sistem yang abstrak adalah susunan yang teratur dari gagasan-gagasan atau konsepsi yang saling tergantung. Sedangkan sistem yang bersifat fisis adalah serangkaian unsur yang bekerjasama untuk mencapai suatu tujuan (Tata Sutabri, 2004).

Menurut Jogiyanto (2005), suatu sistem mempunyai karakteristik atau komponen tertentu yaitu:

a. Objek, merupakan bagian, elemen atau variabel yang dapat berupa benda fisik, abstrak atau keduanya.
b. Atribut, merupakan penentu kualitas atau sifat kepemilikian sistem dan objeknya.
c. Hubungan internal, merupakan penghubungan diantara objek-objek yang terdapat dalam sebuah sistem.
d. Lingkungan, merupakan tempat dimana sistem berada.
e. Tujuan, Setiap sistem memiliki tujuan dan tujuan inilah yang menjadi motivasi yang mengarahkan sistem. Tanpa tujuan, sistem menjadi tidak terkendali. Tentu tujuan antara satu sistem dengan sistem yang lain berbeda.
f. Masukan, adalah sesuatu yang masuk ke dalam sistem dan selanjutnya menjadi bahan untuk diproses. Masukan tersebut dapat berupa hal-hal yang tampak fisik (bahan mentah) atau yang tidak tampak (jasa).

g. Proses, adalah bagian yang melakukan perubahan dari masukan menjadi keluaran yang berguna dan lebih bernilai (informasi) atau yang tidak berguna (limbah).
h. Keluaran, adalah hasil dari proses. Pada sistem informasi berupa informasi atau laporan.
i. Batas, adalah pemisah antara sistem dan daerah luar sistem. Batas disini menentukan konfigurasi, ruang lingkup atau kemampuan sistem. Batas juga dapat diubah atau dimodifikai sehingga dapat merubah perilaku sistem.
j. Mekanisme pengendalian dan umpan balik, digunakan untuk mengendalikan masukan atau proses. Tujuannya untuk mengatur agar sistem berjalan sesuai dengan tujuan.

Dari definisi di atas maka dapat diketahui manfaat dari sitem adalah untuk menyatukan atau mengintegrasikan beberapa unsur elemen yang ada dalam suatu ruang lingkup, yang komponen - komponen di dalamnya tidak dapat berdiri sendiri. Komponen – komponen tersebut harus saling berintegrasi dan saling terhubung untuk menjadi satu kesatuan sehingga sasaran dan tujuan dari sistem tersebut dapat tercapai.

Definisi lain dari sistem menurut Dr. Ir. Harijono Djojodihardjo (1984) adalah sekumpulan objek yang mencakup hubungan fungsional antara tiap-tiap objek dan hubungan antara ciri tiap objek, dan yang secara keseluruhan merupakan suatu kesatuan secara fungsional.

3.1.2 Konsep Dasar Data dan Informasi

Data adalah kumpulan fakta yang tidak teroganisir. Pengolahan data akan mengubah data mentah menjadi informasi (Nurwono, 1994). Sedangkan informasi adalah hasil pengolahan data yang diperoleh dari setiap elemen sistem menjadi bentuk yang mudah dipahami oleh penerimanya dan informasi ini menggambarkan kejadiankejadian nyata untuk menambah pemahamannya terhadap fakta-fakta yang ada, sehingga dapat digunakan untuk pengambilan suatu keputusan. Sumber informasi adalah data. Data adalah kenyataan yang menggambarkan kejadian-kejadian dan kesatuan yang nyata (Tata Sutabri, 2004).

Kualitas informasi

Menurut Jogiyanto (2005), kualitas dari suatu inforamsi (quality of information) tergantung dari tiga hal, yaitu:

a. Akurat

   Informasi harus bebas dari kesalahan-kesalahan dan tidak menyesatkan serta harus jelas mencerminkan maksudnya.

b. Tepat waktu

   Informasi yang datang tepat ketika penerima meminta dan tidak boleh terlambat.

c. Relevan

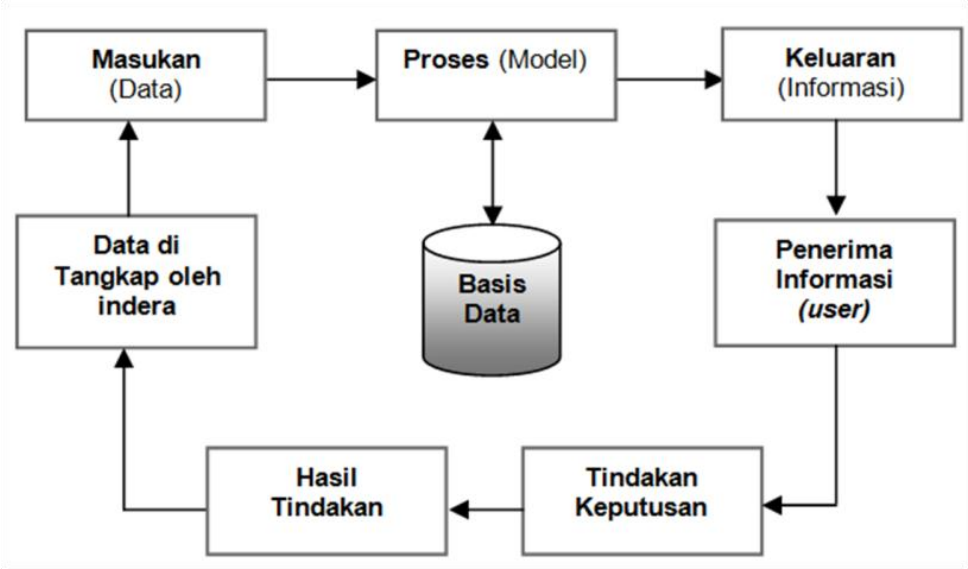
   Informasi tersebut mempunyai manfaat untuk pemakaiannya dan masih layak pakai.

Siklus Informasi

Data diolah melalui suatu model menjadi informasi, penerima kemudian menerima informasi tersebut, kemudian membuat sebuah keputusan dan melakukan tindakan berdasarkan informasi tersebut, yang berarti menghasilkan suatu tindakan yang lain yang akan membuat sejumlah data kembali. Data tersebut akan ditangkap sebagai input kembali untuk diproses kembali lewat suatu model dan seterusnya sehingga proses tersebut membentuk suatu siklus. Siklus ini oleh John Burch disebut dengan siklus informasi

(information cycle). Siklus informasi disebut juga dengan siklus pengolahan data (data processing cycle).

Untuk lebih jelasnya siklus informasi dapat dijelaskan dengan gambar sebagai berikut. Adapun gambar siklus informasi tersebut dapat dilihat pada Gambar 3.1 di bawah ini:



Gambar 3. 1 Siklus Informasi (Jogiyanto, 2005)

3.1.3 Sistem Informasi

Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi organisasi yang bersifat manajerial dalam kegiatan strategi dari suatu organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan laporan – laporan yang diperlukan. (Tata Sutabri,2004).

Komponen Sistem Informasi

Komponen sistem informasi dapat diklasifikasikan sebagai berikut:

1. Perangkat keras (hardware) dan perangkat lunak (software) yang berfungsi sebagai mesin.

2. Manusia (people) dan prosedur (procedures) yang merupakan manusia dan tata cara menggunakan mesin.
3. Data merupakan jembatan penghubung antara manusia dan mesin agar terjadi suatu proses pengolahan data.

Kegiatan Sistem Informasi

Suatu sistem informasi pada dasarnya terbentuk melalui suatu kelompok kegiatan operasi yang tetap, yaitu:

a. Mengumpulkan data
b. Mengelompokkan data
c. Menghitung dan Menganalisa
d. Menyajikan laporan

Manfaat Sistem Informasi

Manfaat adanya sistem informasi dalam suatu instansi yaitu:

a. Meningkatkan penyelesaian tugas.

Pemakai harus lebih produktif agar menghasilkan keluaran yang memiliki mutu yang tinggi.

b. Meningkatkan efektifitas secara keseluruhan.

Sistem harus mudah dan sering digunakan.

c. Meningkatkan efektifitas ekonomi.

Keuntungan yang diperoleh dari sistem harus lebih besar dari biaya yang dikeluarkan.

Klasifikasi Sistem Informasi

Sistem informasi dalam organisasi bertugas mencatat atau merekam berkas yang permanen dan mengelola data untuk menghasilkan informasi yang memiliki nilai agar mendukung proses kerja sebuah organisasi. Sistem informasi datang dalam sebuah bentuk dan ukuran, yang diklasifikasikan berdasarkan fungsi yang dimiliki. Berikut ini merupakan beberapa klasifikasi sistem informasi (Supriyanto, 2005):

a. Sistem pemrosesan transaksi, yaitu memproses transaksi bisnis seperti pemasanan, pembayaran, reservasi, dan sebagainya.
b. Sistem informasi manajemen, yaitu menghasilkan informasi untuk kebutuhan manajerial.
c. Sistem pendukung keputusan, yaitu digunakan untuk membantu pengamblian keputusan atau mengidentifikasi atau memilih antara pilihan dan keputusan.
d. Sistem informasi eksekutif, yaitu untuk kebutuhan eksekutif dalam merencanakan bisnis dan menilai performa terhadap rancangan tersebut.
e. Sistem pakar, yaitu sebuah sistem yang menghasilkan kembali pengetahuan pemecahan masalah ahli ataupun pengambilan keputusan untuk kemudian mensimulasikan pemikiran tersebut.
f. Sistem komunikasi dan kolaborasi, yaitu sistem yang digunakan untuk meningkatkan komunikasi dan kolaborasi antara orang-orang dalam dan luar organisasi.
g. Sistem otomatisasi kantor, yaitu sistem yang membantu karyawan untuk membuat dan berbagi dokumen yang mendukung aktivitas kantor seharihari.

3.2 Daur Hidup Sistem

Siklus hidup sistem (*system life cycle*) adalah proses evolusioner yang diikuti dalam menerapkan sistem atau subsistem informasi berbasis computer. Siklus hidup sistem terdiri dari serangkaian tugas yang erat mengikuti langkah-langkah pendekatan sistem karena tugas-tugastersebut mengikuti pola yang teratur dan dilakukan secara *top down.* Sikulus hidup sistem sering disebut sebagai pendekatan air terjun (*waterfall approach*) bagi pembangunan dan pengembangan sistem (Tata Sutabri, 2003).

Pembangunan sistem hanyalah salah satu dari rangkaian daur hidup suatu sistem. Meskipun demikian, proses ini merupakan aspek yang sangat penting. Kita akan lihat beberapa fase/tahapan dari daur hidup suatu sistem.

Mengenali adanya kebutuhan

Sebelum segala sesuatu terjadi, timbul suatu kebutuhan atau problema yang harus dapat dikenali sebagaimana adanya. Kebutuhan dapat terjadi sebagai hasil perkembangan dari organisasi dan volume yang meningkat melebihi kapasitas dari sistem yang ada. Semua kebutuhan ini harus dapat didefinisikan dengan jelas. Tanpa adanya kejelasan dan kebutuhan yang ada, pembangunan sistem akan kehilangan arah dan efektifitasnya.

Pembangunan Sistem

Suatu proses atau seperangkat prosedur yang harus diikuti untuk menganalisis kebutuhan yang timbul dan membangun suatu sistem untuk dapat memenuhi kebutuhan tersebut.

Pemasangan Sistem

Setelah tahap pembangunan sistem selesai. Sistem kemudian akan dioperasikan. Pemasangan sistem merupakan tahap yang penting pula dalam daur hidup sistem. Peralihan dari tahap pembangunan menuju tahap operasional terjadi pemasangan sistem yang sebenarnya, yang merupakan langkah akhir dari suatu pembangunan sistem.

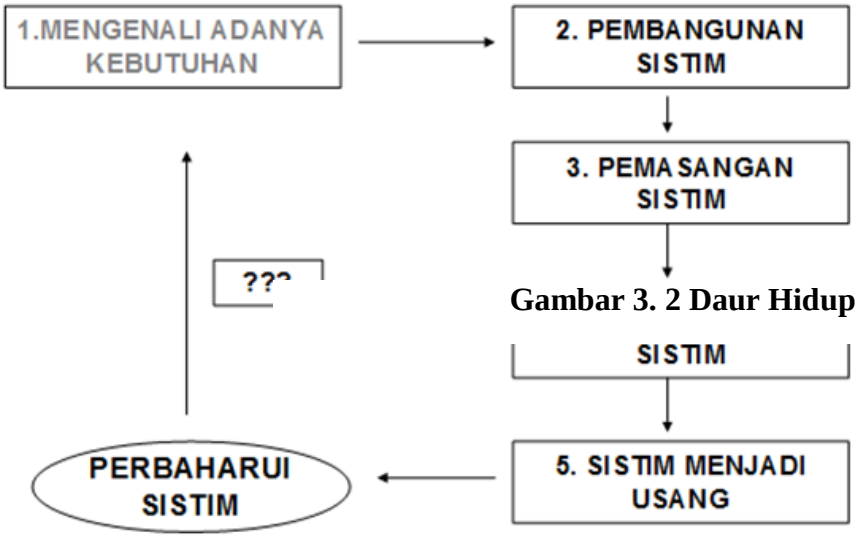
Pengoperasian Sistem

Program-program komputer dan prosedur-prosedur pengoperasian yang membentuk suatu sistem informasi semuanya bersifat statis, sedangkan organisasi ditunjang oleh sistem informasi tadi. Ia selalu mengalami perubahan-perubahan itu karena pertumbuhan kegiatan bisnis, perubahan peraturan, dan kebijaksanaan ataupun kemajuan teknologi. Untuk mengatasi perubahan-perubahan tersebut, sistem harus diperbaiki atau diperbaharui.

Sistem Menjadi Usang

Kadang perubahan yang terjadi begitu drastis sehingga tidak dapat diatasi hanya dengan melakukan perbaikan-perbaikan pada sistem yang berjalan. Tibalah saatnya secara ekonomis dan teknis sistem yang ada sudah tidak layak lagi untuk dioperasikan dan sistem yang baru perlu dibangun untuk menggantikannya.

Sistem informasi kemudian akan melanjutkan daur hidupnya. Sistem dibangun untuk memenuhi kebutuhan yang muncul. Sistem berdaptasi terhadap perubahan-perubahan lingkungan yang dinamis. Sampailah pada kondisi dimana sistem tersebut tidak dapat lagi beradaptasi dengan perubahan-perubahan yang ada ataupun secara ekonomis tidak layak lagi untuk dioperasikan. Sistem yang baru kemudian dibangun untuk menggantikannya. Urutan dari daur hidup sistem dapat dilihat pada Gambar 3.2.



Gambar 3. 2 Daur Hidup Sistem

3.2.1 Analisis Sistem

Sebelum memulai membangun suatu sistem, tahap analisis sistem harus dilalui terlebih dahulu. Tahapan analisis sistem merupakan tahapan yang sangat penting karena kesalahan di dalam tahapan ini akan menyebabkan kesalahan pada tahapan selanjutnya. Proses analisis sistem dalam pengembangan sistem informasi merupakan suatu prosedur yang dilakukan untuk pemeriksaan masalah dan penyusunan pemecahan masalah yang timbul serta membuat spesifikasi sistem yang baru (Tata Sutabri, 2004).

Menurut Abdul Kadir (2003) analisis sistem mencakup analisis kelayakan dan analisis kebutuhan yaitu :

a. Analisis kelayakan

Analisis kelayakan merupakan proses yang mempelajari atau menganalisa permasalahan yang telah ditentukan sesuai dengan tujuan akhir yang akan dicapai. Analisis kelayakan digunakan untuk menentukan kemungkinan keberhasilan solusi yang diusulkan. Tahapan ini berguna untuk memastikan bahwa solusi yang diusulkan tersebut benar-benar dapat tercapai dengan sumber daya dan dengan memperhatikan kendala yang terdapat pada permasalahan serta dampak terhadap lingkungan sekeliling. Lima macam kelayakan dalam merancang sistem informasi yaitu kelayakan teknik, kelayakan ekonomi, kelayakan operasi, kelayakan hukum dan kelayakan jadwal.

b. Analisis kebutuhan

Analisis kebutuhan merupakan proses untuk menghasilkan spesifikasi kebutuhan. Spesifikasi kebutuhan adalah spesifikasi yang rinci tentang pengolahan data yaitu jumlah data yang harus diproses, waktu pengolahan saat data siap diproses sampai informasi yang dihasilkan. Spesifikasi ini digunakan untuk membuat kesepakatan dalam pengembangan sistem.

3.2.2 Perancangan Sistem

Perancangan dapat berupa penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi. Termasuk menyangkut mengkonfigurasikan dari komponen-komponen perangkat lunak dan perangkat keras dari suatu sistem.

Tujuan dari perancangan sistem secara umum adalah untuk memberikan gambaran secara umum kepada *user* dalam merancang sebuah sistem yang baru.

Menurut Jogiyanto (2005) perancangan sistem dibagi menjadi dua bagian, yaitu :

1. *General System Design* yaitu perancangan sistem secara umum *(general)* dari sistem yang akan dibangun.
2. *Detail System Design* yaitu perancangan sistem yang lebih terperinci.

Perancangan sistem bertujuan untuk memenuhi kebutuhan pemakai sistem dan memberikan gambaran rancangan yang jelas dan rancang bangun yang lengkap kepada orang yang membutuhkannya. Kebutuhan-kebutuhan fungsional, penggambaran, perancangan dan pembuatan sketsa dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi terdapat pada tahap perancangan sistem.

3.2.3 Implementasi Sistem

Implementasi sistem atau system implementation adalah proses konstruksi, instalasi, pengujian, dan pengiriman sistem kedalam produksi (Whitten dkk, 2004). Tahap ini merupakan kelanjutan dari tahap perancangan sistem. Tahapan implementasi sistem mencakup pengkodean program dengan menggunakan bahasa pemrograman tertentu, pengujian program, pemasangan program, dan juga pelatihan kepada pengguna. Setelah tahap implementasi ini selesai maka proses pengerjaan sistem telah selesai dan siap untuk memasuki tahap penggunaan.

3.2.4 Pembaruan Sistem

Tahap terakhir adalah tahap pembaruan sistem, dimana pada tahap ini disebut juga tahap maintenance. Hal yang dilakukan pada tahap ini yaitu, salah satunya pembenaran sistem ketika terjadi error saat sistem telah diimplementasikan. Ketika sistem terdeteksi ada kesalahan maka harus segera dilakukan perbaikan sistem untuk menjaga agar kesalahan tidak menjalar ke bagian sistem lainya. Tujuan dari tahap ini agar sistem selalu berjalan dengan normal tanpa terjadi gangguan saat sistem sedang berjalan bahkan sedang digunakan oleh pengguna.

3.3 Basis Data

3.3.1 Konsep Dasar Basis Data

Secara konsep, basis data adalah kumpulan dari data-data yang membentuk suatu berkas (file) yang saling berhubungan (relation) dengan tata cara tertentu untuk membentuk data baru atau informasi. Kumpulan dari data yang saling berhubungan (relasi) antara satu dengan lainnya yang diorganisasikan berdasarkan skema atau struktur tertentu (Supriyanto, 2005).

Menurut Fabbri dan Schwab (Kadir, 2009), basis data adalah sistem berkas terpadu yang dirancang terutama untuk meminimalkan pengulangan data. Secara nyata, basis data merupakan kumpulan berkas-berkas yang memiliki kaitan antar satu dengan yang lain sehingga membentuk suatu bangunan data.

Secara teoritis, basis data tidak harus berurusan dengan komputer. Basis data merupakan suatu koleksi dari data-data yang terorganisasi sedemikian rupa sehingga data mudah disimpan dan dimanipulasi, baik itu diperbarui, dicari, diolah dengan perhitungan-perhitungan tertentu ataupun dihapus (Nugroho, 2011).

3.3.2 Model Basis Data

Model basis data menyatakan hubungan antar rekaman yang tersimpan dalam basis data (Kadir, 2009). Berikut merupakan macam-macam model data:

Model hirarkis (hierarchical model)

Model hirarkis sering disebut model pohon. model ini menggunakan pola hubungan orangtua-anak (parent-child). Simpul yang terhubung ke simpul pada level dibawahnya disebut orangtua (parent). Setiap orangtua bisa memiliki lebih dari satu anak, tetapi anak hanya memiliki satu orangtua. Simpul yang posisinya berada dibawah simpul orangtua disebut anak (child). Simpul orangtua yang tidak memiliki orangtua disebut akar (root). Simpul yang tidak memiliki anak disebut daun. Adapun hubungan antara orangtua dan anak disebut cabang (Kadir, 2009).

Model jaringan (network model)

Model ini menyerupai model hirarkis, namun memiliki perbedaan suatu simpul anak bisa memiliki lebih dari satu orangtua.

Model keterhubungan entitas (Entity-Relationship Model)

Model ini berisi elemen-elemen himpunan entitas dan relasi yang masingmasing dilengkapi dengan atribut-atribut, dan dapat digambarkan dengan menggunakan diagram entity-relationship (diagram E-R). Perancangan basis data seringkali diasosiasikan dengan pembuatan model entity-relationship (Model E-R), dimana kelompok-kelompok data dan relasi antar kelompok-kelompok tersebut diwujudkan dalam bentuk diagram.

E-R diagram merupakan suatu model yang digunakan untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan atau relasi antara objek-objek tersebut. E-R diagram digunakan untuk memodelkan struktur data dan hubungan antara data. E-R diagram menggunakan sejumlah notasi dan simbol untuk menggambarkan struktur dan hubungan data. Simbol-simbol dalam diagram E-R dapat dilihat pada Tabel 3.1.

Tabel 3. 1 Simbol dalam ER-Diagram

Notasi	Keterangan
ENTITAS	**Entitas**, adalah suatu objek yang dapat diidentifikasi dalam lingkungan pemakai.
RELASI	**Relasi**, menunjukkan adanya hubungan di antara sejumlah entitas yang berbeda.
ATRIBUT	**Atribut**, berfungsi mendeskripsikan karakter entitas (atribut yg berfungsi sebagai key diberi garis bawah)
———	**Garis**, sebagai penghubung antara relasi dengan entitas, relasi dan entitas dengan atribut.

Keterangan :

1. Entitas

   Menurut Nugroho (2011), Entitas adalah sesuatu atau objek di dunia nyata (real word) yang dapat dibedakan dengan sesuatu atau objek lainnya. Sebagai contoh, setiap mahasiswa dalam suatu universitas adalah suatu entitas. Suatu entitas biasanya direpresentasikan dengan sejumlah atribut.

2. Atribut

   Menurut Nugroho (2011), atribut adalah properti deskriptif yang dimiliki oleh setiap anggota dari himpunan entitas. Sebagai contoh, entitas mahasiswa atribut-atribut yang mungkin adalah, nim, nama, tanggal lahir, alamat.

3. Kunci

   Menurut Nugroho (2011), kunci merupakan atribut yang unik yang dapat membedakan suatu entitas dengan entitas lainya dalam suatu himpunan entitas. Ada beberapa kunci yang dapat diterapkan pada tabel dalam model basis unik data relasional (Kadir, 2009), yaitu:

   a. Superkey, satu atau lebih atribut yang dapat membedakan setiap baris data dalam sebuah tabel secara unik .
   b. Candidate-key, merupakan kumpulan atribut minimal yang dapat membedakan setiap baris data sebuah tabel secara unik.
   c. Primary-key, merupakan kunci yang diambil dari beberapa candidatekey yang didasari pada:
      1. Kunci tersebut lebih sering dijadikan acuan.
      2. Kunci tersebut lebih ringkas.
      3. Jaminan bahwa kunci tersebut lebih unik dibandingkan dengan kunci yang lain.
   d. Foreign-key, kunci tamu atau kunci asing dari suatu tabel dimana kunci ini juga terdapat pada tabel lain sebagai primary-key.

4. Relasi

Menurut Nugroho (2011), hubungan antara suatu himpunan entitas dengan himpunan entitas yang lainnya. Misalnya, entitas mahasiswa mempunyai hubungan dengan entitas matakuliah (mahasiswa mengambil matakuliah).

5. Kardinalitas

Menurut Pressman (2010), kardinalitas merupakan spesifikasi dari sejumlah peristiwa dari satu objek yang dapat dihubungkan ke sejumlah peristiwa dari objek yang lain. Kardinalitas menunjukan jumlah maksimum entitas yang dapat berelasi dengan entitas pada himpunan entitas yang lain. Entitas merupakan inidvidu yang mewakili sesuatu yang nyata dan dapat dibedakan dengan suatu entitas yang lain.

Menurut Pressman (2010), kardinalitas dapat terjadi antara dua himpunan entitas berupa:

a. Satu ke satu (one to one), terjadi apabila entitas himpunan pertama berelasi hanya dengan satu entitas di himpunan kedua dan sebaliknya.
b. Satu ke banyak (one to many), terjadi apabila entitas himpunan pertama berelasi dengan lebih dari satu entitas di himpunan kedua, tetapi entitas di himpunan kedua berelasi hanya satu entitas di himpunan pertama.
c. Banyak ke satu (many to one), terjadi apabila entitas himpunan pertama berelasi hanya dengan satu entitas di himpunan kedua, tetapi entitas di himpunan kedua berelasi dengan lebih dari satu entitas di himpunan pertama.
d. Banyak ke banyak (many to many), terjadi apabila entitas himpunan pertama berelasi dengan lebih dari satu entitas di himpunan kedua dansebaliknya

3.3.3 Structured query language (SQL)

Meskipun terkandung unsur query didalam namanya serta dirujuk sebagai bahasa query. SQL tidak melulu digunakan untuk melakukan query pada basis data. SQL juga dapat digunakan untuk mendefinisikan struktur data, memodifikasi data pada basis data, menspesifikasi batasan keamanan (security),

hingga pemeliharaan kinerja basis data dan pengaturan pemrosesan pada CPU paralel (Nugroho, 2011).

Menurut Nugroho (2011), secara umum bahasa SQL memiliki beberapa bagian penting, yaitu:

a. Data Definition Language (DDL)

   SQL DDL menyediakan perintah-perintah untuk mendefinisikan skema relasi, menghapus relasi, serta memodifikasi skema relasi.

b. Data Manipulation Language (DML)

   SQL DML mencakup bahasa SQL untuk menyisipkan rekaman pada relasi, menghapus rekaman pada relasi, serta memodifikasi rekaman pada relasi.

c. View Definition

   SQL memuat perintah-perintah untuk mendefinisikan tampilantampilan (view) yang dikehendaki pengguna.

d. Transaction Control

   SQL memuat perintah-perintah untuk menspesifikasi awal dan akhir suatu transaksi

e. Embedded SQL dan Dynamic SQL

   Terminologi ini mencakup kemampuan SQL untuk disisipkan pada beberapa bahasa pemrograman, misalnya pada Visual BASIC, Delphi, C++, Java, PHP dan sebagainya Integrity Authorization

f. Integrity

   SQL DDL mencakup perintah-perintah untuk menspesifikasi batasanbatasan integritas.

g. Authorization

   SQL DDL mencakup perintah-perintah untuk membatasi akses pada basis data demi alasan keamanan.

3.4 World Wide Web (WWW)

World Wide Web biasa disingkat sebagai WWW atau W3, umumnya lebih dikenal sebagai web. Web adalah bagian yang paling menarik dari internet.

Melalui web, para pengguna dapat mengakses informasi-informasi yang tidak hanya berupa teks, tetapi dapat juga berupa gambar, suara, video, dan animasi.

Secara garis besar, web bekerja berdasarkan pada tiga mekanisme berikut ini:

1. Protokol, merupakan aturan standar yang digunakan untuk berkomunikasi pada jaringan, Hypertext Transfer Protocol (HTTP) misalnya, adalah protokol yang bertugas menangani komunikasi dengan halaman web.
2. Alamat, web memiliki aturan penamaan yang disebut dengan Uniform Resource Locator (URL).
3. Halaman web yang dibangun menggunakan Hypertext Markup Language (HTML).

3.4.1 Web Browser

Untuk mengakses halaman web agar dapat ditampilkan kepada pengguna maka dibutuhkan sebuah perangkat lunak yang bernama web browser. Web browser merupakan aplikasi perangkat lunak untuk mengambil, penyajian, dan melintasi sumber informasi di web. Alamat sebuah informasi diidentifikasikan dengan URL. Melalui URL, browser akan menampilkan halaman yang biasanya disebut dengan homepage. Beberapa contoh browser yang sering digunakan oleh masyarakat adalah Firefox, Google Chrome, Internet Explorer, Opera, dan Safari.

3.4.2 Web Server

Web server merupakan server internet yang mampu melayani koneksi transfer data dalam protocol HTTP. Dimana web server merupakan bagian yang terpenting dari server di internet dibandingkan server lainnya seperti e-mail server. Dengan ini kita mempunyai banyak pilihan di dunia open source, tentu saja semuanya itu tergantung pada keperluan sendiri. Dalam penulisan ini penulis menggunakan salah satu web server yang sangat terkenal dan menjadi standart de facto setiap distribusi linux, yaitu Apache.

Apache adalah sebuah nama web server yang bertanggung jawab pada request-response HTTP dan logging informasi secara *detail* (kegunaan basicnya). Selain itu, Apache juga diartikan sebagai suatu web server yang kompak, modular, mengikuti standar protokol HTTP, dan tentu saja sangat digemari. Kesimpulan ini bisa didapatkan dari jumlah pengguna yang jauh melebihi para pesaingnya. Bulan Januari 2005 jumlahnya tidak kurang dari 68% pangsa web server yang berjalan di Internet. Ini berarti jika semua web server selain Apache digabung, masih belum bisa mengalahkan jumlah Apache.

3.5 Perangkat Pengembangan Sistem

Dalam pengembangan sebuah sistem membutuhkan dukungan teknologiteknologi terkini agar mampu menghasilkan solusi permasalahan yang tidak usang dan menjadikan sistem lebih sempurna. Berikut adalah beberapa teknologi yang dipakai dalam pengembangan sistem ini.

3.6.1 HTML (Hyper Text Markup Language)

HTML (Hyper Text Markup Language) adalah sekumpulan simbol – simbol atau tag – tag yang dituliskan dalam sebuah file yang dimaksudkan untuk menampilkan halaman pada web browser. Tag – tag tersebut memberitahukan kepada browser bagaimana menampilkan halaman web dengan lengkap kepada pengguna. Tag – tag HTML selalu diawali dengan <x> dan diakhiri dengan </x> dimana x tag HTML seperti b, i, u dan sebagainya (Astamal, 2006). Sebuah halaman website akan diapit oleh tag <html>……</html>. File-file HTML selalu berakhiran dengan ekstensi *.htm atau *.html. Jadi jika anda mengetik sebuah naskah dan menyimpannya dengan ekstensi *.html maka anda membuat file yang berformat HTML.

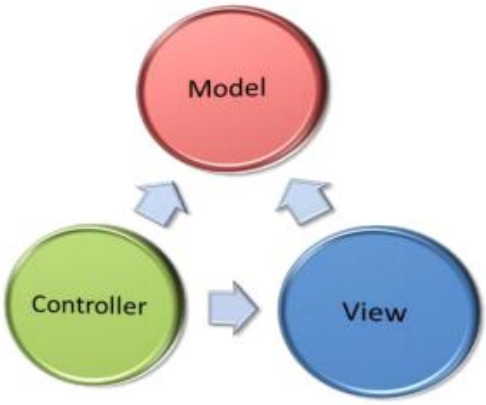
3.6.2 PHP (PHP Hypertext Preprocessor)

PHP adalah bahasa pemrograman yang digunakan secara luas untuk penanganan pembuatan dan pembuatan dan pengembangan sebuah web dan bisa digunakan pada HTML. PHP dibuat pertama kali oleh satu orang yaitu Rasmus

Lerdorf, yang pada awalnya dibuat untuk menghitung jumlah pengunjung pada homepagenya. Diawal Januari 2001, PHP telah dipakai lebih dari 5 juta domain diseluruh dunia, dan akan terus bertambah karena kemudahan aplikasi PHP ini dibandingkan dengan bahasa Server side yang lain. Saat ini PHP sangat populer dan menggantikan Perl yang sebelumnya juga popular sebagai bahasa scripting web. PHP telah menjadi modul Apache terpopuler melebihi FrontPage dan mod_perl.

3.6.3 Model View Controller (MVC)

MVC adalah arsitektur aplikasi yang memisahkan kode-kode aplikasi dalam tiga lapisan, Model, View dan Control. Arsitektur MVC ini memungkinkan adanya perubahan dalam domain model tanpa harus mengubah code untuk menampilkan domain model tersebut. Hal ini sangat bermanfaat ketika aplikasi mempunyai domain model dan view komponen sangat besar dan kompleks. Model ViewController merupakan suatu konsep yang cukup populer dalam pembangunan aplikasi web, MVC memisahkan pengembangan aplikasi berdasarkan komponen utama yang membangun sebuah aplikasi seperti manipulasi data, user interface, dan bagian yang menjadi kontrol aplikasi. Terdapat 3 jenis komponen yang dapat dilihat pada Gambar 3.3 dalam membangun suatu MVC pattern dalam suatu aplikasi.



Gambar 3. 3 Komponen MVC

1. *View*, merupakan bagian yang menangani *presentation logic*. Pada suatu aplikasi web bagian ini biasanya berupa *file template* HTML, yang diatur oleh *controller*. *View* berfungsi untuk menerima dan merepresentasikan data kepada *user*. Bagian ini tidak memiliki akses langsung terhadap bagian model.
2. Model, biasanya berhubungan langsung dengan Database untuk memanipulasi data *(insert, update, delete, search*), menangani validasi dari bagian *controller*, namun tidak dapat berhubungan langsung dengan bagian *view*.
3. *Controller*, merupakan bagian yang mengatur hubungan antara bagian model dan bagian view, *controller* berfungsi untuk menerima *request* dan data dari *user* kemudian menentukan apa yang akan diproses oleh aplikasi.

3.6.4 Framework Code Igniter

Code Igniter adalah aplikasi open source yang berupa *framework* dengan model MVC (Model, *View*, Controller) untuk membangun *website* dinamis dengan menggunakan PHP. CodeIgniter memudahkan *developer* untuk membuat aplikasi *web* dengan cepat dan mudah dibandingkan dengan membuatnya dari awal.

Sebelum membahas *framework* codeigniter, berikut ada sedikit pengertian tentang *framework*. *Framework* secara sederhana dapat diartikan kumpulan dari fungsi-fungsi/prosedur-prosedur dan *Class-Class* untuk tujuan tertentu yang sudah siap digunakan sehingga bisa lebih mempermudah dan mempercepat pekerjaan seorang pemrograman, tanpa harus membuat fungsi atau *Class* dari awal.

Ada beberapa alasan mengapa menggunakan *Framework*:

1. Mempercepat dan mempermudah pembangunan sebuah aplikasi *web*.
2. Relatif memudahkan dalam proses maintenance karena sudah ada pola tertentu dalam sebuah *framework* (dengan syarat programmer mengikuti pola standar yang ada)

3. Umumnya *framework* menyediakan fasilitas-fasilitas yang umum dipakai sehingga kita tidak perlu membangun dari awal (misalnya validasi, *pagination*, *multiple database*, scaffolding, pengaturan *session*, *error handling*, dll
4. Lebih bebas dalam pengembangan jika dibandingkan CMS.

Ada beberapa kelebihan CodeIgniter (CI) dibandingkan dengan *Framework* PHP lain :

1. Performa sangat cepat : salah satu alasan tidak menggunakan *framework* adalah karena eksekusinya yang lebih lambat daripada PHP from the scracth, tapi Codeigniter sangat cepat bahkan mungkin bisa dibilang codeigniter merupakan *framework* yang paling cepat dibanding *framework* yang lain.
2. Konfigurasi yang sangat minim (*nearly zero configuration*) : tentu saja untuk menyesuaikan dengan *database* dan keleluasaan routing tetap diizinkan melakukan konfigurasi dengan mengubah beberapa *file* konfigurasi seperti *database*.php atau autoload.php, namun untuk menggunakan codeigniter dengan setting standard, anda hanya perlu merubah sedikit saja *file* pada *folder* config.
3. Banyak komunitas: dengan banyaknya komunitas CI ini, memudahkan kita untuk berinteraksi dengan yang lain, baik itu bertanya atau teknologi terbaru.
4. Dokumentasi yang sangat lengkap : Setiap paket instalasi codeigniter sudah disertai *user guide* yang sangat bagus dan lengkap untuk dijadikan permulaan, bahasanya pun mudah dipahami.

3.6.5 Javascript

Javascript merupakan bahasa pemrograman berbasis *script*. *Javascript* memiliki kemampuan untuk menciptakan halaman web yang dinamis serta didukung oleh banyak Web Browser. Pada aplikasi *client-side*, *javascript* berjalan di dalam kode HTML (Irawan, 2008). Contoh penggunaan *javascript* dapat dilihat pada Gambar 3.4.

```
<!DOCTYPE
<html>
<body>
.
.
<script>
document.write("<h1>This is a heading</h1>");
document.write("<p>This is a paragraph</p>");
</script>
.
.
</body>
</html>
```

Gambar 3. 4 Contoh penggunaan javascript (www.w3schools.com)

3.6.6 Jquery

JQuery merupakan sebuah *Javacripts Library* atau biasa disebut juga sebagai pustaka dari kumpulan kode / *listing javacripts* yang siap digunakan. Dalam artian sederhana, *JQuery* dapat digunakan untuk meringkas sebuah *listing javascript* yang panjang dalam pengerjaan halaman web (Sanjaya, 2012). *JQuery* merupakan salah satu pustaka yang membuat program web di sisi *klien* tidak terlihat sebagai program *javascript* biasa, yang harus secara *eksplisit* disisipkan pada dokumen web. Pada teknik pemrograman sisi *klien* dengan menggunakan *javascript* biasa, setiap elemen yang akan memiliki *event*, secara *eksplisit* terlihat memiliki *event* yang dilekatkan pada elemen tersebut. Contoh kode *JQuery* dapat dilihat pada Gambar 3.5.

```
$(document).ready(function(){
  $("p").click(function(){
    $(this).hide();
  }); });
```

Gambar 3. 5 Contoh kode JQuery

Menurut Sanjaya (2012), beberapa keunggulan *JQuery* semenjak diluncurkan pada tahun 2006:

a. Mudah dioperasikan, karena hanya menyisipkan beberapa baris kode saja, sehingga tidak perlu menuliskan program yang panjang.

b. *JQuery* menyediakan fasilitas *plugin* yang beragam. Plugin-plugin tersebut sangat berguna dalam modifikasi ataupun penggunaan teknologi pada halaman website.
c. Menyediakan fasilitas untuk tampilan animasi dan tampilan yang interaktif yang menyerupai animasi pada *Flash*.
d. Komunitas pendukung yang tersebar diseluruh dunia dan selalu aktif melakukan inovasi-inovasi.
e. *JQuery* berbasis *open source*, sehingga bebas dan gratis digunakan dan dikembangkan oleh siapa saja

3.6.7 MySQL

Menurut Nugroho (2005), MySQL adalah sebuah *database* server, dapat juga berperan sebagai *client* sehingga sering disebut *database client/server*, yang *open source* dengan kemampuan dapat berjalan baik di OS(*Operating System*) manapun, dengan *platform* windows maupun linux. Selain itu *database* ini memiliki beberapa kelebihan dibanding *database* lain, diantaranya adalah:

a. MySQL sebagai *Database Management System* (DBMS).
b. MySQL sebagai *Relation Database Management System* (RDBMS).
c. MySQL adalah sebuah *software database* yang *open source*, artinya program ini bersifat gratis atau bebas digunakan oleh siapa saja tanpa harus membeli dan membayar lisensi kepada pembuatnya.
b. MySQL merupakan sebuah *database server*, jadi dengan menggunakan database ini anda dapat menghubungkan ke media internet sehingga dapat diakses dari jauh.
c. MySQL merupakan sebuah database *client*. Selain menjadi *server* yang melayani permintaan, MySQl juga melakukan *query* yang mengakses *database* pada *server*. Jadi MySQL dapat juga berperan sebagai client.
d. MySQL mampu menerima *query* yang bertumpuk dalam satu permintaan atau yang disebut *Multi-Threading*.

e. MySQL merupakan sebuah *database* yang mampu menyimpan data berkapasaitas sangat besar hingga berukuran *Gigabyte* sekalipun.
f. MySQl didukung oleh *driver* ODBC, artinya *database* MySQL dapat diakses menggunakan aplikasi apa saja termasuk berupa *visual* seperti Delphi maupuan VB.
g. MySQL adalah database menggunakan enkripsi *password*. Jadi *database* ini cukup aman karena *password* untuk mengaksesnya.
h. MySQL merupakan *server database* yang *multi user*, artinya *database* ini tidak hanya digunakan oleh sepihak orang akan tetapi merupakan *database* yang digunakan oleh banyak pengguna.
i. MySQL dapat menciptakan lebih dari 16 kunci per tabel, dan dalam satu kunci memungkinkan berisi belasan *field* (kolom).
j. MySQL mendukung kolom yang dijadikan sebagai kunci primer dan kunci unik.
k. MySQL didukung oleh sebuah komponen C dan perl API, sehingga MySQL dapat diakses melalui sebuah program aplikasi yang berbeda di bawah protocol internet berupa web. Biasanya aplikasi yang sering digunakan adalah PHP dan perl
l. MySQL memiliki kecepatan dalam pembuatan tabel maupun pengupdate-an tabel.
m. MySQL menggunakan suatu bahasa permintaan standar yang bernama SQL.

3.6 UML *(Unifed Modeling Language)*

Menurut Nugroho (2010:6), UML (*Unified Modeling Language*) adalah bahasa pemodelan untuk sistem atau perangkat lunak yang berparadigma berorientasi objek. Pemodelan *(modeling)* sesungguhnya digunakan untuk penyederhanaan permasalahan-permasalahan yang kompleks sedemikian rupa sehingga lebih mudah dipelajari dan dipahami.

UML berorientasi objek menerapkan banyak level abstraksi, tidak bergantung proses pengembangan, tidak bergantung bahasa dan teknologi. Pemaduan beberapa notasi di beragam metodologi usaha bersama dari banyak

pihak, di dukung oleh kakas- kakas yang di integrasikan lewat XML.

3.6.1 *Use Case* Diagram

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan dalam sistem,dan bukan adalah “a “bagaimana”. *Use case* merepresentasikan.*Use case* menggambarkan pekerjaan tertentu, misalnya login ke sistem, membuat daftar belanja, dan sebagainya.

Aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu. *Use case* diagram dapat sangat membantu ketika sedang menyusun analisis kebutuhan sebuah sistem, mengkomunikasikan rancangan dengan client, dan merancang test case untuk semua fitur yang ada pada sistem.

Use case dapat memasukkan (include) fungsionalitas *use case* lain sebagai bagian dari proses dalam dirinya. Secara umum diasumsikan bahwa *use case* yang di-include akan dipanggil setiap kali *use case* yang meng-include dieksekusi secara normal. 2Sebuah *use case* dapat di-include oleh lebih dari satu *use case* lain, sehingga duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsionalitas yang sejenis.

Use case juga dapat meng-extend *use case* lain dengan behaviour-nya sendiri. Sementara hubungan generalisasi antar *use case* menunjukkan bahwa *use case* yang satu merupakan spesialisasi dari yang lain. Terdapat beberapa simbol yang digunakan dalam pembuatan sebuah *Use case* diagram ditunjukan pada Tabel 3.2.

Tabel 3. 2 Simbol *Use Case* Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		*Actor*	Menspesifikasikan himpuan peran yang pengguna mainkan ketika berinteraksi dengan *use case*.
2		*Include*	Menspesifikasikan bahwa *use case* sumber secara *eksplisit*.
3		*Association*	Apa yang menghubungkan antara objek satu dengan objek lainnya.
4		*System*	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
5		*Use Case*	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor

3.6.2 *Activity* Diagram

Activity diagram menggambarkan berbagai alir kegiatan dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, decision yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity* diagram juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.

Activity diagram merupakan state diagram khusus, di mana sebagian besar state adalah action dan sebagian besar transisi di-trigger oleh selesainya state sebelumnya *(internal processing).* Oleh karena itu *activity* diagram tidak menggambarkan *behavior internal* sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur kegiatan dari level atas secara umum.

Sebuah kegiatan dapat direalisasikan oleh satu *use case* atau lebih. Kegiatan menggambarkan proses yang berjalan, sementara *use case* menggambarkan bagaimana aktor menggunakan sistem untuk melakukan kegiatan. Sama seperti state, standar UML menggunakan segiempat dengan sudut membulat untuk menggambarkan kegiatan. *Decision* digunakan untuk

menggambarkan *behaviour* pada kondisi tertentu. Untuk mengilustrasikan proses-proses paralel *(fork* dan *join)* digunakan titik sinkronisasi yang dapat berupa titik, garis horizontal atau vertikal.

Activity diagram dapat dibagi menjadi beberapa objek *swimlane* untuk menggambarkan objek mana yang bertanggung jawab untuk kegiatan tertentu. Terdapat beberapa simbol yang digunakan dalam pembuatan sebuah *activity* diagram ditunjukan pada Tabel 3.3.

Tabel 3. 3 Simbol *Activity* Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		*Activity*	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		*Action*	State dari sistem yang mencerminkan eksekusi dari suatu aksi
3		*Start State*	Bagaimana objek dibentuk atau diawali.
4		*End State*	Bagaimana objek dibentuk dan dihancurkan
5		*State Transitio*	State transition menunjukkan kegiatan apa berikutnya setelah suatu kegiatan
6		*Fork*	Percabangan yang menunjukkan aliran pada *Activity Diagram*
7		*Join*	Penggabungan yang menjadi arah aliran pada *Activity Diagram*
8			Pilihan untuk mengambil keputusan
9		*Flow Final*	Aliran akhir

3.6.3 *Class* Diagram

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi). *Class* diagram menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain. Terdapat beberapa simbol yang digunakan dalam pembuatan sebuah

class diagram ditunjukan pada Tabel 3.4.

Tabel 3. 4 Simbol *Class* Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		*Generalization*	Hubungan dimana objek anak *(descendent)* berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*).
2		*Class*	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.

3.7 Profil Umum

Ibie Laundry & Wet Cleaning mempunyai visi misi dan motto dalam proses bisnisnya

Visi

1. Menjadi salah satu perusahaan laundry yang ramah lingkungan, sehat, dan bersih.
2. Menjadi pelopor perusahaan laundry dengan konsep open kitchen.
3. Menjadi perusahaan laundry pilihan utama bagi konsumen dan calon konsumen.

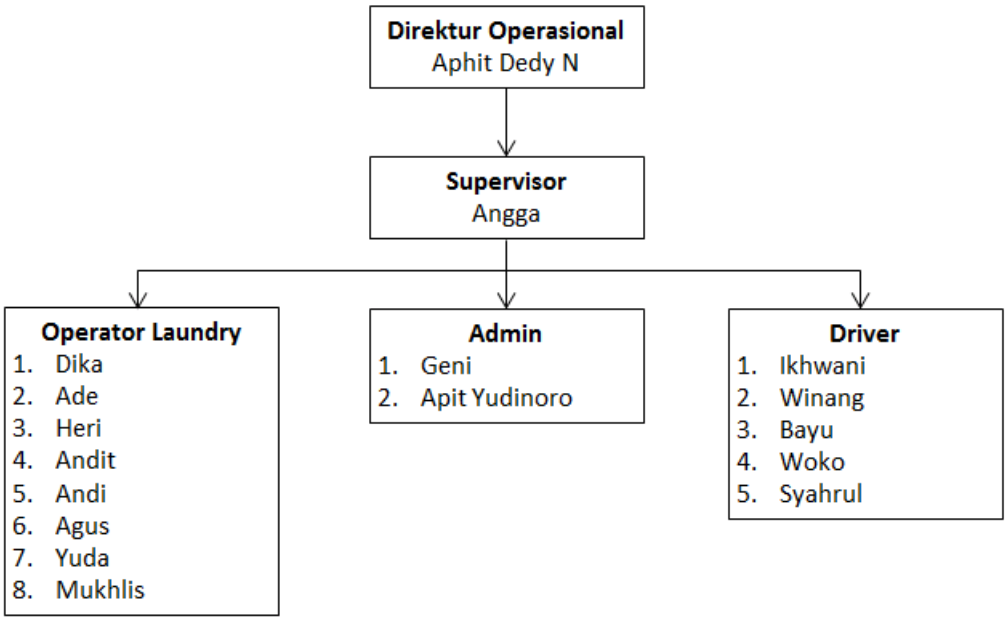
Misi

1. Memberikan pelayanan terbaik untuk kepuasan konsumen.
2. Pengelolaan manajemen secara profesional untuk meningkatkan kualitas usaha.
3. Mengembangkan bisnis untuk keuntungan pelanggan, karyawan dan pemilik.
4. Mendirikan, menjalankan dan mengembangkan perusahaan laundry yang ramah lingkungan di seluruh Indonesia.

Motto

Health and Clean

Struktur Organisasi



Gambar 3. 6 Stuktur organisasi

BAB IV
ANALISI DAN PERANCANGAN SISTEM

4.1 Analisis Sistem

Analisis sistem adalah mendefinisikan dan menspesifikasi dengan *detail* apa yang harus dilakukan oleh sistem. Tahapan ini merupakan tahapan yang penting karena sebagai acuan dalam tahapan selanjutnya.

4.1.1 Analisis sistem yang ada

Ibie Laundry & Wet Cleaning merupakan salah satu laundry di Kabupaten Sleman, Yogyakarta yang berdiri sejak tahun 2015. Saat ini *Ibie Laundry & Wet Cleaning* sudah menggunakan suatu sistem dalam pencatatan data transaksi, tetapi sistem tersebut masih sering terjadi banyak kesalahan sehingga masih harus dilakukan pengecekan secara manual. Dengan pengelolaan data yang menggunakan sistem lama, proses pengerjaannya terhitung harus menggunakan waktu yang cukup lama.

Untuk pengelolaan data *omzet dan profit* sudah menggunakan sistem dalam bentuk *excel*. Dalam pengelolaan data stok masih menggunakan manual juga, sehingga bisa terjadi redundansi data. Pengunaan sistem dalam bentuk *excel* sudah cukup membantu kerja dari seorang *manager*, akan tetapi masih kurang efektif untuk melakukan pencarian dan pelaporan.

4.1.2 Analisis pengguna sistem

Dalam perancangan sistem ini, pengguna yang akan memanfaatkan sistem ini terbagi menjadi dua jenis pengguna, yaitu:

1. *Admin*

   Entitas *admin* sebagai *admin*istrator merupakan entitas yang dapat melakukan pengolahan semua data pada sistem laundry. Data tersebut terdiri dari data transaksi, barang, perlengkapan, dan *customer*.

2. *Manager*

Entitas *manager* merupakan entitas yang hanya dapat melihat semua laporan yang sudah direkap oleh *administrator* dalam pengolahan semua data pada sistem. Laporan tersebut berupa laporan *omzet, profit,* transaksi, barang, perlengkapan, dan *customer*.

4.1.3 Analisis kebutuhan sistem

Berdasarkan penjabaran permasalahan yang telah dijelaskan, maka sistem memiliki kebutuhan fungsional sebagai berikut:

1. Sistem menangani validasi login, digunakan untuk masuk ke sistem sebagai *admin*, atau *manager*.
2. *Admin* dapat melakukan proses transaksi.
3. *Admin* dapat mencatat data barang.
4. *Admin* dapat mencatat data *customer*.
5. *Admin* dapat mencatat data perlengkapan.
6. *Admin* dapat melihat laporan data barang.
7. *Admin* dapat melihat laporan data *customer*.
8. *Admin* dapat melihat laporan data perlengkapan.
9. *Admin* dapat mengirim data transaksi kepada *customer* melalui *email*.
10. *Manager* dapat mencatat data perlengkapan.
11. *Manager* dapat melihat laporan transaksi.
12. *Manager* dapat melihat laporan data barang.
13. *Manager* dapat melihat laporan data *customer*.
14. *Manager* dapat melihat laporan data perlengkapan.
15. *Manager* dapat melihat laporan *profit*.
16. *Manager* dapat melihat laporan *omzet*.

Selain kebutuhan akan fungsionalitas yang telah disebutkan diatas, sistem ini juga memiliki kebutuhan non fungsional sebagai berikut:

1. Sistem dijalankan dengan *web browser*.
2. Sistem membutuhkan koneksi internet.
3. Hak akses sistem hanya diberikan kepada *admin* dan *manager*.
4. File unduhan berupa file .xps.

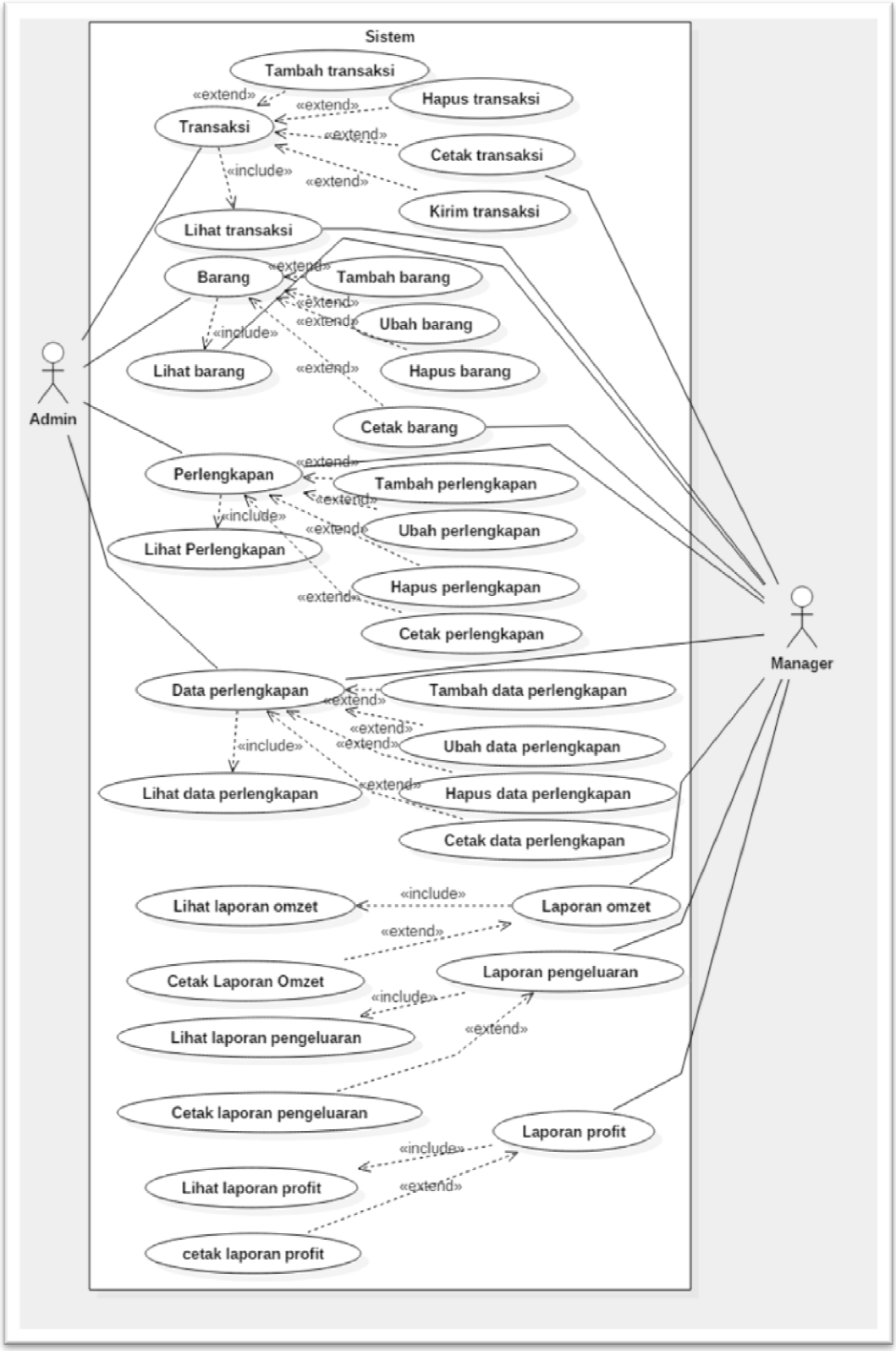
4.2 Perancangan Sistem

4.2.1 Perancangan Proses

Perancangan proses menjelaskan gambaran proses-proses atau alur berjalan pada Sistem Informasi Pengolahan Transaksi Laundry yang akan dibangun sehingga tercipta suatu informasi yang benar. Perancangan antarmuka proses dalam pembangunan Sistem Informasi Pengolahan Transaksi Laundry dibuat menggunakan *Unified Modelling Language* (UML) yang dapat digambarkan dengan *use case* diagram, *activity* diagram, dan *class* diagram.

***Use case* diagram**

Use case diagram digunakan untuk menggambarkan interaksi sistem, sistem *eksternal*, dan pengguna. Berdasarkan hasil analisis kebutuhan *user, use case* diagram pada Sistem Informasi Pengolahan Transaksi Laundry yang akan dibangun terdapat dua aktor, yaitu *admin,* dan *manager*. *Use case* diagram Sistem Pengolahan Transaksi Laundry dapat dilihat pada Gambar 4.1

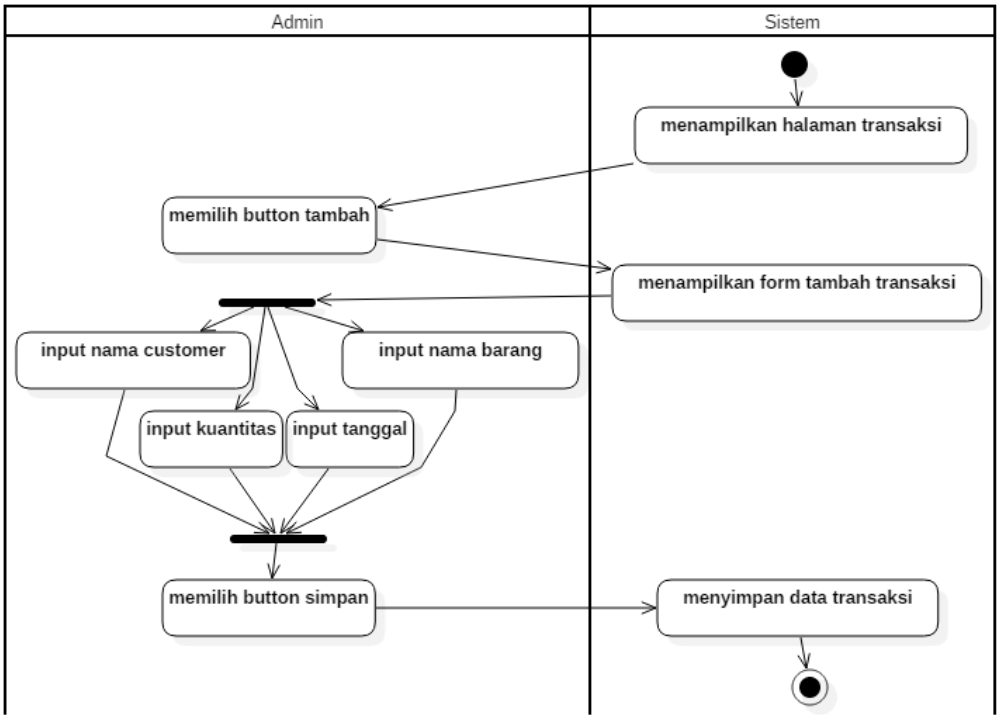


Gambar 4. 1 *Use case* diagram

***Activity* Diagram**

1. *Activity* diagram *admin* tambah transaksi

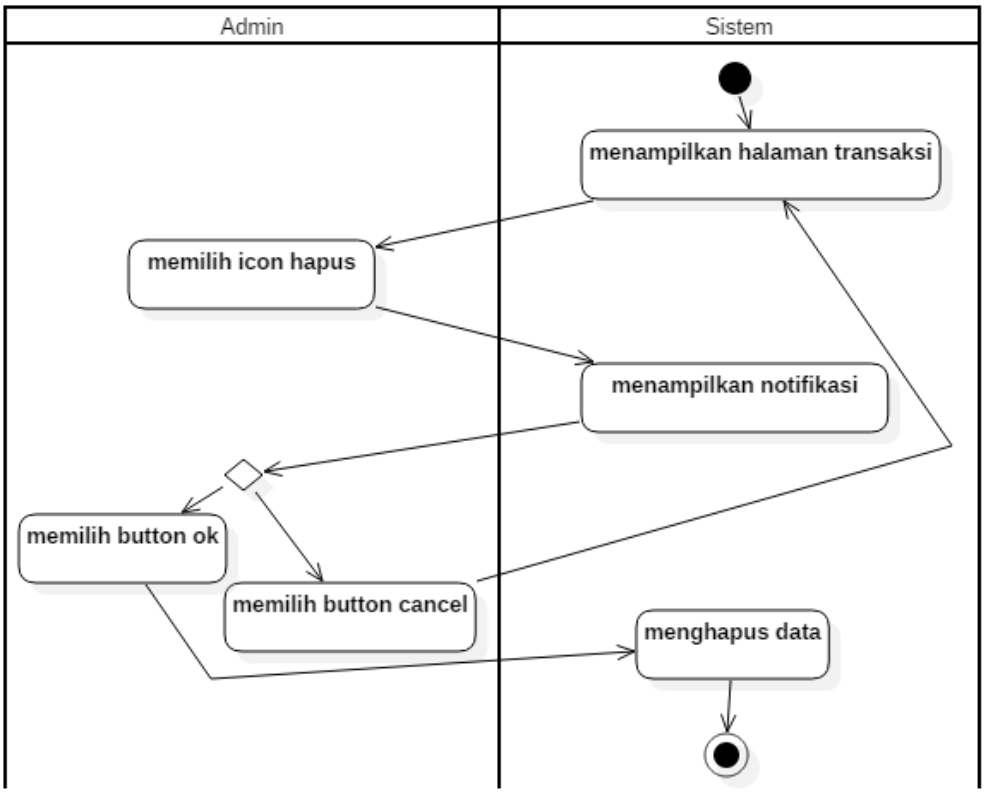
Dalam *activity* diagram *admin* dapat menambah transaksi meliputi nama *customer*, nama barang, kuantitas dan tanggal masuk. Tampilan *activity* diagram tambah transaksi dapat dilihat pada Gambar 4.2



Gambar 4. 2 *Activity* diagram *admin* tambah transaksi

2. *Activity* diagram *admin* hapus transaksi

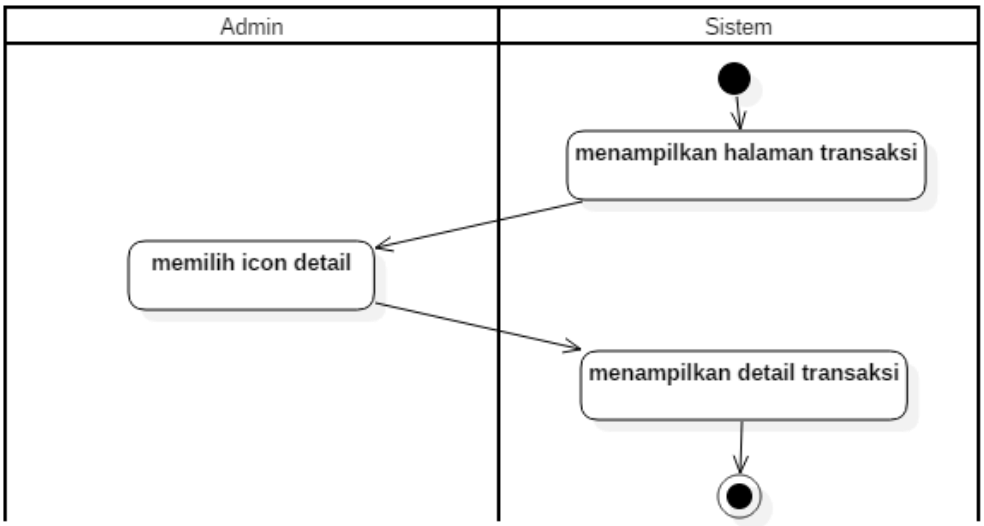
Dalam *activity* diagram *admin* dapat menghapus data transaksi. Tampilan *activity* diagram hapus transaksi dapat dilihat pada Gambar 4.3



Gambar 4. 3 *Activity* diagram *admin* hapus transaksi

3. *Activity* diagram *admin* lihat *detail* transaksi

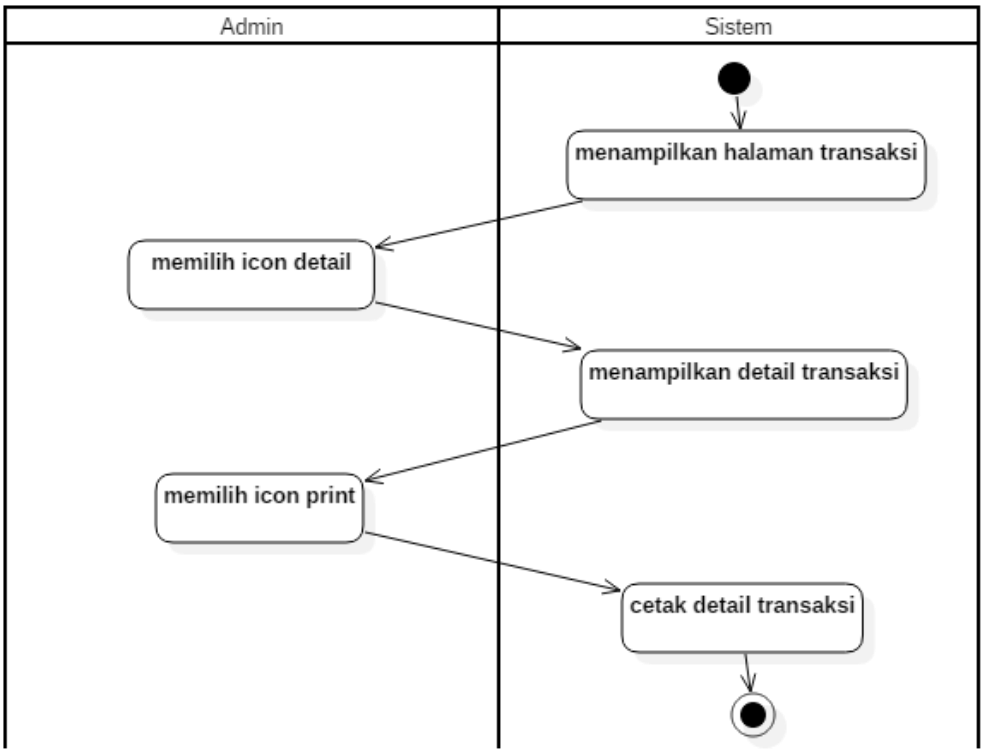
Dalam *activity* diagram *admin* dapat melihat *detail* data transaksi meliputi id transaksi, nama *customer*, lunas, status, total, tanggal masuk, tanggal kembali, nama barang, kuantitas, harga barang dan sub total. Tampilan *activity* diagram lihat *detail* transaksi dapat dilihat pada Gambar 4.4



Gambar 4. 4 *Activity* diagram *admin* lihat *detail* transaksi

4. *Activity* diagram *admin* cetak *detail* transaksi

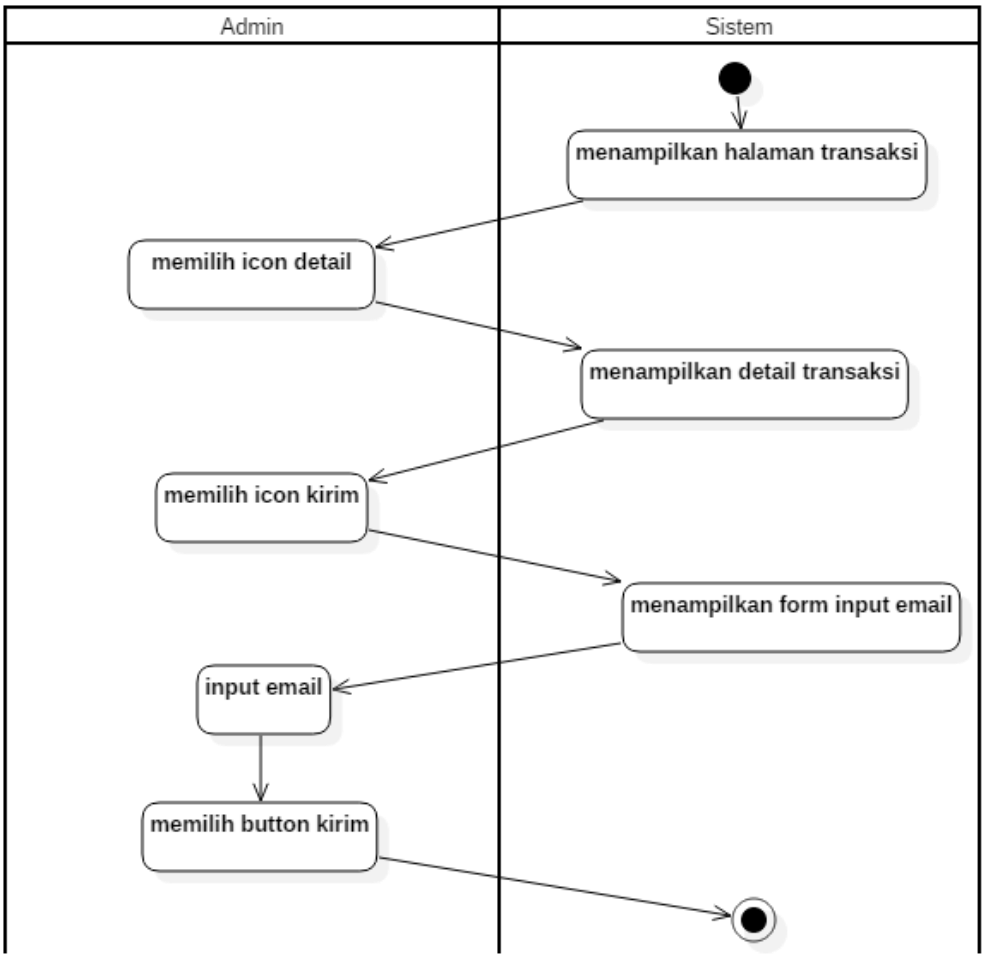
Dalam *activity* diagram *admin* dapat mencetak *detail* data transaksi meliputi id transaksi, nama *customer*, lunas, status, total, tanggal masuk, tanggal kembali, nama barang, kuantitas, harga barang dan sub total. Tampilan *activity* diagram cetak *detail* transaksi dapat dilihat pada Gambar 4.5



Gambar 4. 5 *Activity* diagram *admin* lihat *detail* transaksi

5. *Activity* diagram *admin* kirim *detail* transaksi

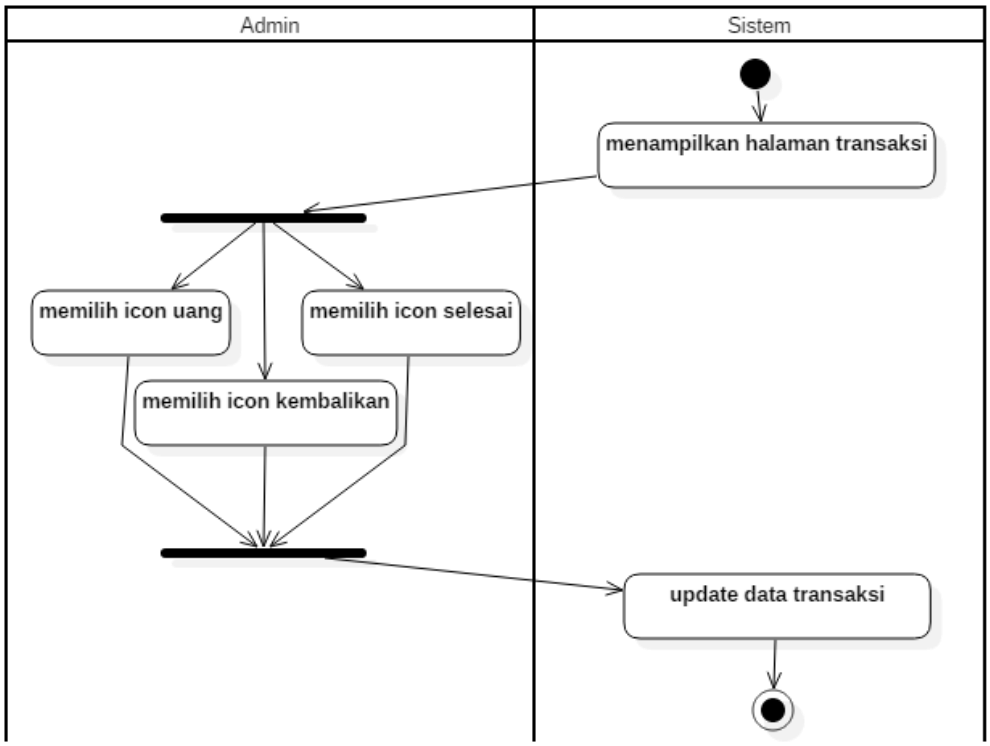
Dalam *activity* diagram *admin* dapat mengirim *detail* data transaksi meliputi id transaksi, nama *customer*, lunas, status, total, tanggal masuk, tanggal kembali, nama barang, kuantitas, harga barang dan sub total. Tampilan *activity* diagram kirim *detail* transaksi dapat dilihat pada Gambar 4.6



Gambar 4. 6 *Activity* diagram *admin* kirim *detail* transaksi

6. *Activity* diagram *admin* update transaksi

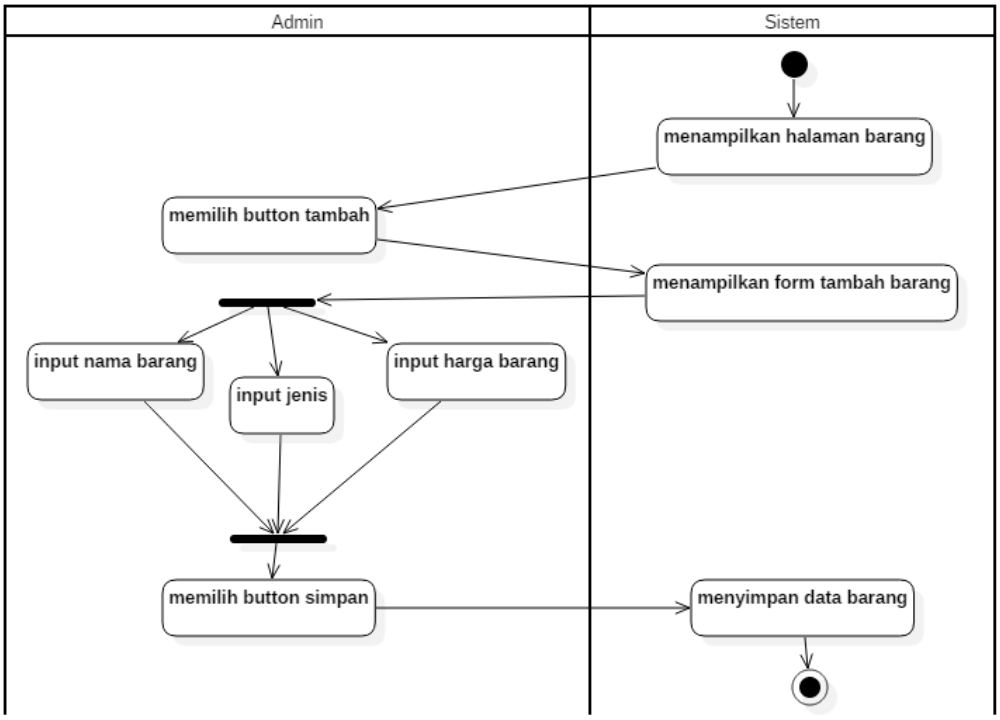
Dalam *activity* diagram *admin* dapat meng*update* data transaksi meliputi lunas, status, dan tanggal kembali. Tampilan *activity* diagram update transaksi dapat dilihat pada Gambar 4.7



Gambar 4. 7 *Activity* diagram *admin* update transaksi

7. *Activity* diagram *admin* tambah barang

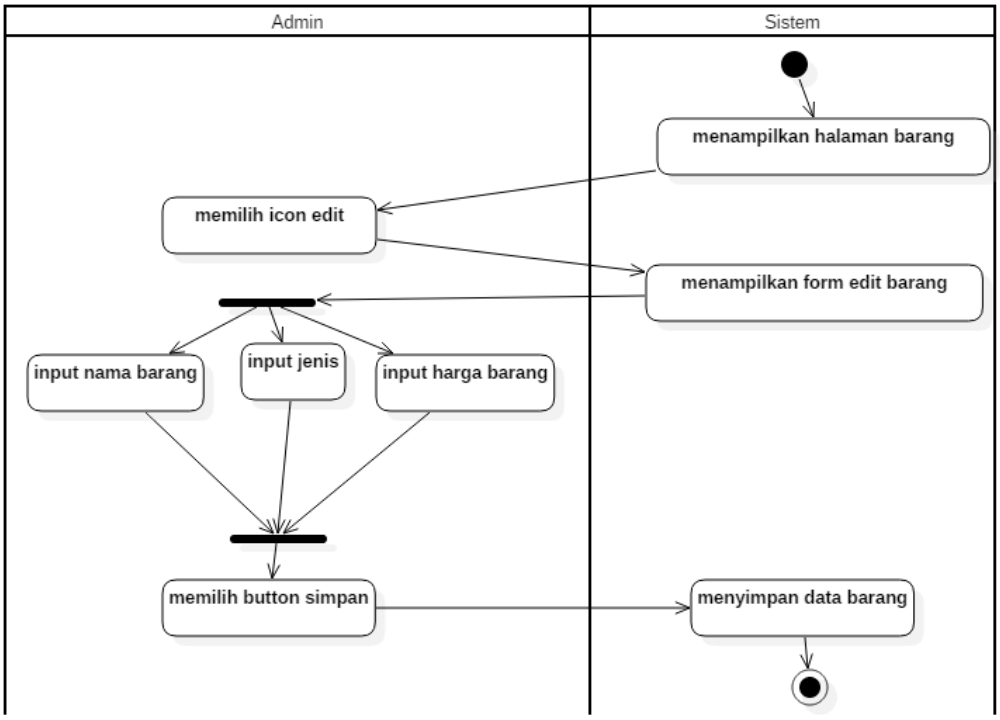
Dalam *activity* diagram *admin* dapat menambah data barang meliputi nama barang, jenis dan harga barang. Tampilan *activity* diagram tambah barang dapat dilihat pada Gambar 4.8



Gambar 4. 8 *Activity* diagram *admin* tambah barang

8. Activity diagram *admin edit* barang

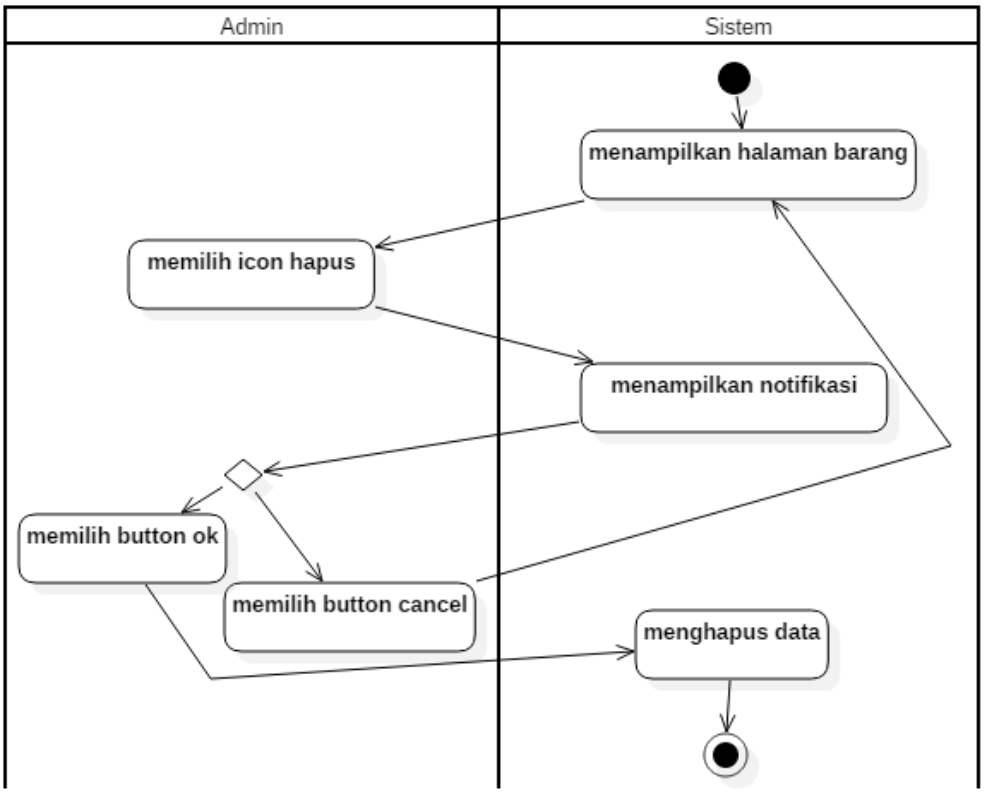
Dalam *activity* diagram *admin* dapat mengubah data barang meliputi nama barang, jenis dan harga barang. Tampilan *activity* diagram *edit* barang dapat dilihat pada Gambar 4.9



Gambar 4. 9 *Activity* diagram *admin* edit barang

9. *Activity* diagram *admin* hapus barang

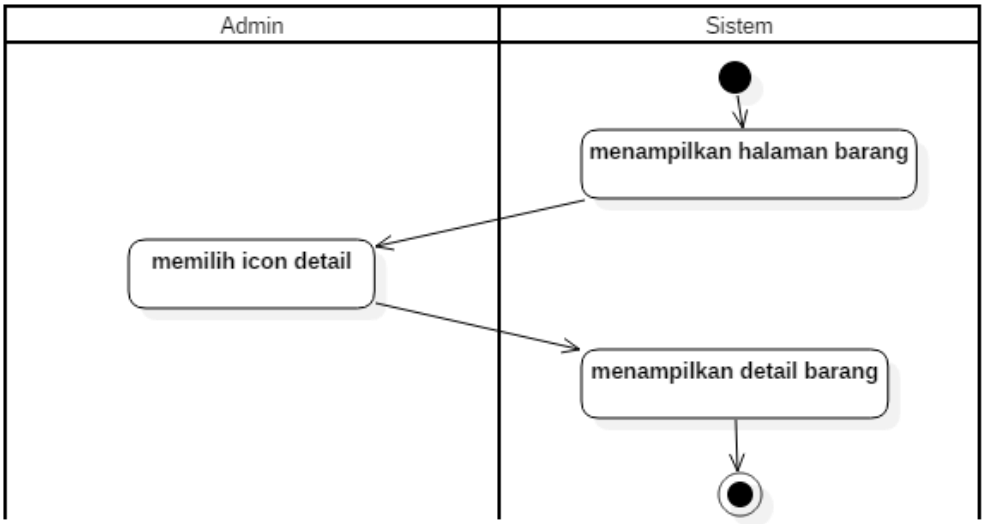
Dalam *activity* diagram *admin* dapat menghapus data barang. Tampilan *activity* diagram hapus barang dapat dilihat pada Gambar 4.10



Gambar 4. 10 *Activity* diagram *admin* hapus barang

10. *Activity* diagram *admin* lihat *detail* barang

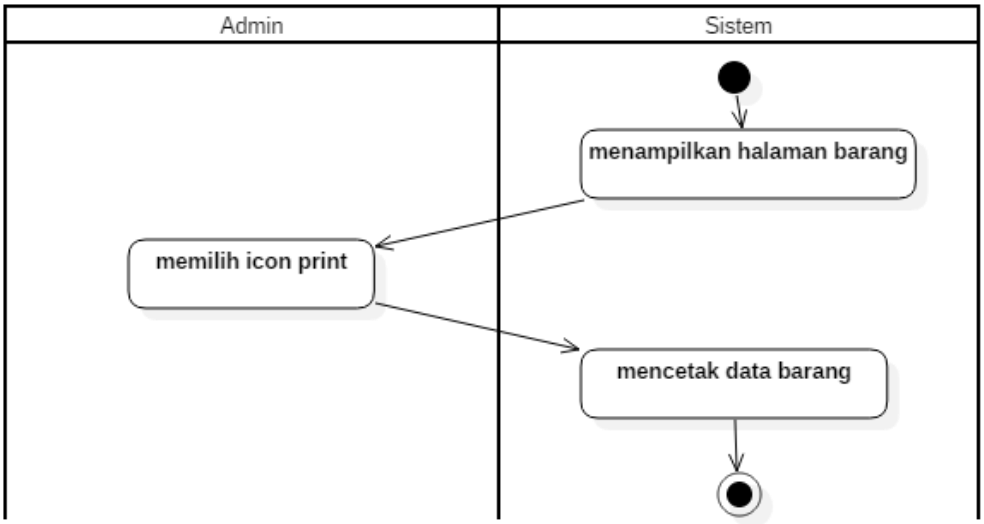
Dalam *activity* diagram *admin* dapat melihat *detail* data barang meliputi id barang, nama barang, dan harga barang. Tampilan *activity* diagram lihat *detail* barang dapat dilihat pada Gambar 4.11



Gambar 4. 11 *Activity* diagram *admin* lihat *detail* barang

11. *Activity* diagram *admin* cetak data barang

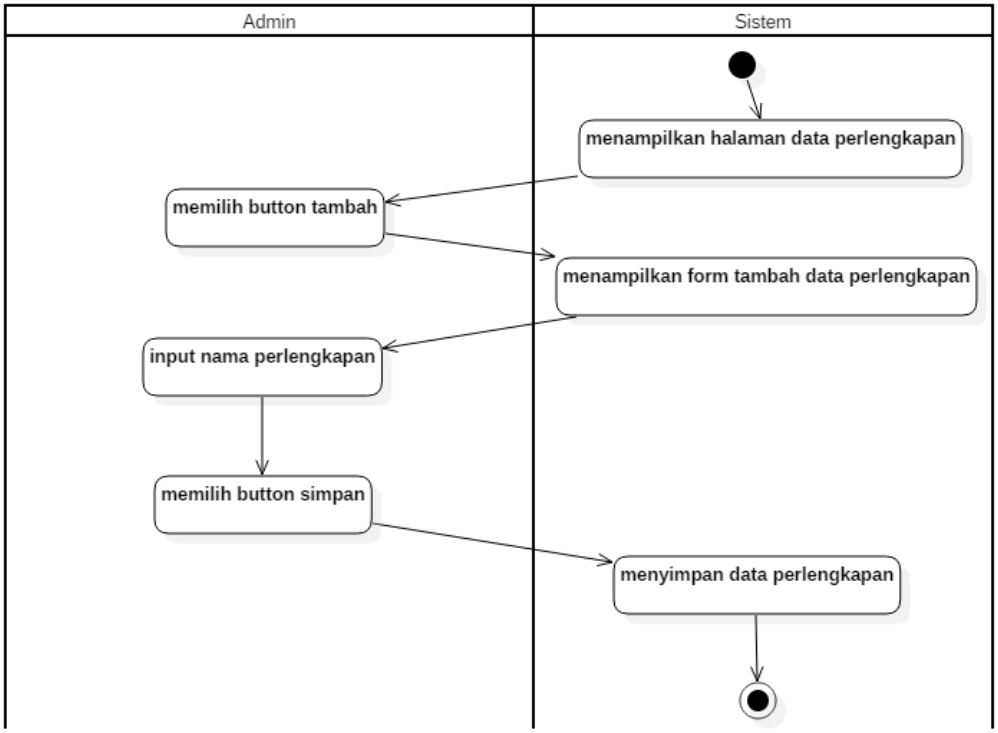
Dalam *activity* diagram *admin* dapat mencetak data barang meliputi id barang, nama barang, dan harga barang. Tampilan *activity* diagram cetak data barang dapat dilihat pada Gambar 4.12



Gambar 4. 12 *Activity* diagram *admin* cetak data barang

12. *Activity* diagram *admin* tambah data perlengkapan

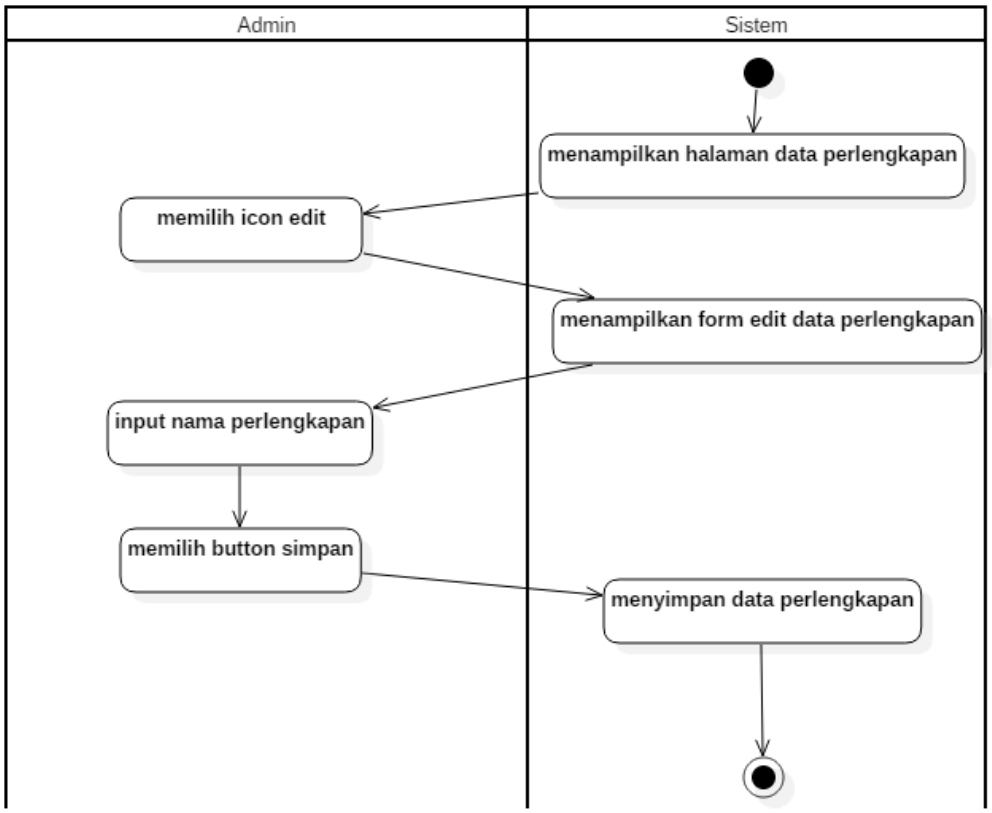
Dalam *activity* diagram *admin* dapat menambah data perlengkapan meliputi nama perlengkapan. Tampilan *activity* diagram tambah data perlengkapan dapat dilihat pada Gambar 4.13



Gambar 4. 13 *Activity* diagram *admin* tambah data perlengkapan

13. *Activity* diagram *admin* edit data perlengkapan

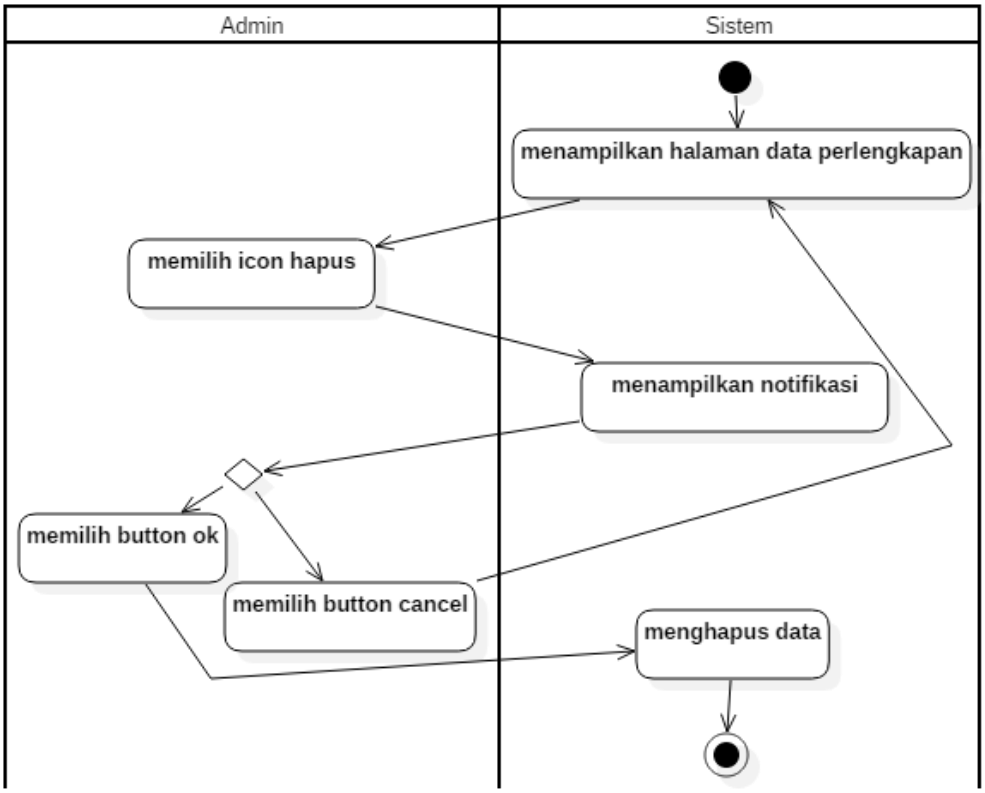
Dalam *activity* diagram *admin* dapat mengubah data perlengkapan meliputi nama perlengkapan. Tampilan *activity* diagram *edit* data perlengkapan dapat dilihat pada Gambar 4.14



Gambar 4. 14 *Activity* diagram *admin* edit data perlengkapan

14. *Activity* diagram *admin* hapus data perlengkapan

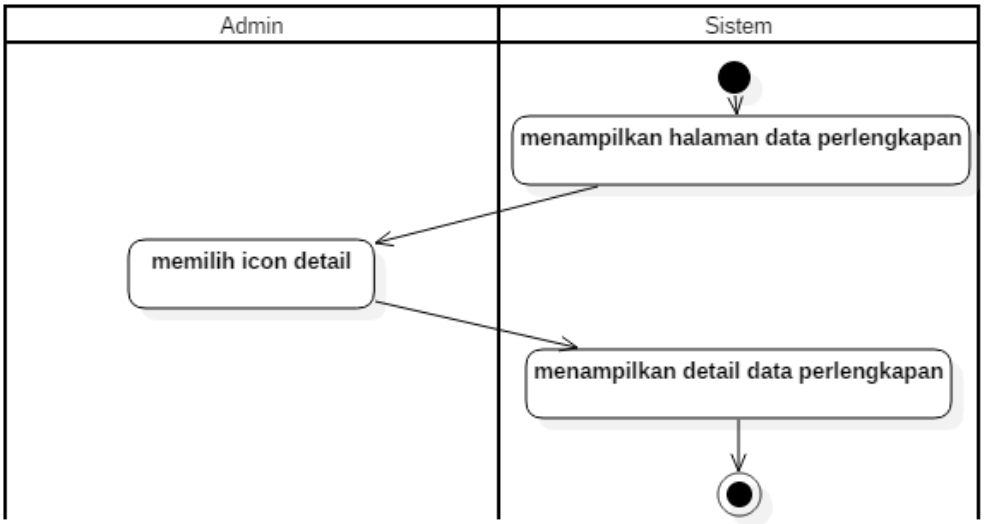
Dalam *activity* diagram *admin* dapat menghapus data perlengkapan. Tampilan *activity* diagram hapus data perlengkapan dapat dilihat pada Gambar 4.15



Gambar 4. 15 *Activity* diagram *admin* hapus data perlengkapan

15. *Activity* diagram *admin* lihat *detail* data perlengkapan

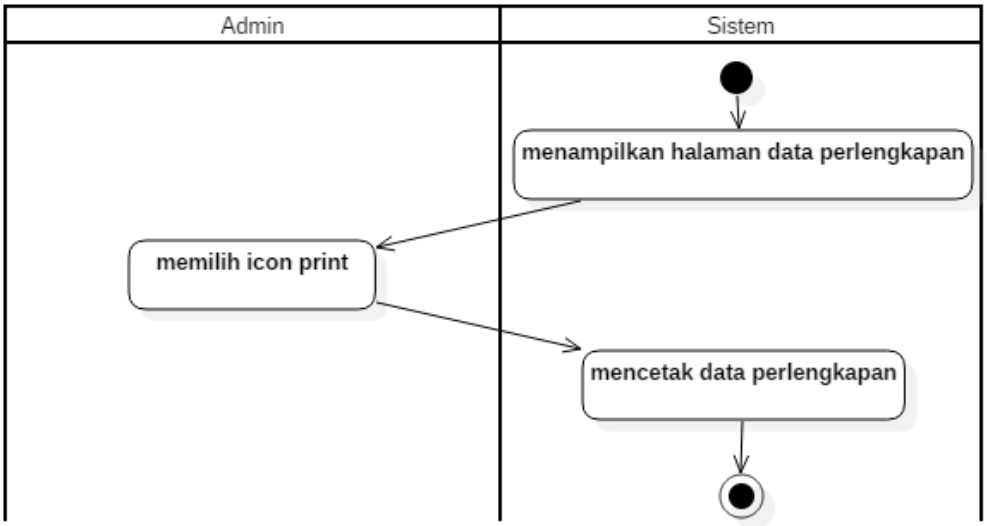
Dalam *activity* diagram *admin* dapat melihat *detail* data perlengkapan meliputi id perlengkapan dan nama perlengkapan. Tampilan *activity* diagram lihat *detail* data perlengkapan dapat dilihat pada Gambar 4.16



Gambar 4. 16 *Activity* diagram *admin* lihat *detail* data perlengkapan

16. *Activity* diagram *admin* cetak data perlengkapan

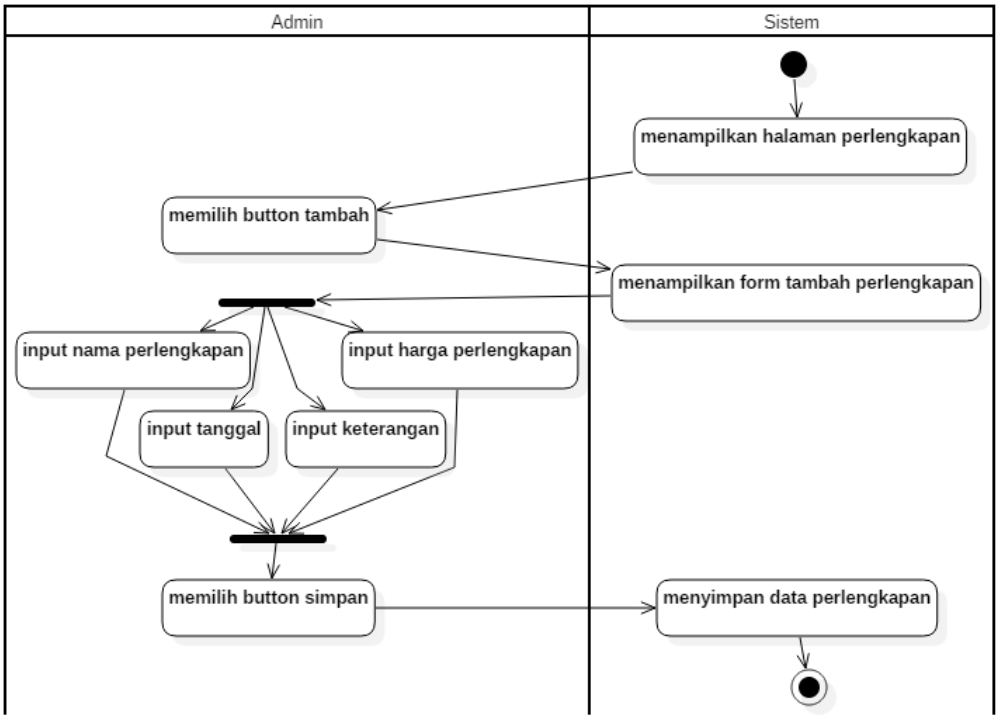
Dalam *activity* diagram *admin* dapat mencetak data perlengkapan meliputi id perlengkapan dan nama perlengkapan. Tampilan *activity* diagram cetak data perlengkapan dapat dilihat pada Gambar 4.17



Gambar 4. 17 *Activity* diagram *admin* cetak data perlengkapan

17. *Activity* diagram *admin* tambah perlengkapan

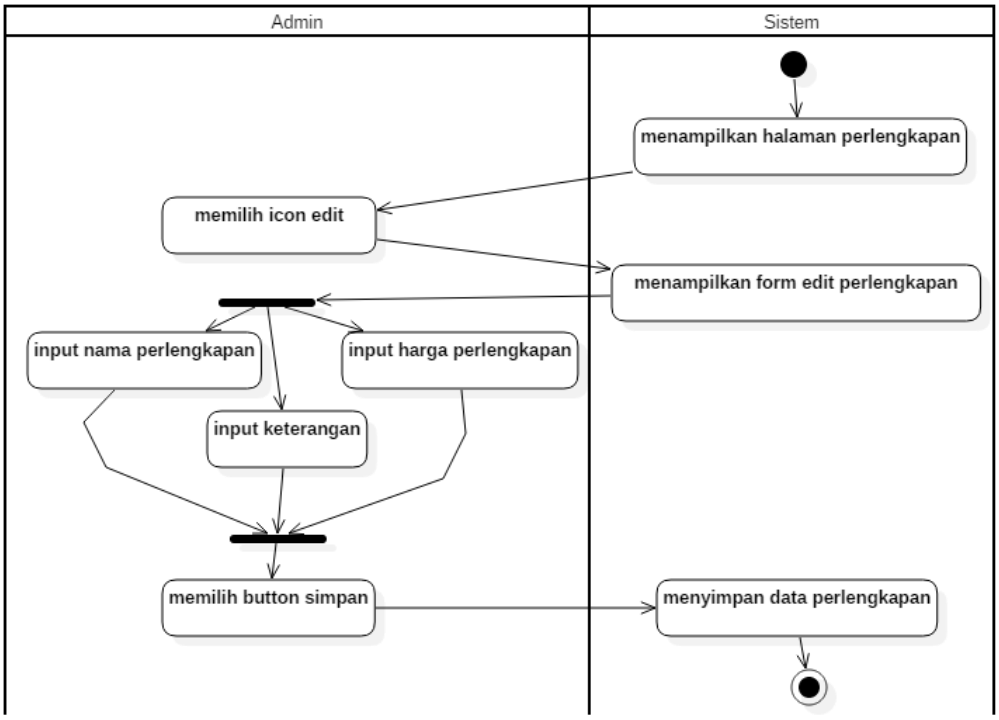
Dalam *activity* diagram *admin* dapat menambah perlengkapan meliputi tanggal, nama perlengkapan, harga dan keterangan. Tampilan *activity* diagram tambah perlengkapan dapat dilihat pada Gambar 4.18



Gambar 4. 18 *Activity* diagram *admin* tambah perlengkapan

18. *Activity* diagram *admin* edit perlengkapan

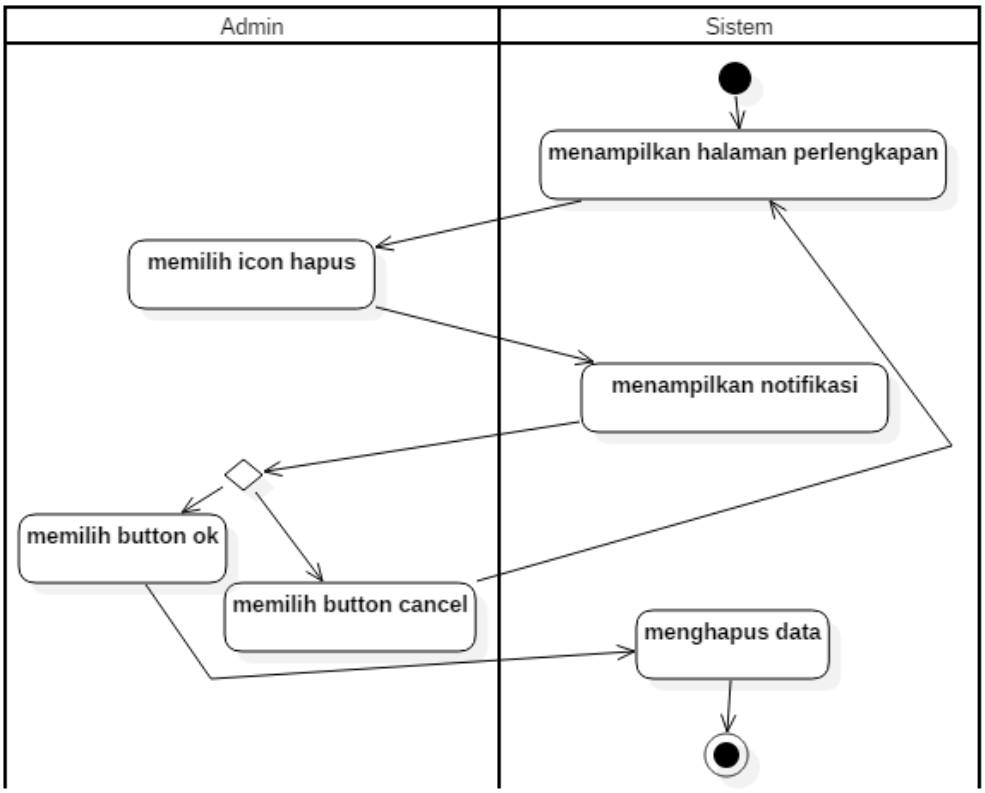
Dalam *activity* diagram *admin* dapat mengubah perlengkapan meliputi nama perlengkapan, harga dan keterangan. Tampilan *activity* diagram *edit* perlengkapan dapat dilihat pada Gambar 4.19



Gambar 4. 19 *Activity* diagram *admin* edit perlengkapan

19. *Activity* diagram *admin* hapus perlengkapan

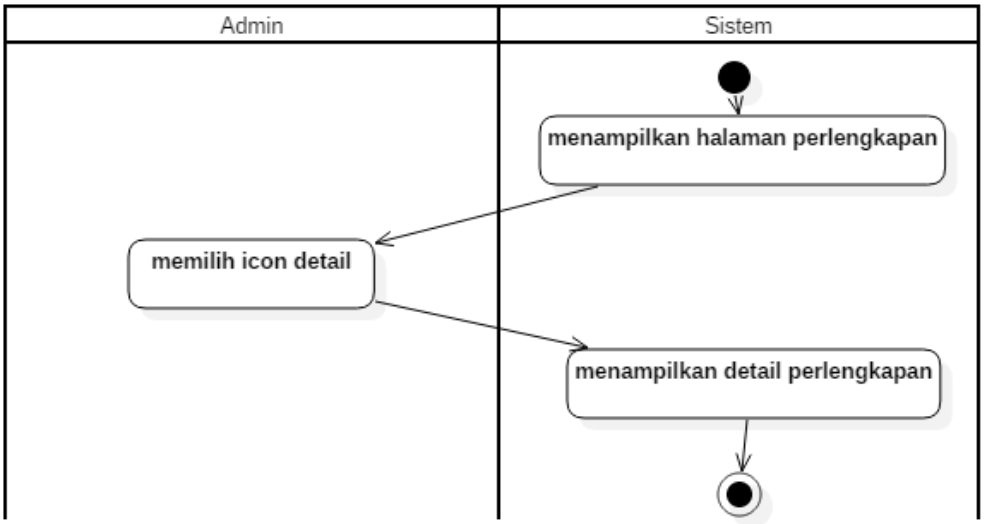
Dalam *activity* diagram *admin* dapat menghapus perlengkapan. Tampilan *activity* diagram hapus perlengkapan dapat dilihat pada Gambar 4.20



Gambar 4. 20 *Activity* diagram *admin* hapus perlengkapan

20. *Activity* diagram *admin* lihat *detail* perlengkapan

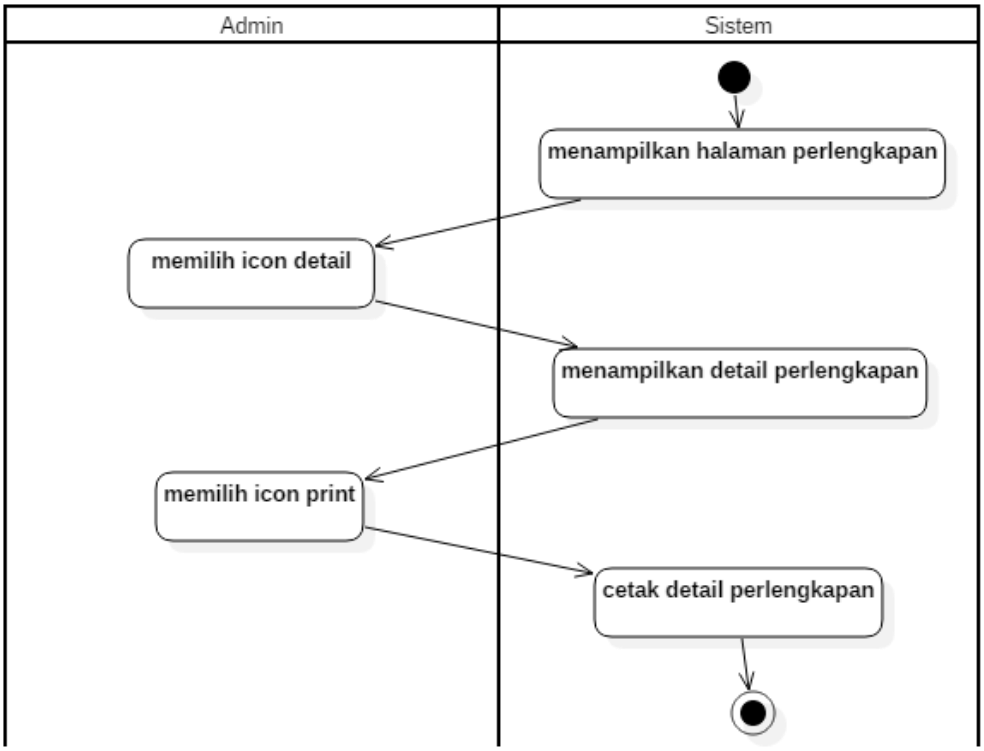
Dalam *activity* diagram *admin* dapat melihat *detail* perlengkapan meliputi id perlengkapan, tanggal, nama perlengkapan, harga dan keterangan. Tampilan *activity* diagram lihat *detail* perlengkapan dapat dilihat pada Gambar 4.21



Gambar 4. 21 *Activity* diagram *admin* lihat *detail* perlengkapan

21. *Activity* diagram *admin* cetak perlengkapan

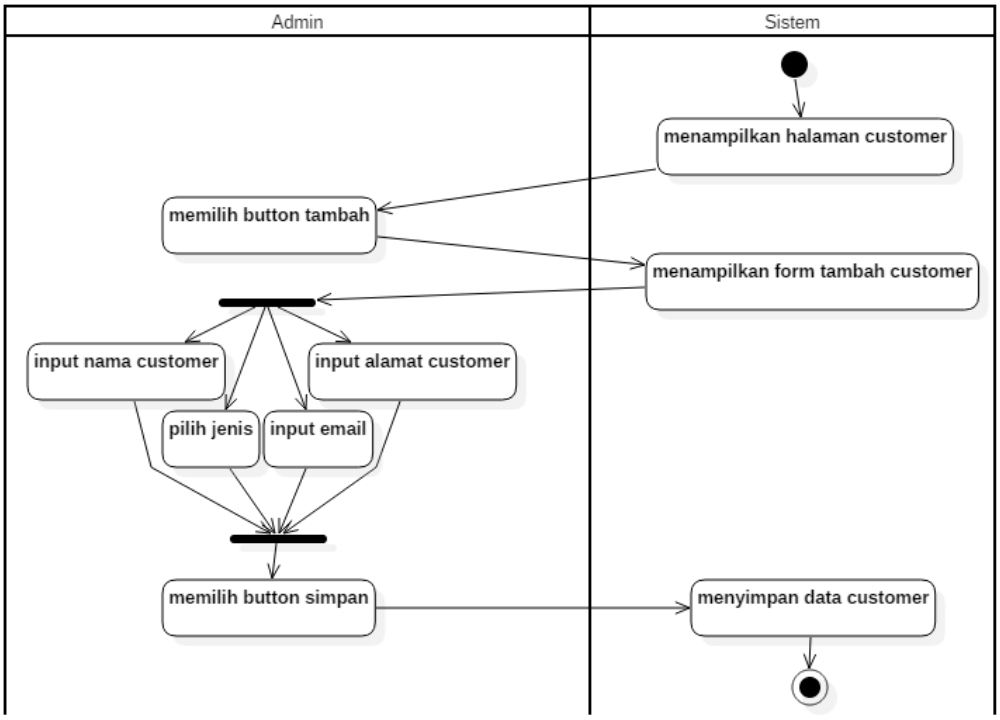
Dalam *activity* diagram *admin* dapat mencetak perlengkapan meliputi id perlengkapan, tanggal, nama perlengkapan, harga dan keterangan. Tampilan *activity* diagram lihat *detail* perlengkapan dapat dilihat pada Gambar 4.22



Gambar 4. 22 *Activity* diagram *admin* cetak perlengkapan

22. *Activity* diagram *admin* tambah *customer*

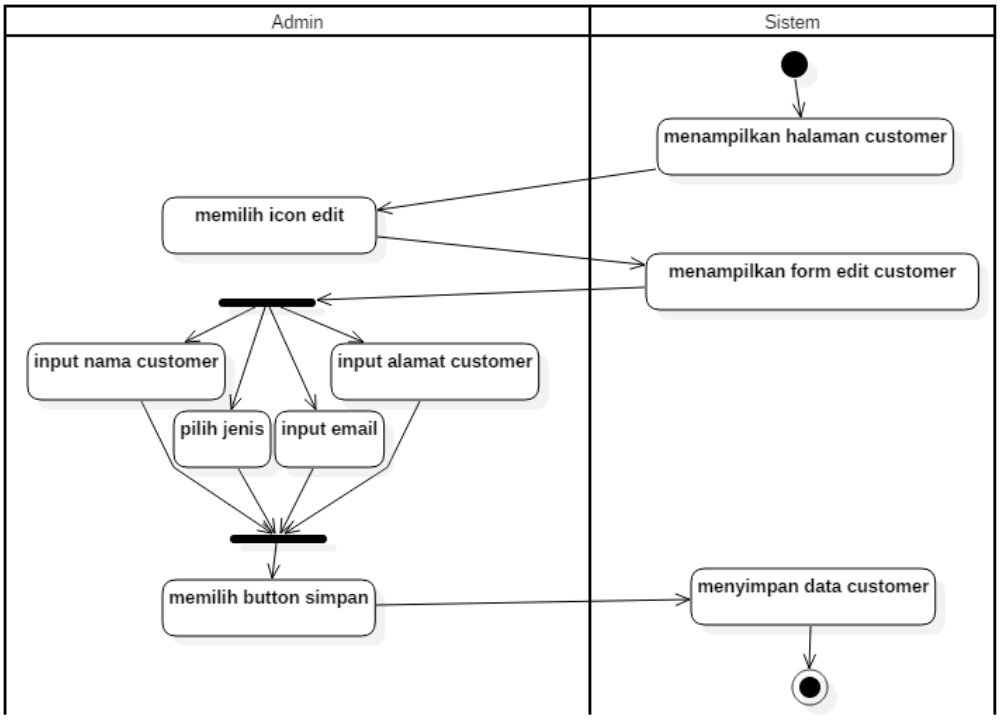
Dalam *activity* diagram *admin* dapat menambah data *customer* meliputi id *customer*, nama *customer*, alamat *customer*, *email* dan jenis. Tampilan *activity* diagram tambah *customer* dapat dilihat pada Gambar 4.23



Gambar 4. 23 *Activity* diagram *admin* tambah *customer*

23. *Activity* diagram *admin edit customer*

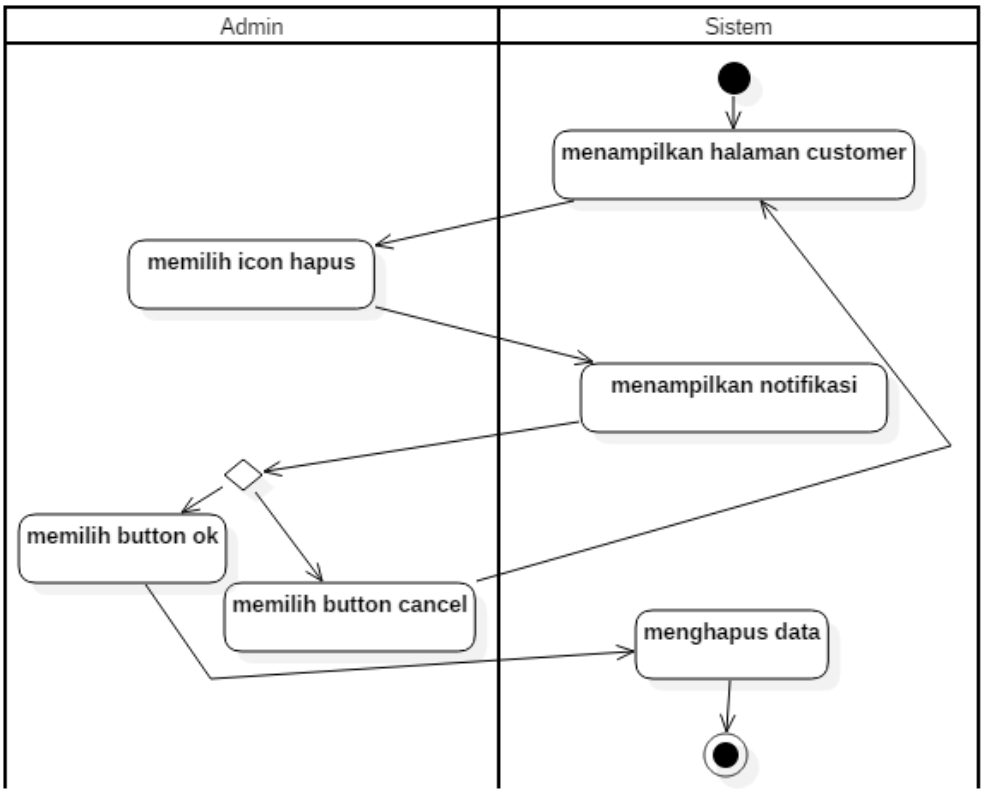
Dalam *activity* diagram *admin* dapat mengubah data *customer* meliputi nama *customer*, alamat *customer*, *email* dan jenis. Tampilan *activity* diagram *edit customer* dapat dilihat pada Gambar 4.24



Gambar 4. 24 *Activity* diagram *admin* edit *customer*

24. *Activity* diagram *admin* hapus *customer*

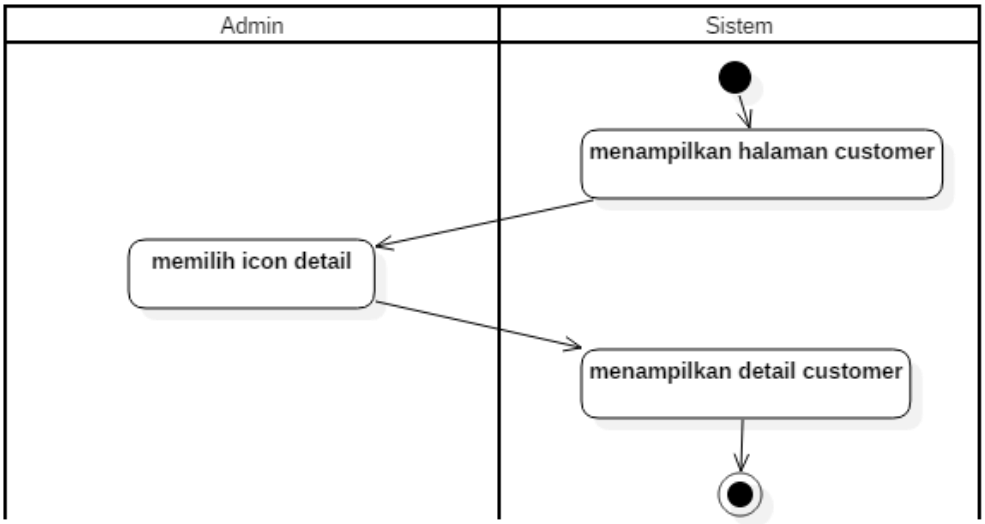
Dalam *activity* diagram *admin* dapat menghapus data *customer*. Tampilan *activity* diagram hapus *customer* dapat dilihat pada Gambar 4.25



Gambar 4. 25 *Activity* diagram *admin* hapus *customer*

25. *Activity* diagram *admin* lihat *detail customer*

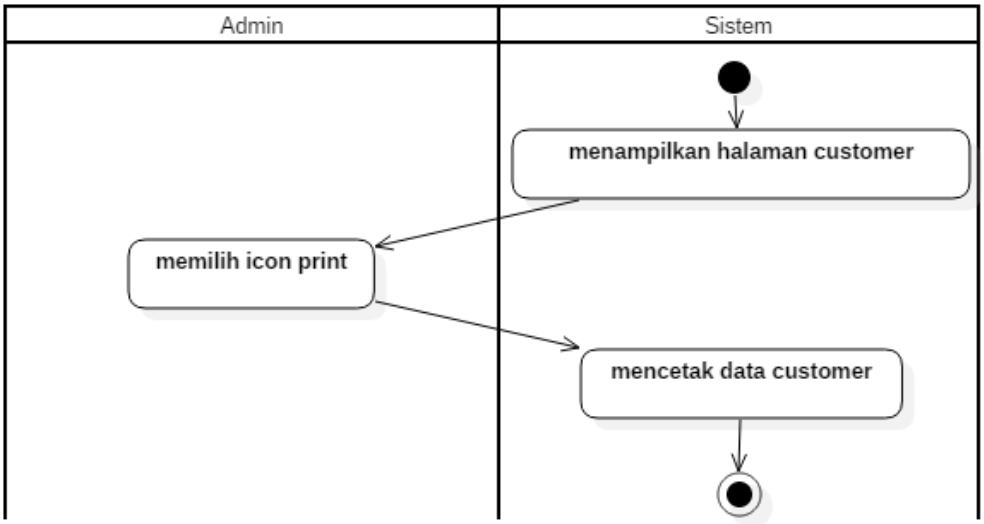
Dalam *activity* diagram *admin* dapat melihat *detail* data *customer* meliputi id *customer*, nama *customer*, alamat *customer*, *email* dan jenis. Tampilan *activity* diagram lihat *detail customer* dapat dilihat pada Gambar 4.26



Gambar 4. 26 *Activity* diagram *admin* lihat *detail customer*

26. *Activity* diagram *admin* cetak data *customer*

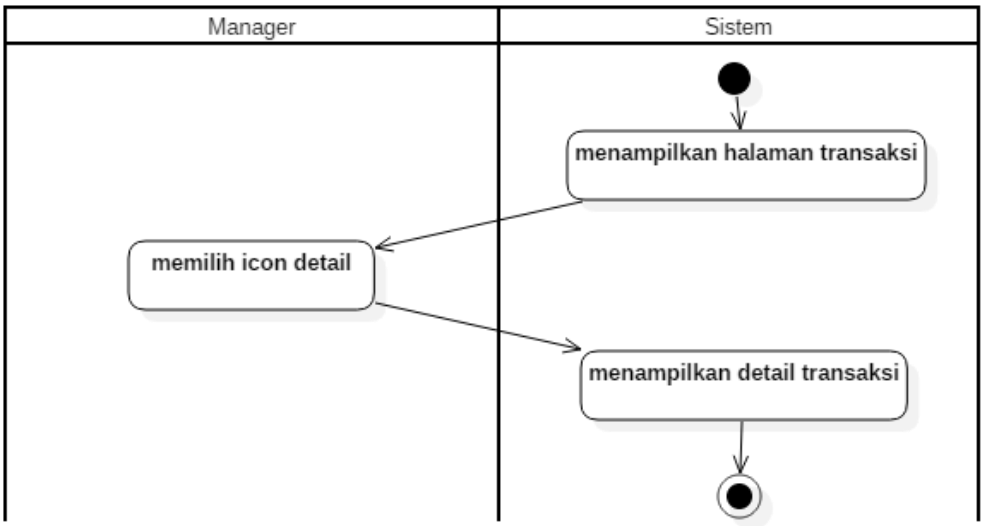
Dalam *activity* diagram *admin* dapat mencetak data *customer* meliputi id *customer*, nama *customer*, alamat *customer*, *email* dan jenis. Tampilan *activity* diagram cetak data *customer* dapat dilihat pada Gambar 4.27



Gambar 4. 27 *Activity* diagram *admin* cetak data *customer*

27. *Activity* diagram *manager* lihat *detail* transaksi

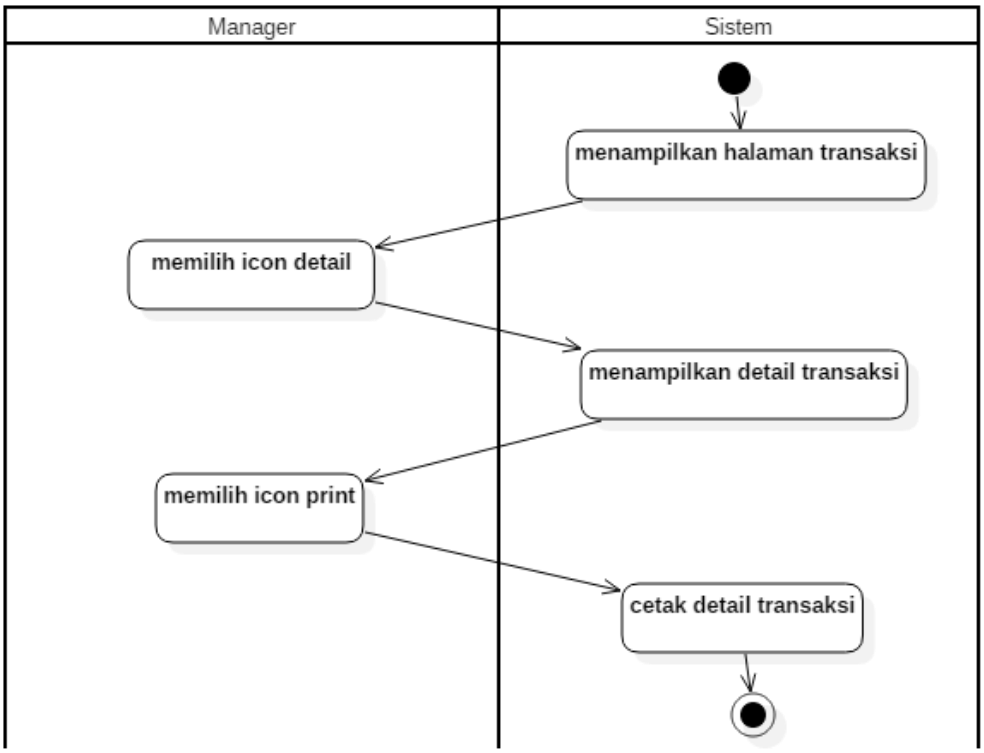
Dalam *activity* diagram *manager* dapat melihat *detail* data transaksi meliputi id transaksi, nama *customer*, lunas, status, total, tanggal masuk, tanggal kembali, nama barang, kuantitas, harga barang dan sub total. Tampilan *activity* diagram lihat *detail* transaksi dapat dilihat pada Gambar 4.28



Gambar 4. 28 *Activity* diagram *manager* lihat *detail* transaksi

28. *Activity* diagram *manager* cetak *detail* transaksi

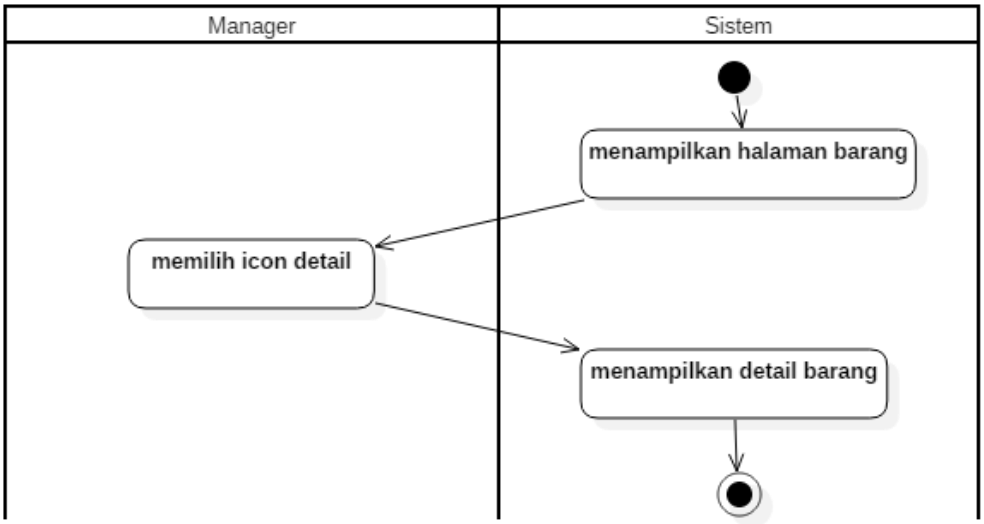
Dalam *activity* diagram *manager* dapat mencetak *detail* data transaksi meliputi id transaksi, nama *customer*, lunas, status, total, tanggal masuk, tanggal kembali, nama barang, kuantitas, harga barang dan sub total. Tampilan *activity* diagram cetak *detail* transaksi dapat dilihat pada Gambar 4.29



Gambar 4. 29 *Activity* diagram *manager* lihat *detail* transaksi

29. *Activity* diagram *manager* lihat *detail* barang

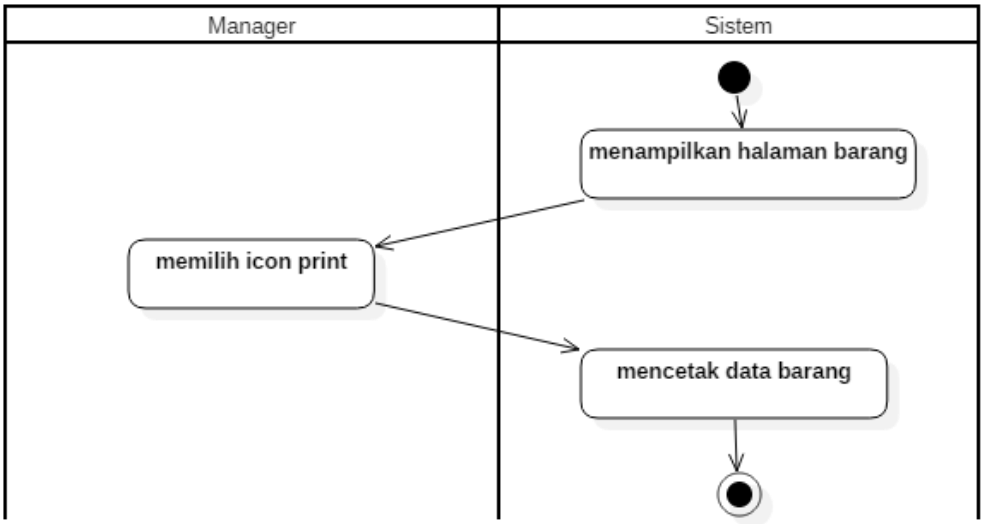
Dalam *activity* diagram *manager* dapat melihat *detail* data barang meliputi id barang, nama barang, dan harga barang. Tampilan *activity* diagram lihat *detail* barang dapat dilihat pada Gambar 4.30



Gambar 4. 30 *Activity* diagram *manager* lihat *detail* barang

30. *Activity* diagram *manager* cetak data barang

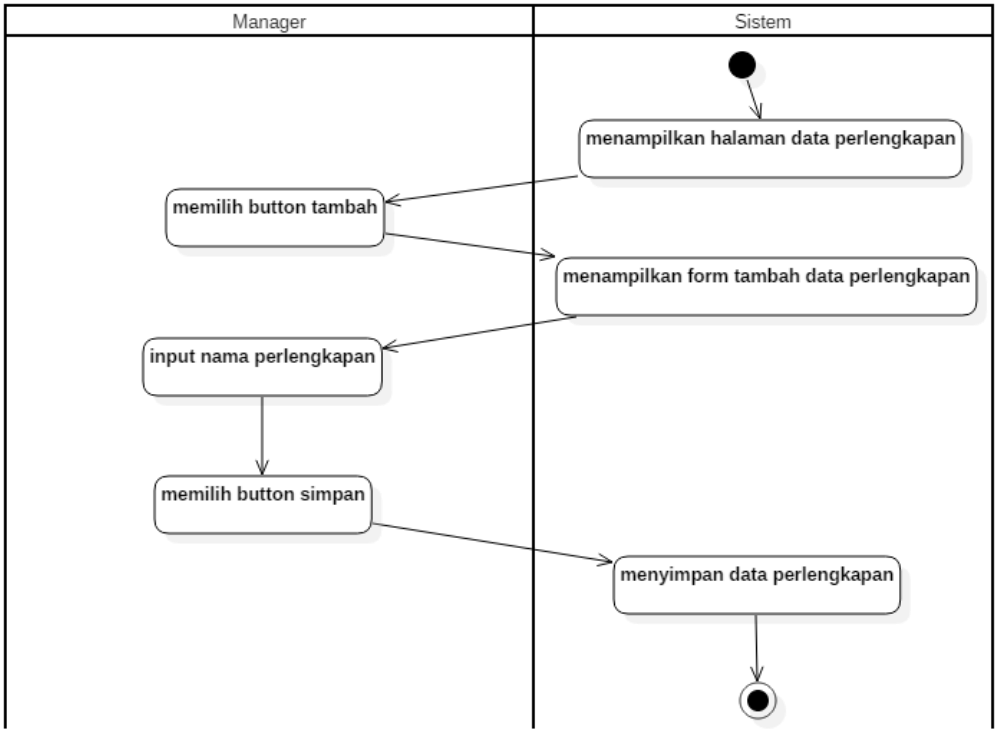
Dalam *activity* diagram *admin* dapat mencetak data barang meliputi id barang, nama barang, dan harga barang. Tampilan *activity* diagram cetak data barang dapat dilihat pada Gambar 4.31



Gambar 4. 31 *Activity* diagram *manager* cetak data barang

31. *Activity* diagram *manager* tambah data perlengkapan

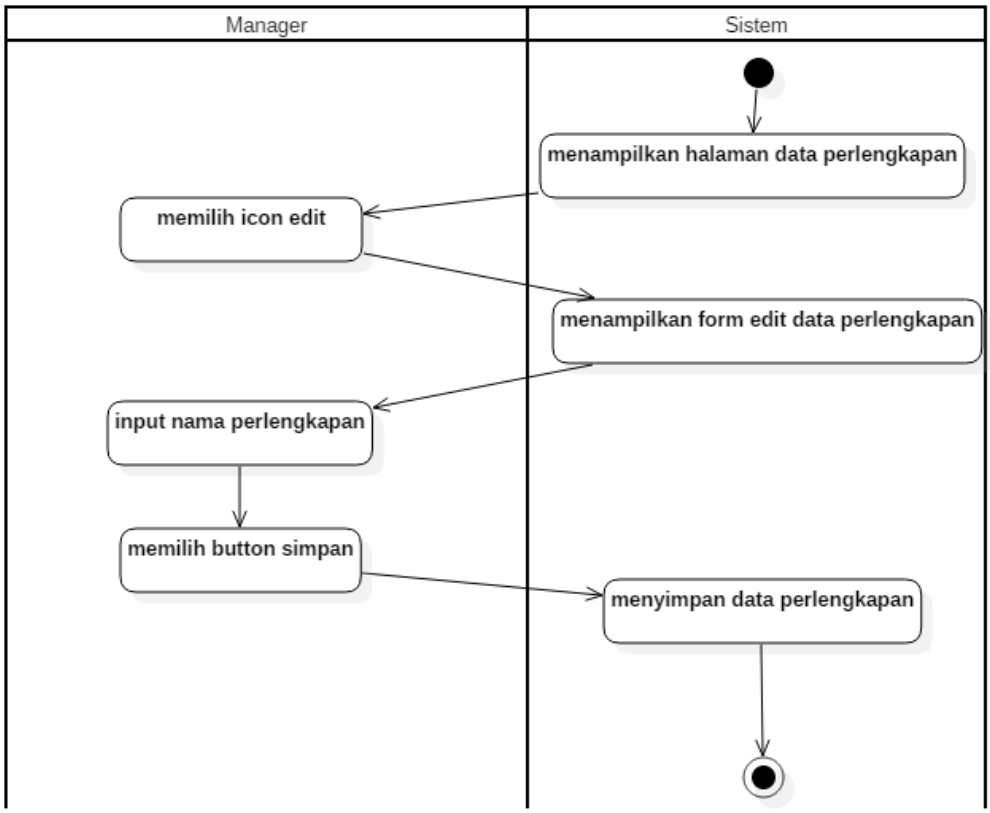
Dalam *activity* diagram *manager* dapat menambah data perlengkapan meliputi nama perlengkapan. Tampilan *activity* diagram tambah data perlengkapan dapat dilihat pada Gambar 4.32



Gambar 4. 32 *Activity* diagram *manager* tambah data perlengkapan

32. *Activity* diagram *manager* edit data perlengkapan

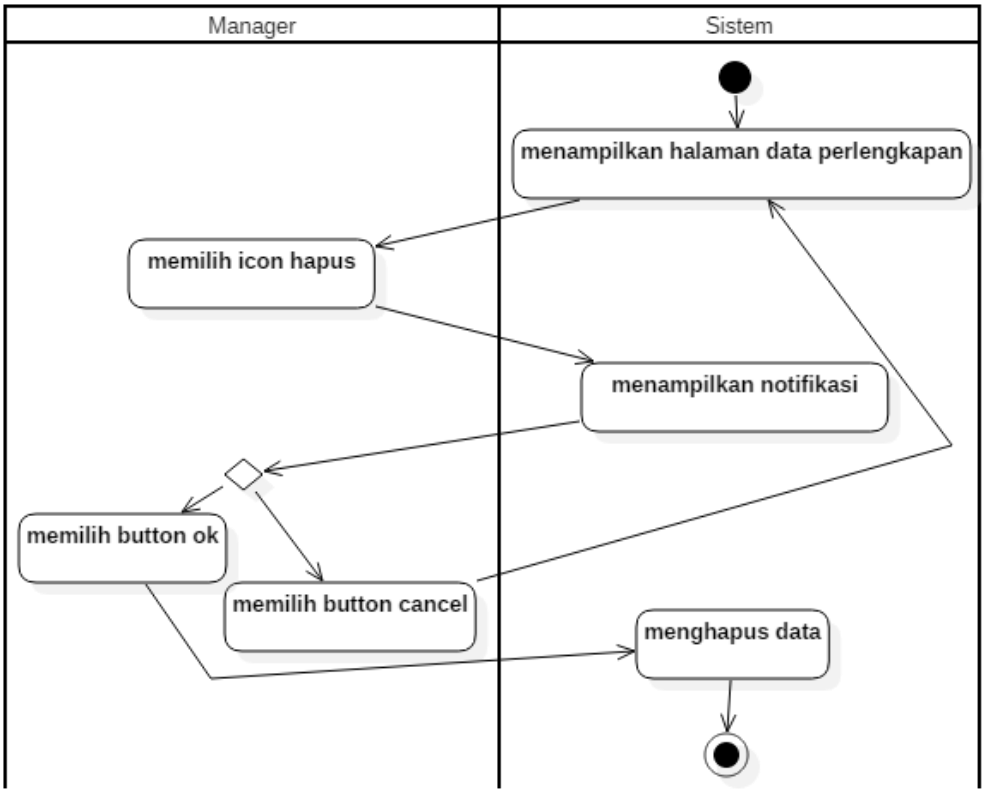
Dalam *activity* diagram *manager* dapat mengubah data perlengkapan meliputi nama perlengkapan. Tampilan *activity* diagram *edit* data perlengkapan dapat dilihat pada Gambar 4.33



Gambar 4. 33 *Activity* diagram *manager* edit data perlengkapan

33. *Activity* diagram *manager* hapus data perlengkapan

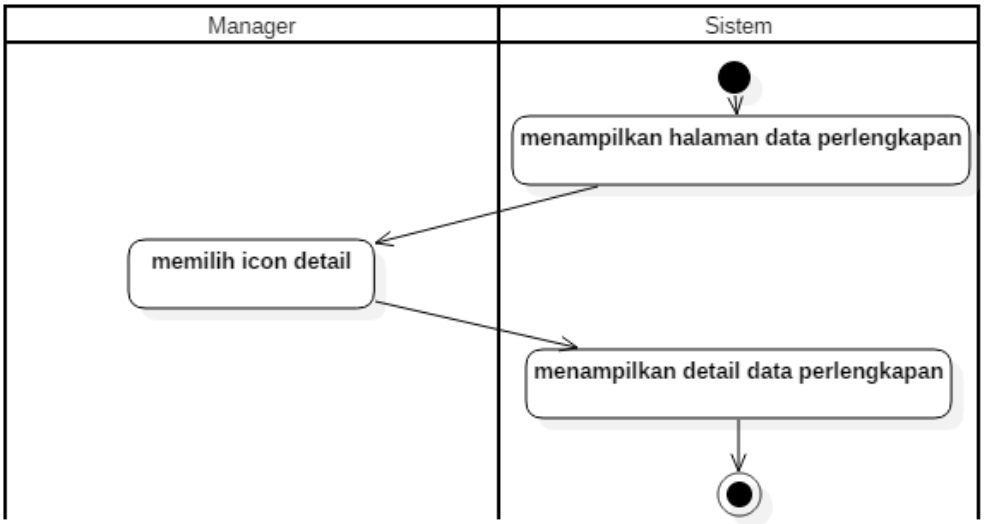
Dalam *activity* diagram *manager* dapat menghapus data perlengkapan. Tampilan *activity* diagram hapus data perlengkapan dapat dilihat pada Gambar 4.34



Gambar 4. 34 *Activity* diagram *manager* hapus data perlengkapan

34. *Activity* diagram *manager* lihat *detail* data perlengkapan

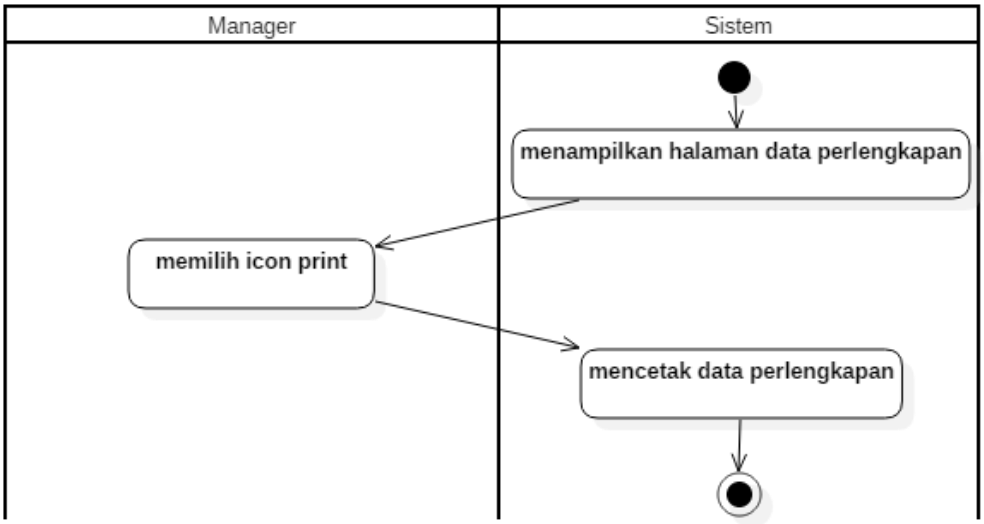
Dalam *activity* diagram *manager* dapat melihat *detail* data perlengkapan meliputi id perlengkapan dan nama perlengkapan. Tampilan *activity* diagram lihat *detail* data perlengkapan dapat dilihat pada Gambar 4.35



Gambar 4. 35 *Activity* diagram *manager* lihat *detail* data perlengkapan

35. *Activity* diagram *manager* cetak data perlengkapan

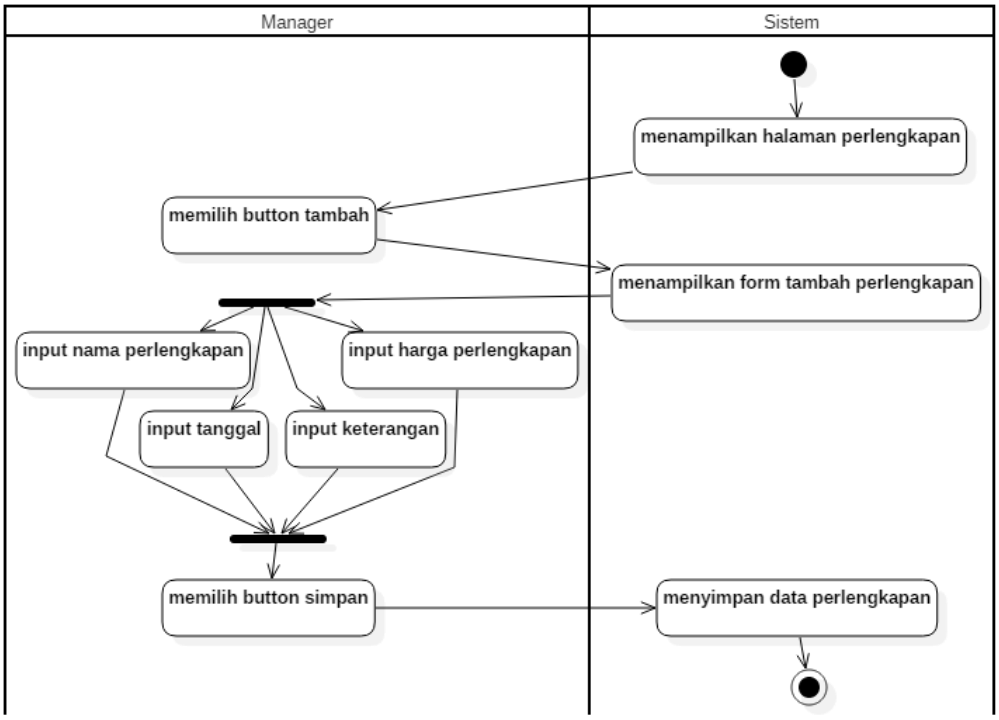
Dalam *activity* diagram *manager* dapat mencetak data perlengkapan meliputi id perlengkapan dan nama perlengkapan. Tampilan *activity* diagram cetak data perlengkapan dapat dilihat pada Gambar 4.36



Gambar 4. 36 *Activity* diagram *manager* cetak data perlengkapan

36. *Activity* diagram *manager* tambah perlengkapan

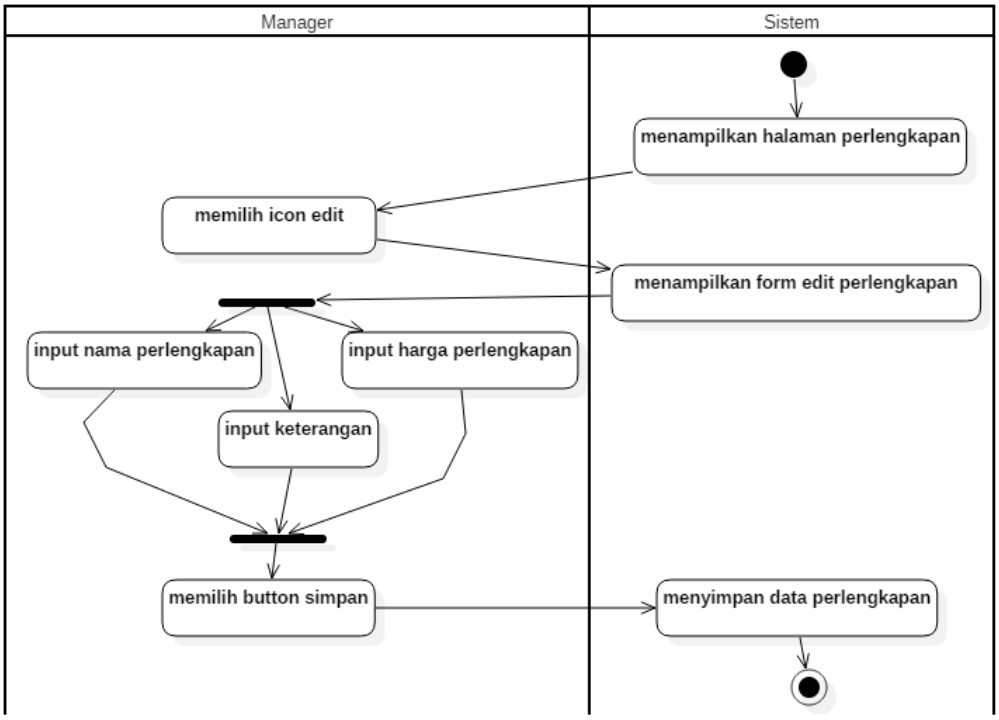
Dalam *activity* diagram *manager* dapat menambah perlengkapan meliputi tanggal, nama perlengkapan, harga dan keterangan. Tampilan *activity* diagram tambah perlengkapan dapat dilihat pada Gambar 4.37



Gambar 4. 37 *Activity* diagram *manager* tambah perlengkapan

37. *Activity* diagram *manager* edit perlengkapan

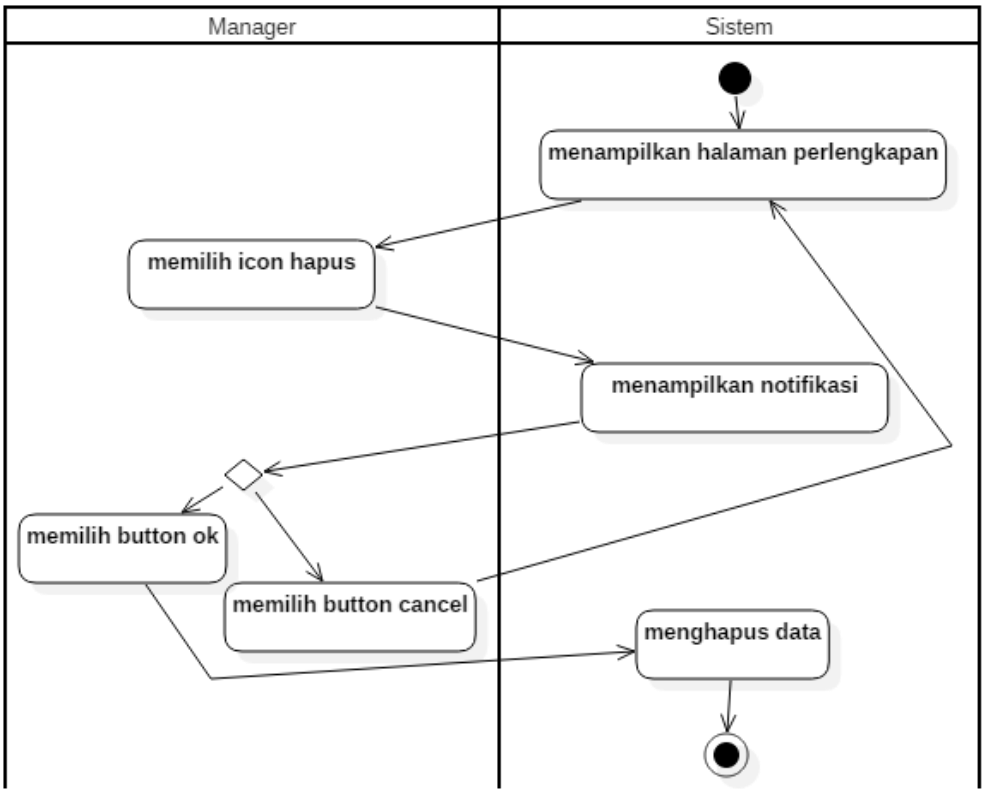
Dalam activity diagram *manager* dapat mengubah perlengkapan meliputi nama perlengkapan, harga dan keterangan. Tampilan *activity* diagram *edit* perlengkapan dapat dilihat pada Gambar 4.38



Gambar 4. 38 *Activity* diagram *manager* edit perlengkapan

38. *Activity* diagram *manager* hapus perlengkapan

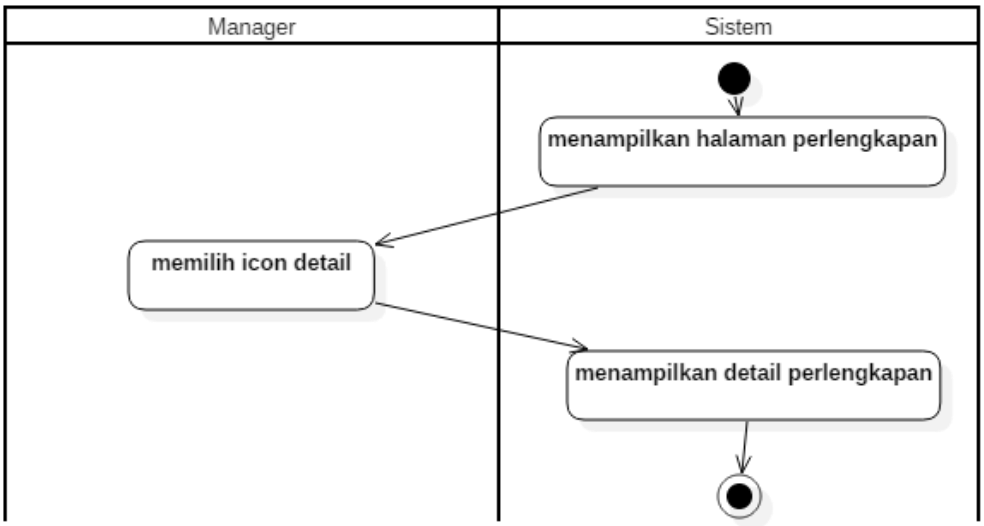
Dalam *activity* diagram *manager* dapat menghapus perlengkapan. Tampilan ***activity*** diagram hapus perlengkapan dapat dilihat pada Gambar 4.39



Gambar 4. 39 *Activity* diagram *manager* hapus perlengkapan

39. *Activity* diagram *manager* lihat *detail* perlengkapan

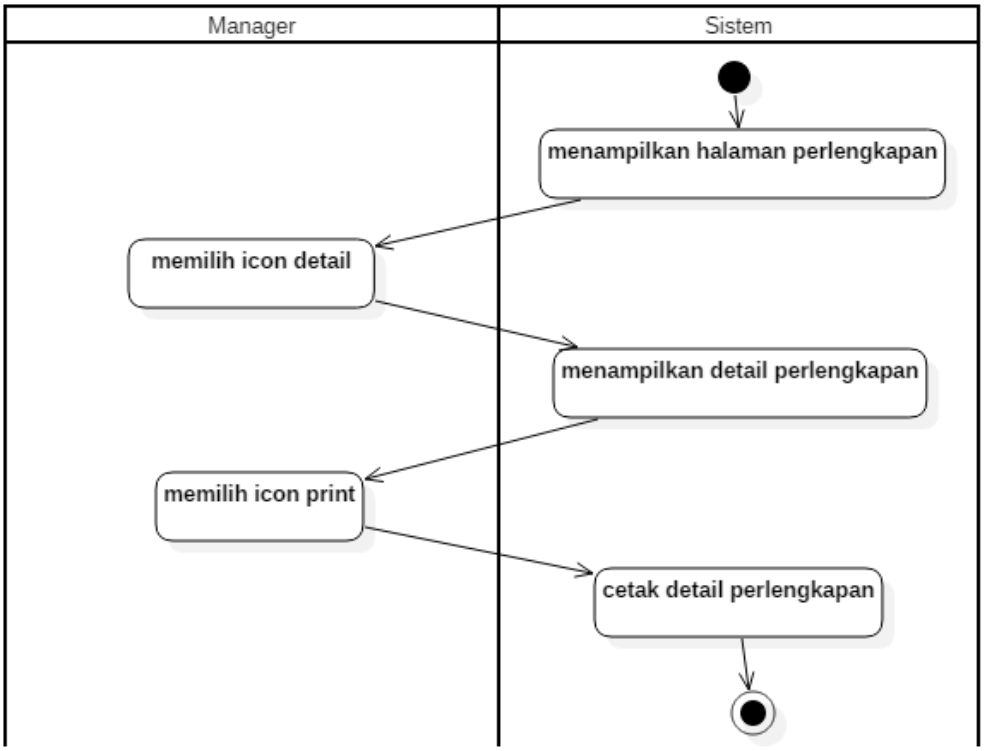
Dalam *activity* diagram *manager* dapat melihat *detail* perlengkapan meliputi id perlengkapan, tanggal, nama perlengkapan, harga dan keterangan. Tampilan *activity* diagram lihat *detail* perlengkapan dapat dilihat pada Gambar 4.40



Gambar 4. 40 *Activity* diagram *manager* lihat *detail* perlengkapan

40. *Activity* diagram *manager* cetak perlengkapan

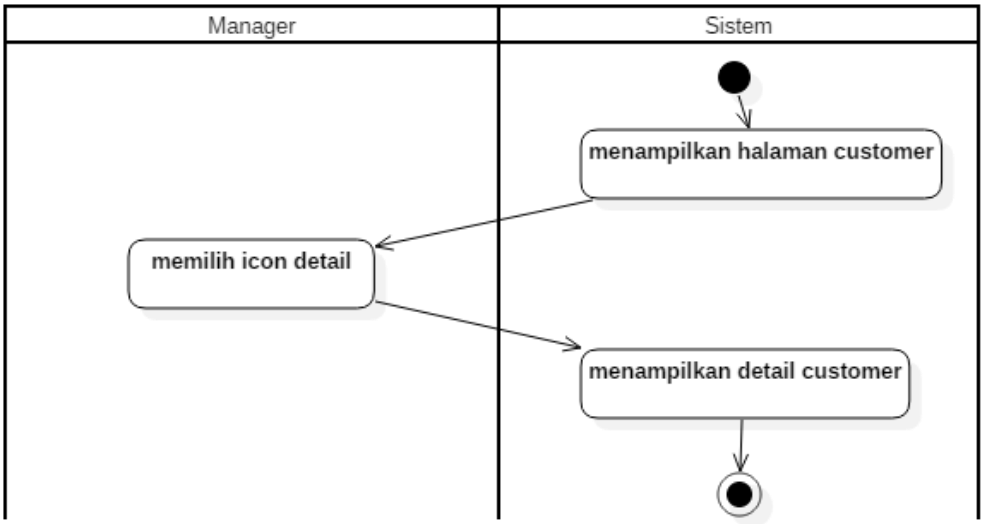
Dalam *activity* diagram *manager* dapat mencetak perlengkapan meliputi id perlengkapan, tanggal, nama perlengkapan, harga dan keterangan. Tampilan *activity* diagram lihat *detail* perlengkapan dapat dilihat pada Gambar 4.41



Gambar 4. 41 *Activity* diagram *manager* cetak perlengkapan

41. *Activity* diagram *manager* lihat *detail customer*

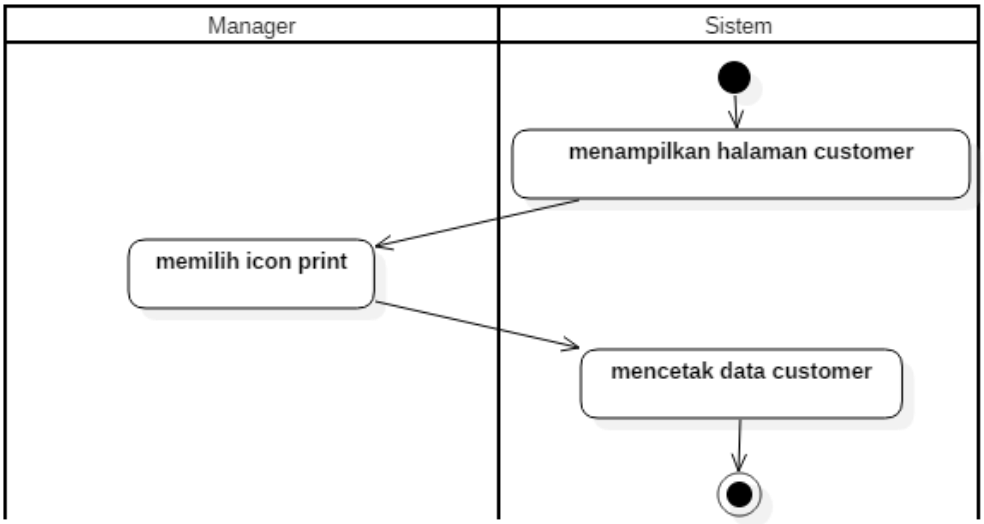
Dalam *activity* diagram *manager* dapat melihat *detail* data *customer* meliputi id *customer*, nama *customer*, alamat *customer*, *email* dan jenis. Tampilan *activity* diagram lihat *detail customer* dapat dilihat pada Gambar 4.42



Gambar 4. 42 *Activity* diagram *manager* lihat *detail customer*

42. *Activity* diagram *manager* cetak data *customer*

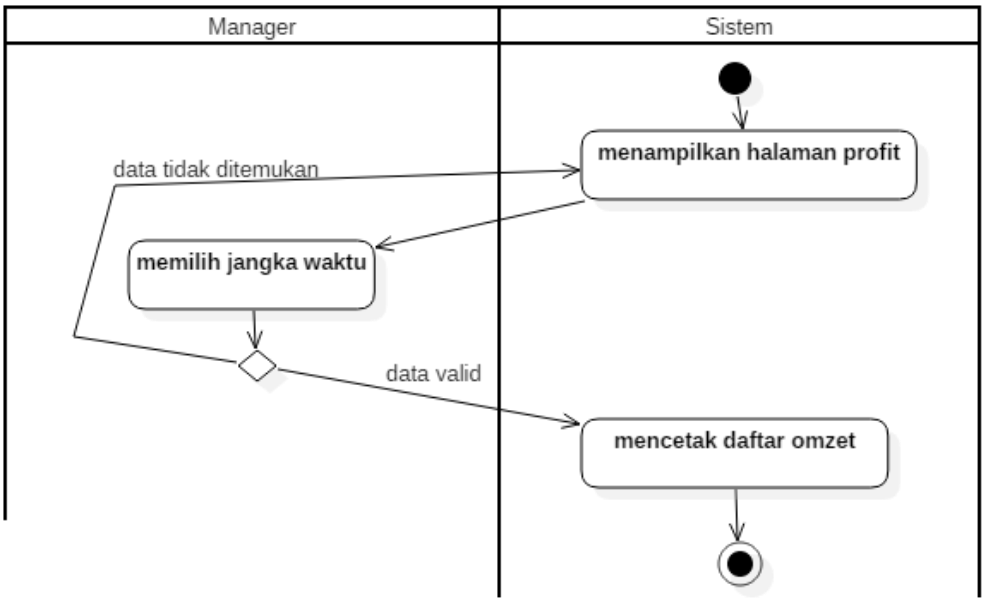
Dalam *activity* diagram *manager* dapat mencetak data *customer* meliputi id *customer*, nama *customer*, alamat *customer*, *email* dan jenis. Tampilan *activity* diagram cetak data *customer* dapat dilihat pada Gambar 4.43



Gambar 4. 43 *Activity* diagram *manager* cetak data *customer*

43. *Activity* diagram *manager* pilih laporan *omzet*

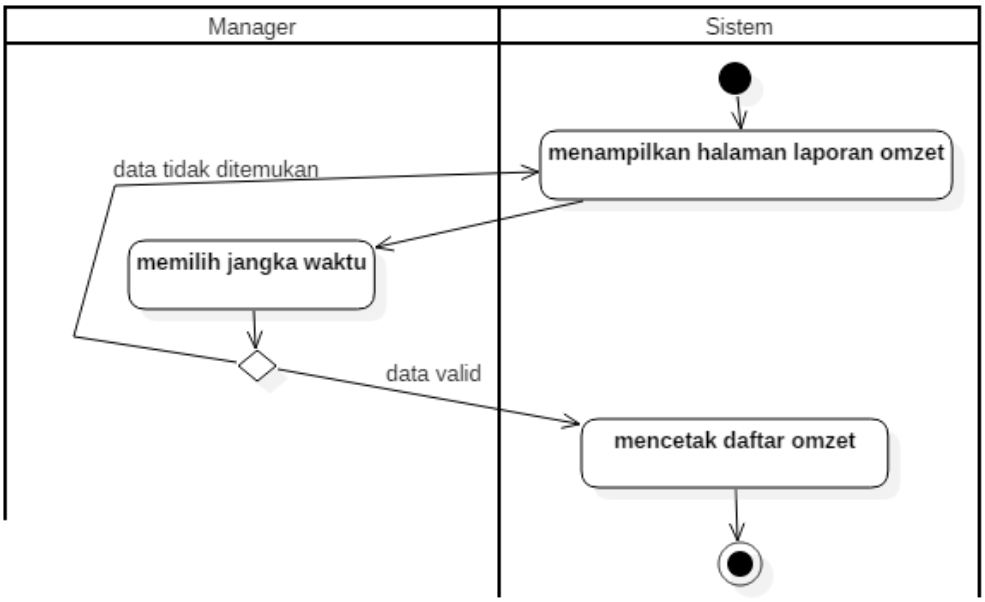
Dalam *activity* diagram *manager* dapat memilih laporan *omzet* per-hari atau per-bulan, kemudian data yang ditampilkan meliputi id transaksi, nama *customer*, lunas, status, total, tanggal masuk, tanggal kembali,dan total. Tampilan *activity* diagram pilih laporan *omzet* dapat dilihat pada Gambar 4.44



Gambar 4. 44 *Activity* diagram *manager* pilih laporan *omzet*

44. *Activity* diagram *manager* cetak laporan *omzet*

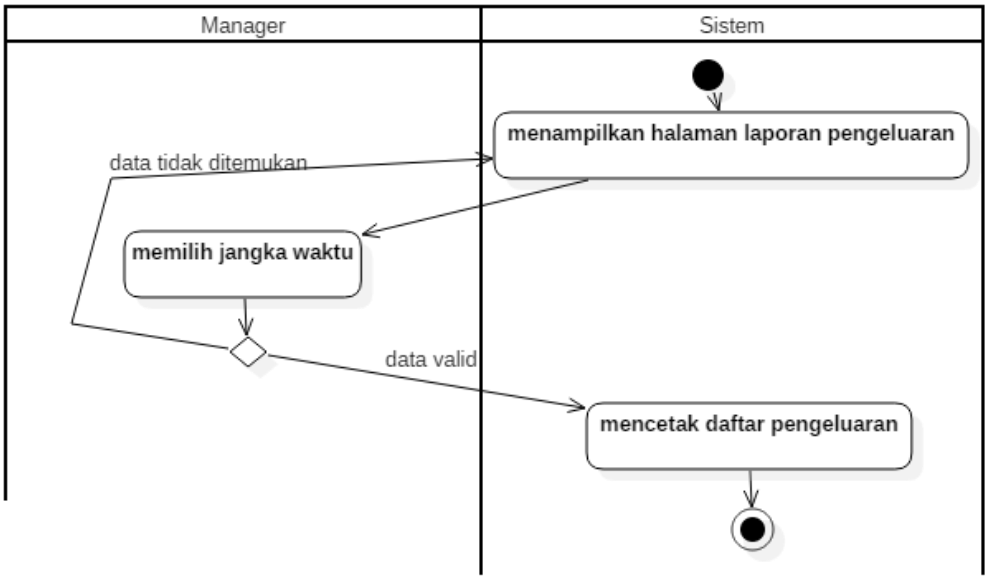
Dalam *activity* diagram *manager* dapat mencetak laporan *omzet* per-hari atau per-bulan, kemudian data yang ditampilkan meliputi id transaksi, nama *customer*, lunas, status, total, tanggal masuk, tanggal kembali,dan total. Tampilan *activity* diagram cetak laporan *omzet* dapat dilihat pada Gambar 4.45



Gambar 4. 45 *Activity* diagram *manager* cetak laporan *omzet*

45. *Activity* diagram *manager* pilih laporan pengeluaran

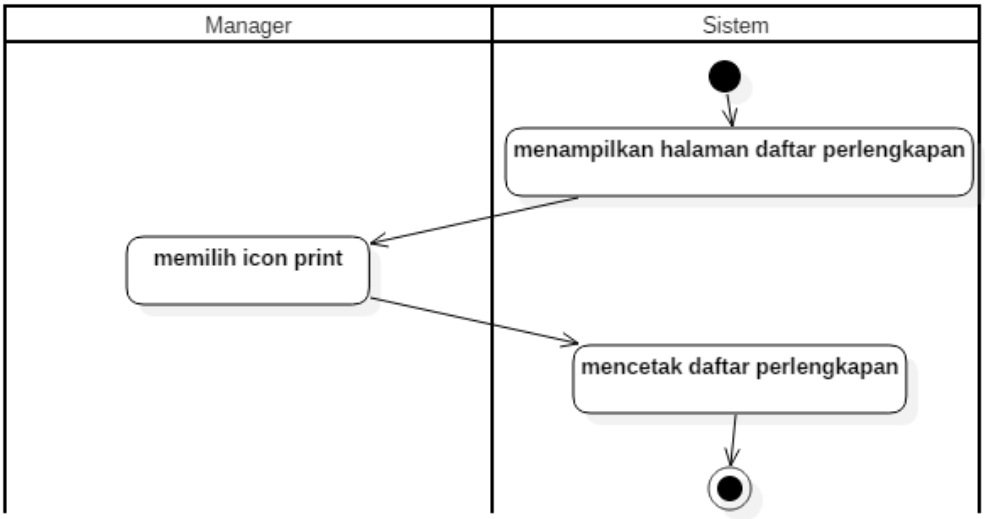
Dalam *activity* diagram *manager* dapat memilih laporan *omzet* per-hari atau per-bulan, kemudian data yang ditampilkan meliputi id perlengkapan, tanggal, dan total. Tampilan *activity* diagram pilih laporan pengeluaran dapat dilihat pada Gambar 4.46



Gambar 4. 46 *Activity* diagram *manager* pilih laporan pengeluaran

46. *Activity* diagram *manager* cetak laporan pengeluaran

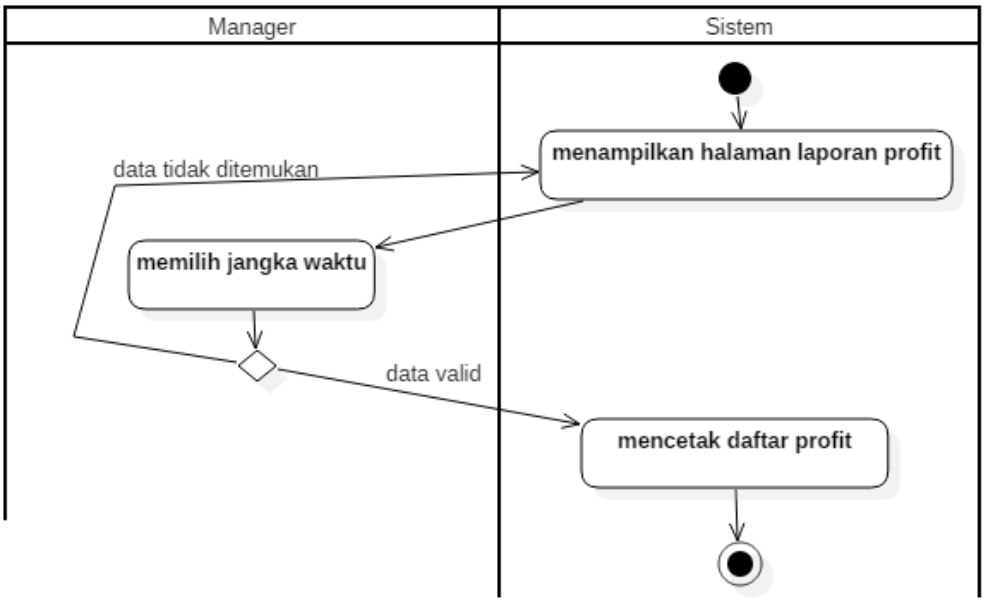
Dalam *activity* diagram *manager* dapat mencetak laporan pengeluaran meliputi id perlengkapan, tanggal, dan total. Tampilan *activity* diagram cetak laporan pengeluaran dapat dilihat pada Gambar 4.47



Gambar 4. 47 *Activity* diagram *manager* cetak laporan pengeluaran

47. *Activity* diagram *manager* pilih laporan *profit*

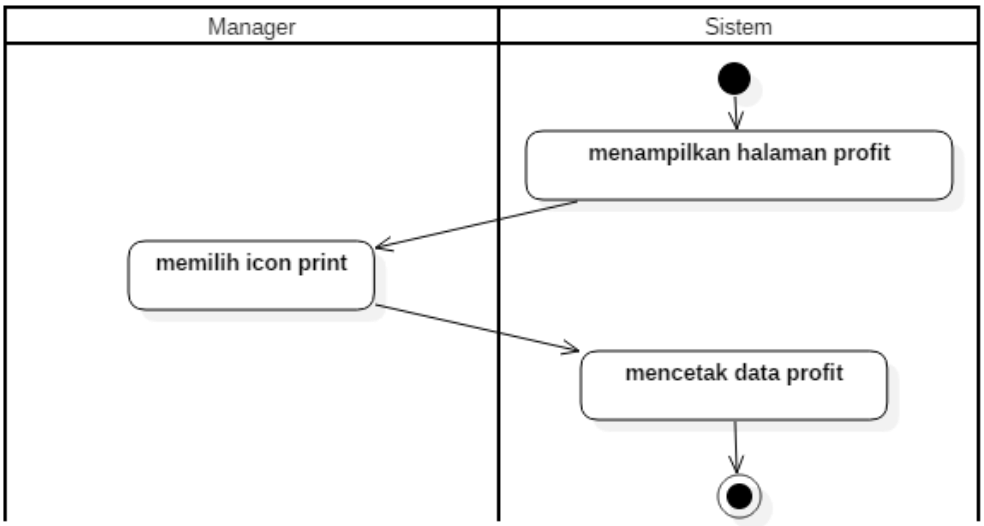
Dalam *activity* diagram *manager* dapat memilih laporan *profit* per-hari atau per-bulan, kemudian data yang ditampilkan meliputi laporan *omzet*, laporan penglengkapan dan *profit*. Tampilan *activity* diagram pilih laporan *profit* dapat dilihat pada Gambar 4.48



Gambar 4. 48 *Activity* diagram *manager* pilih laporan *profit*

48. *Activity* diagram *manager* cetak laporan *profit*

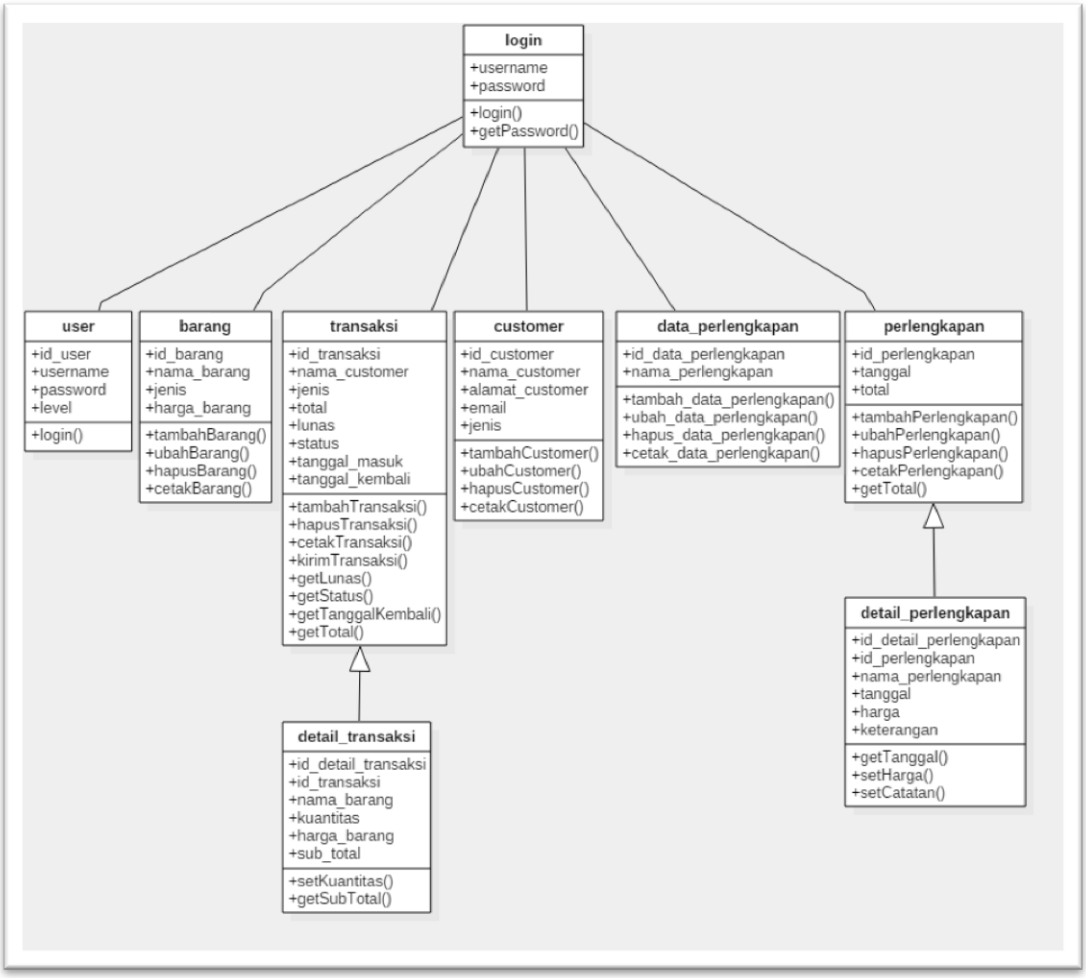
Dalam *activity* diagram *manager* dapat mencetak laporan *profit* per-hari atau per-bulan, kemudian data yang ditampilkan meliputi laporan *omzet*, laporan penglengkapan dan *profit*. Tampilan *activity* diagram pilih laporan *profit* dapat dilihat pada Gambar 4.49



Gambar 4. 49 *Activity* diagram *manager* cetak laporan *profit*

***Class* diagram**

Class diagram digunakan untuk mendeskripsikan jenis-jenis objek dalam sistem dam berbagai macam hubungan statis yang terjadi. *Class* diagram Sistem Pengolahan Transaksi Laundry dapat dilihat pada Gambar 4.50

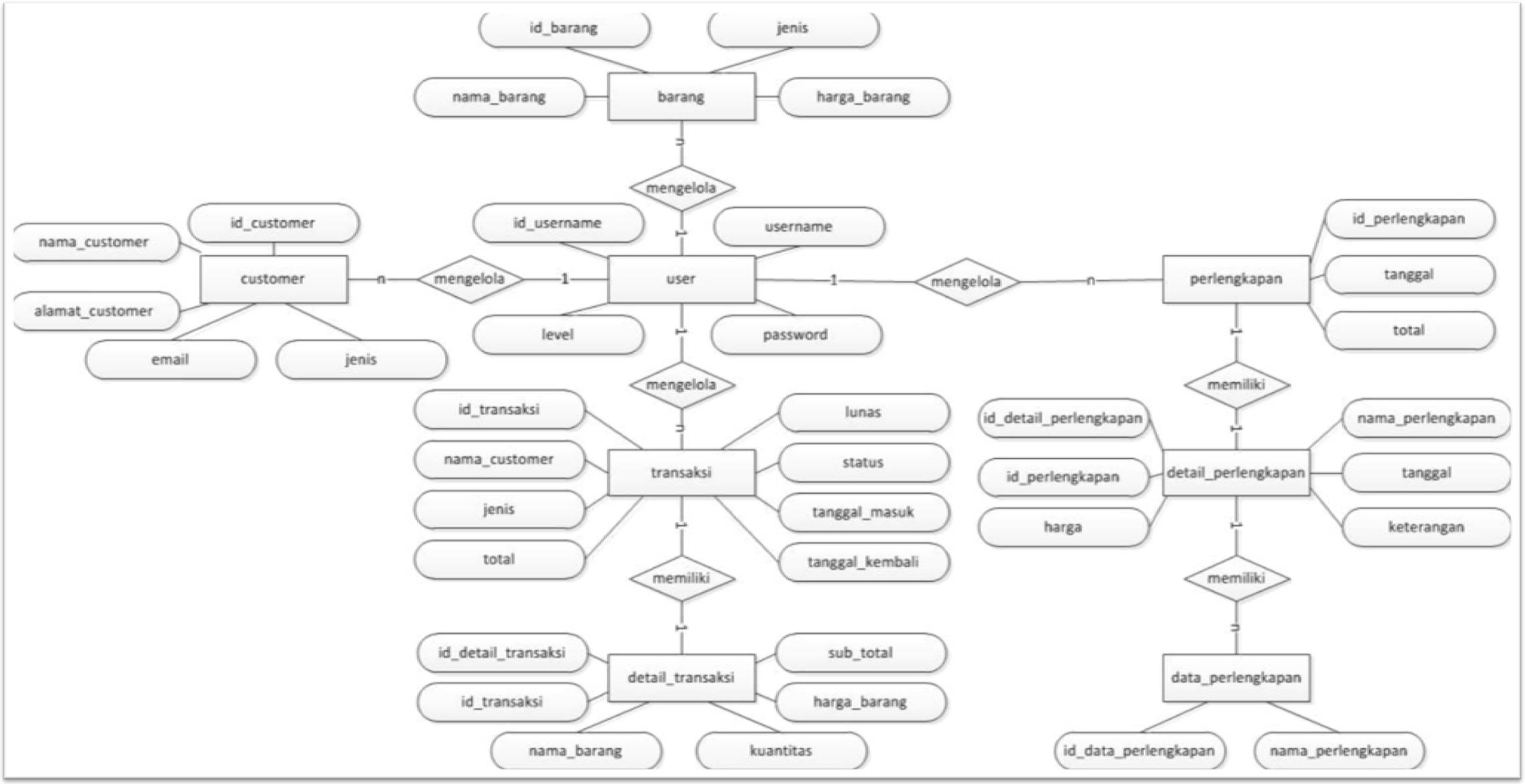


Gambar 4. 50 *Class* Diagram

4.2.2 Perancangan basis data

Pada sistem ini, perancangan basis data dibuat dengan menggunakan *Entity Relationship Diagram* (ERD) yang nantinya dapat diimplementasikan dalam pembuatan tabel-tabel basis data. *Entity Relationship Diagram* (ERD) merupakan perangkat yang digunakan untuk menggambarkan diagram yang berisi entitas maupun relasi yang ada dalam basisdata. Diagram ER pada sistem laundry ini menggambarkan hubungan antar entitas yang kemudian dapat dijadikan acuan dalam pembuatan tabel-tabel basis data.

Relasi utama dari diagram ER ini dapat dilihat pada Gambar 4.51.



Q

Gambar 4. 51 Entity Relationship Diagram (ERD)

Dari hasil desain Diagram E-R yang telah dibuat pada Gambar 4.51 maka dapat diimplementasikan dan dirancang menjadi 8 tabel, antara lain tabel barang, tabel *customer*, tabel data perlengkapan, tabel *detail* perlengkapan, tabel *detail* transaksi, tabel perlengkapan, tabel transaksi dan tabel user. Penjelasan dari masing – masing peranangan tabel tersebut adalah sebagai berikut :

1. Tabel *User*

   Tabel *user* merupakan tabel yang berisi atribut tentang pengguna sistem. Pengguna sistem ini adalah *admin* dan *manager*. Perincian rancangan tabel *user* dapat dilihat pada Tabel 4.1.

Tabel 4. 1 Rancangan Tabel User

Atribut	Tipe Data	Keterangan
Id_username	Int(11)	Primary Key, Auto Increment
Username	Varchar(20)	Username pengguna
Password	Varchar(50)	Kode unik yang digunakan untuk autentifikasi sistem
Level	Varchar(20)	Level untuk menentukan hak akses pada sistem

2. Tabel Barang

   Tabel barang merupakan tabel yang berisi atribut tentang data barang. Perincian rancangan tabel barang dapat dilihat pada Tabel 4.2.

Tabel 4. 2 Rancangan Tabel Barang

Atribut	Tipe Data	Keterangan
Id_barang	Int(11)	Primary Key, Auto Increment
Nama_barang	Varchar(20)	Menyimpan nama barang
Jenis	Varchar(20)	Menyimpan jenis barang
Harga_barang	Double	Menyimpan harga barang

3. Tabel *Customer*

   Tabel *customer* merupakan tabel yang berisi atribut tentang data *customer*. Perincian rancangan tabel *customer* dapat dilihat pada Tabel 4.3.

Tabel 4. 3 Rancangan Tabel *Customer*

Atribut	Tipe Data	Keterangan
Id_customer	Int(11)	Primary Key, Auto Increment
Nama_customer	Varchar(20)	Menyimpan nama customer
Alamat_customer	Varchar(50)	Menyimpan alamat customer
Email	Varchar(20)	Menyimpan email customer
Jenis	Varchar(20)	Menyimpan jenis customer

4. Tabel Data Perlengkapan

Tabel data perlengkapan merupakan tabel yang berisi atribut tentang data perlengkapan. Perincian rancangan tabel data perlengkapan dapat dilihat pada Tabel 4.4.

Tabel 4. 4 Rancangan Tabel Data Perlengkapan

Atribut	Tipe Data	Keterangan
Id_data_perlengkapan	Int(11)	Primary Key, Auto Increment
Nama_perlengkapan	Varchar(20)	Menyimpan nama perlengkapan

5. Tabel *Detail* Perlengkapan

Tabel *detail* perlengkapan merupakan tabel yang berisi atribut tentang *detail* pembelian perlengkapan. Perincian rancangan tabel *detail* perlengkapan dapat dilihat pada Tabel 4.45.

Tabel 4. 5 Rancangan Tabel *Detail* Perlengkapan

Atribut	Tipe Data	Keterangan
Id_detail_perlengkapan	Int(11)	Primary Key, Auto Increment
Id_perlengkapan	Int(11)	Foreign Key dari tabel perlengkapan
Nama_perlengkapan	Varchar(20)	Menyimpan nama perlengkapan
Tanggal	Datetime	Menyimpan tanggal pembelian perlengkapan
Harga	Double	Menyimpan harga perlengkapan
Keterangan	Varchar(20)	Menyimpan kuantitas perlengkapan

6. Tabel Perlengkapan

Tabel perlengkapan merupakan tabel yang berisi atribut tentang pembelian perlengkapan. Perincian rancangan tabel perlengkapan dapat dilihat pada Tabel 4.6.

Tabel 4. 6 Rancangan Tabel Perlengkapan

Atribut	Tipe Data	Keterangan
Id_perlengkapan	Int(11)	Primary Key, Auto Increment
Tanggal	Datetime	Menyimpan tanggal pembelian perlengkapan
Total	Double	Menyimpan total harga perlengkapan

7. Tabel *Detail* Transaksi

Tabel *detail* transaksi merupakan tabel yang berisi atribut tentang *detail* transaksi. Perincian rancangan tabel *detail* transaksi dapat dilihat pada Tabel 4.7.

Tabel 4. 7 Rancangan Tabel *Detail* Transaksi

Atribut	Tipe Data	Keterangan
Id_detail_transaksi	Int(11)	Primary Key, Auto Increment
Id_transaksi	Int(11)	Foreign Key dari tabel perlengkapan
Nama_barang	Varchar(20)	Menyimpan nama barang
Kuantitas	Int(3)	Menyimpan kuantitas barang
Harga_barang	Double	Menyimpan harga barang
Sub_total	Double	Menyimpan sub total barang

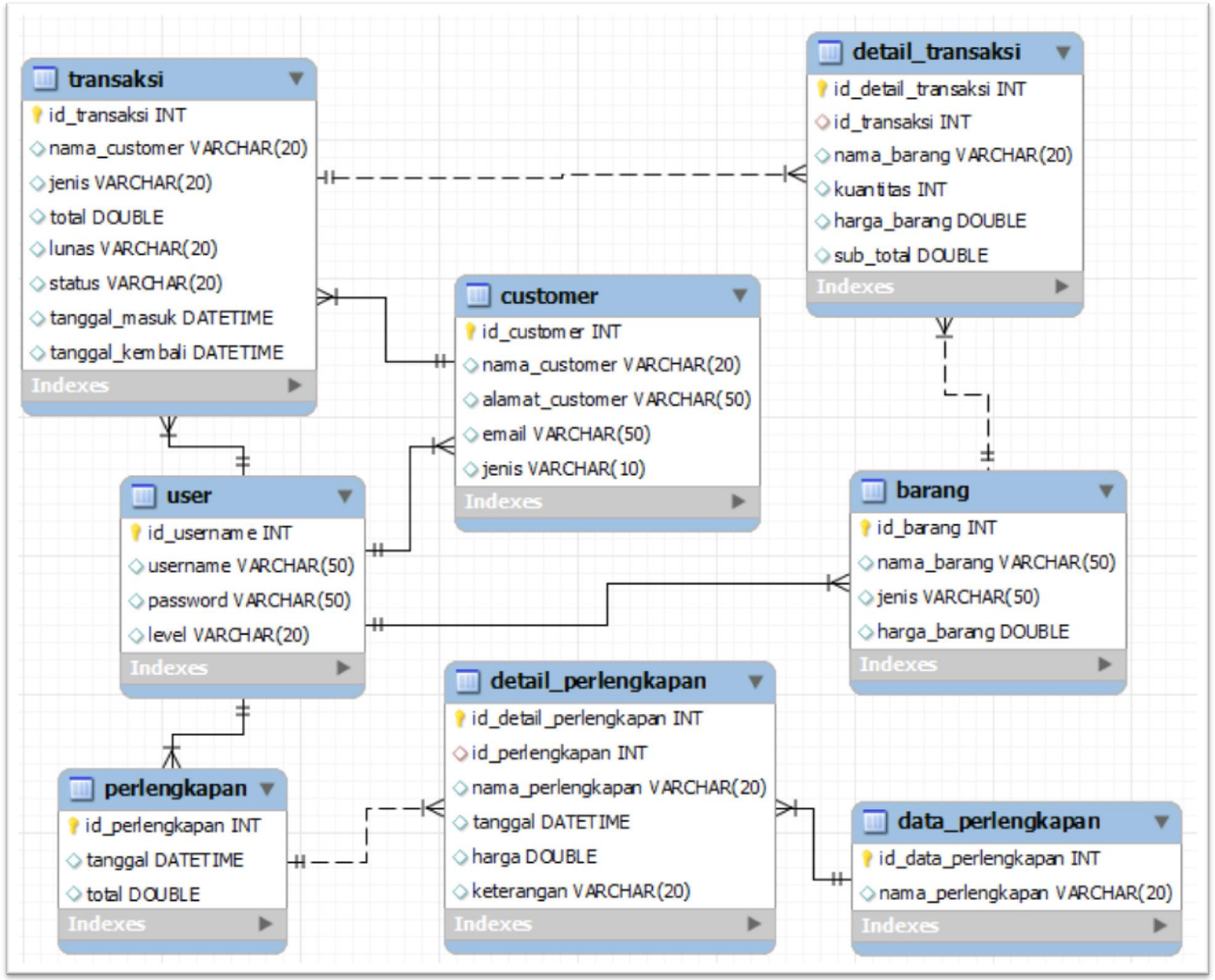
8. Tabel Transaksi

Tabel transaksi merupakan tabel yang berisi atribut tentang transaksi. Perincian rancangan tabel transaksi dapat dilihat pada Tabel 4.8.

Tabel 4. 8 Rancangan Tabel Transaksi

Atribut	Tipe Data	Keterangan
Id_transaksi	Int(11)	Primary Key, Auto Increment
Nama_customer	Varchar(20)	Menyimpan nama customer
Jenis	Varchar(20)	Menyimpan jenis customer
Total	Double	Menyimpan total transaksi
Lunas	Varchar(20)	Menyimpan status lunas/belum lunas
Status	Varchar(20)	Menyimpan status cucian sedang dikerjakan/selesai
Tanggal_masuk	Datetime	Menyimpan tanggal cucian masuk
Tanggal_kembali	Datetime	Menyimpan tanggal cucian kembali

Dari rancangan tabel-tabel di atas maka dapat dibuat rancangan relasi antar tabel yang dapat dilihat pada Gambar 4.52.

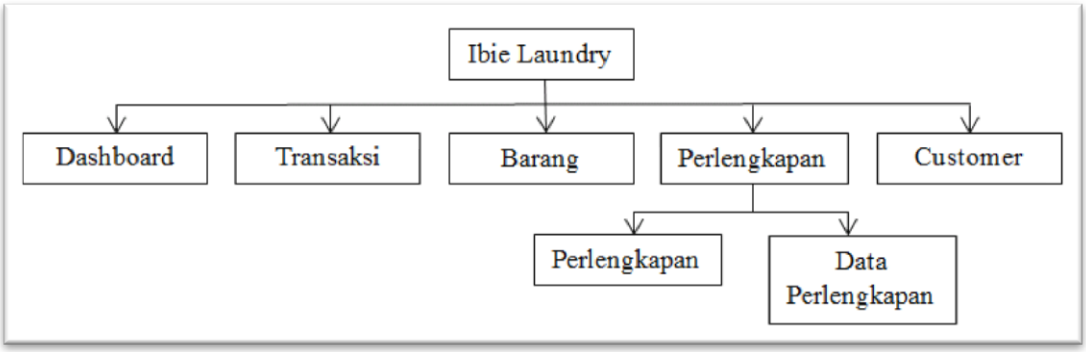


Gambar 4. 52 Rancangan Relasi Antar Tabel

4.2.3 Perancangan struktur menu

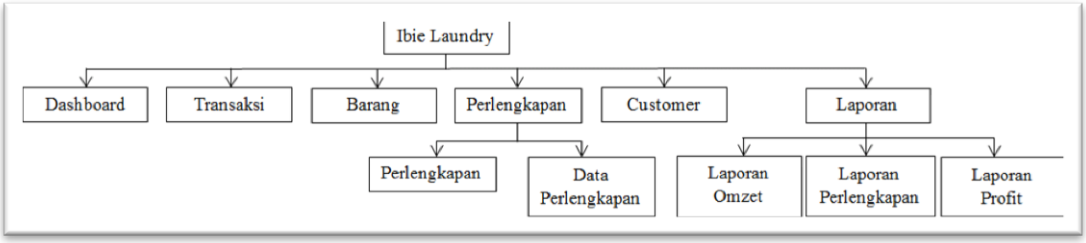
Sistem Informasi Pengolahan Transaksi Laundry pada *Ibie Laundry & Wet Cleaning* mempunyai dua pengguna yaitu *admin* dan *manager*. Masing-masing pengguna memiliki kebutuhan akan bisnis proses yang berbeda-beda. Berdasarkan hal tersebut struktur menu untuk masing-masing pengguna dirancang sesuai kebutuhan masing-masing pengguna.

Untuk struktur menu admin dapat dilihat pada Gambar 4.53



Gambar 4. 53 Struktur menu *admin*

Untuk struktur menu *manager* Gambar 4.54



Gambar 4. 54 Struktur menu *manager*

4.2.4 Perancangan antarmuka pengguna

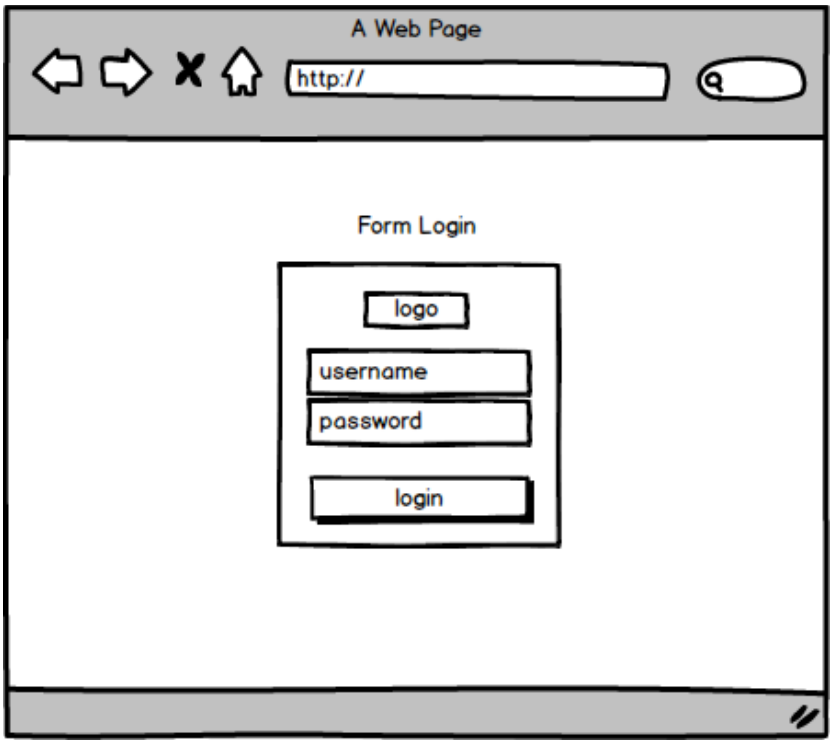
Perancangan antarmuka pengguna adalah salah satu bagian dari tahapan perancangan sistem dalam pembuatan sistem informasi pengolahan transaksi laundry. Antarmuka adalah media yang akan menghubungkan interaksi

antara sistem dengan pengguna. Rancangan antarmuka akan menunjukkan bagaimana komunikasi antara pengguna sistem dengan komputer. Komunikasi tersebut dapat berupa aktivitas memasukkan data ke sistem, menampilkan informasi atau meminta keluaran dari sistem. Perancangan antar muka untuk *admin* dan *manager* sebenarnya hampir sama, perbedaanya hanya pada menu laporan yang hanya dapat diakses oleh *manager*.

Pada antar muka *admin* terdapat menu transaksi, barang, perlengkapan dan *customer*. Sedangkan untuk antar muka *manager* sama dengan antar muka *admin* tetapi terdapat satu menu tambahan, yaitu menu laporan yang hanya bisa diakses oleh *manager*.

1. Rancangan halaman login

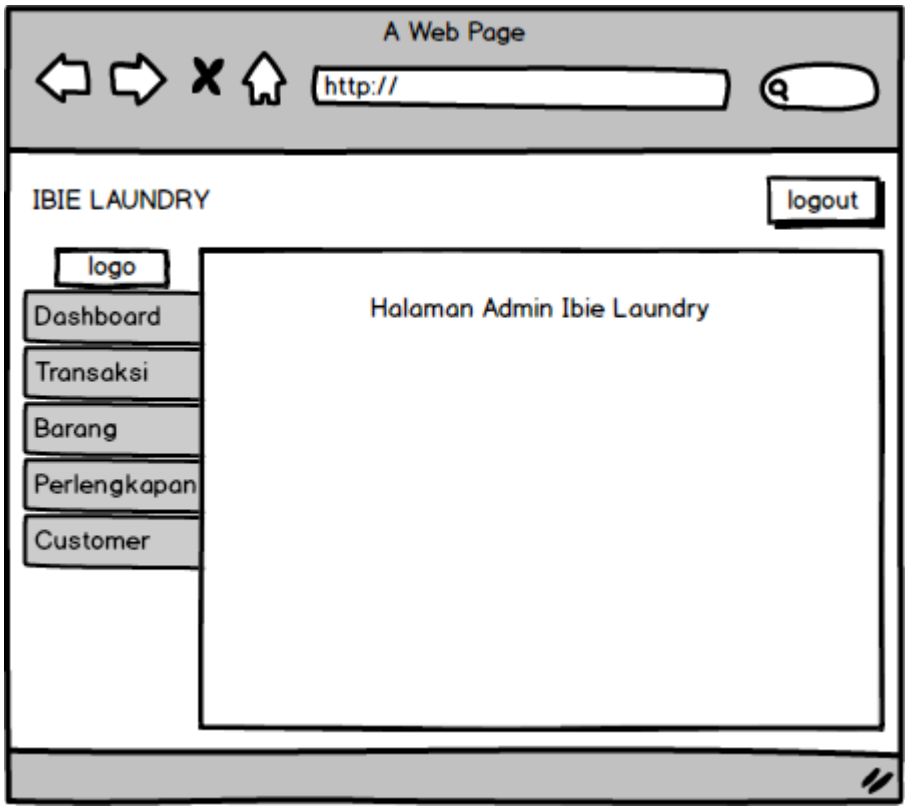
Halaman *login* digunakan untuk verifikasi pengguna yang akan masuk ke sistem. Semua pengguna baik *admin* dan *manager* yang akan masuk ke sistem melalui prosses login, disini pengguna diminta memasukkan username dan password. Rancangan halaman login dapat dilihat pada Gambar 4.55.



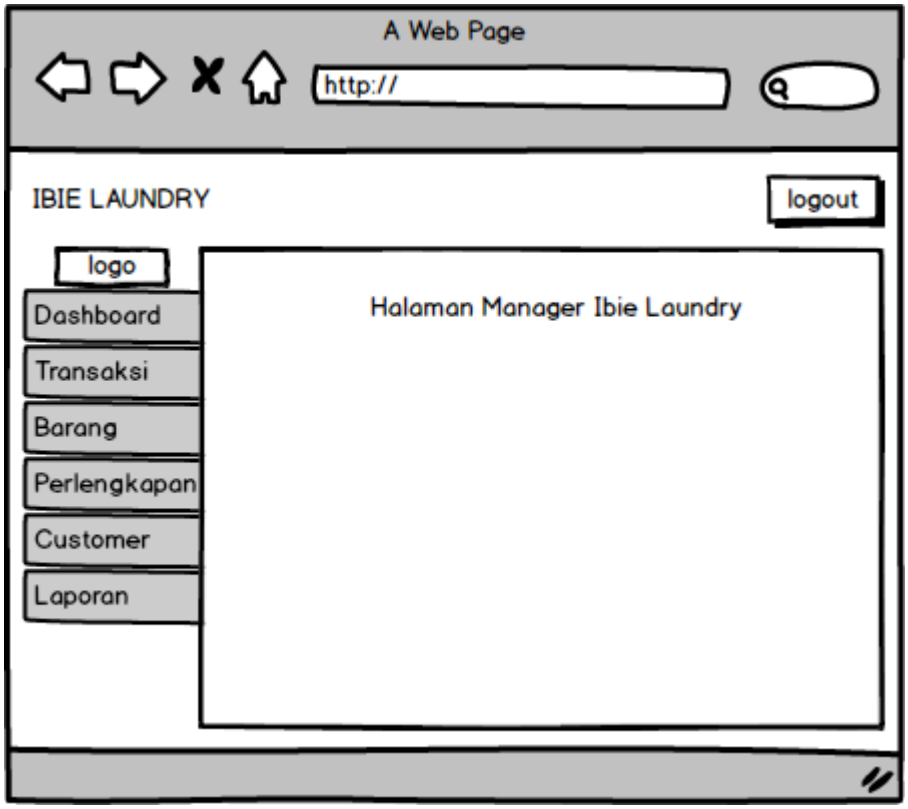
Gambar 4. 55 Rancangan halaman login

2. **Rancangan halaman dashboard**

Halaman dashboard merupakan halaman yang diberikan kepada pengguna setelah berhasil melakukan *login*. Rancangan dashboard *admin* dan *manager* dapat dilihat pada Gambar 4.56 dan 4.57.



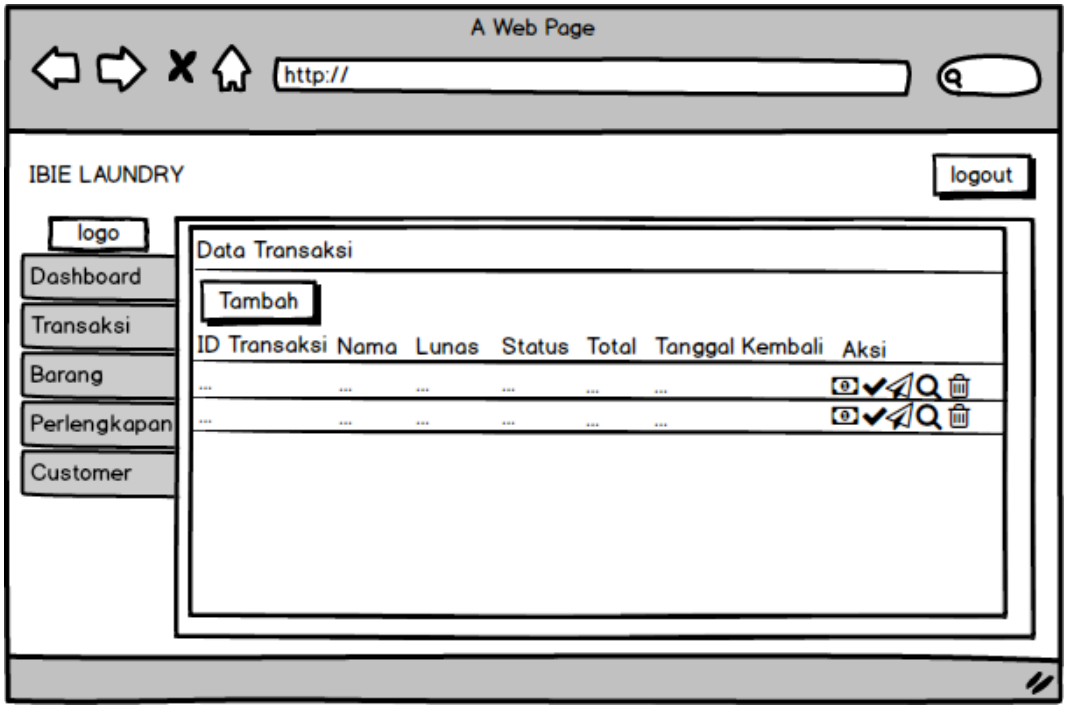
Gambar 4. 56 Rancangan halaman dashboard *admin*



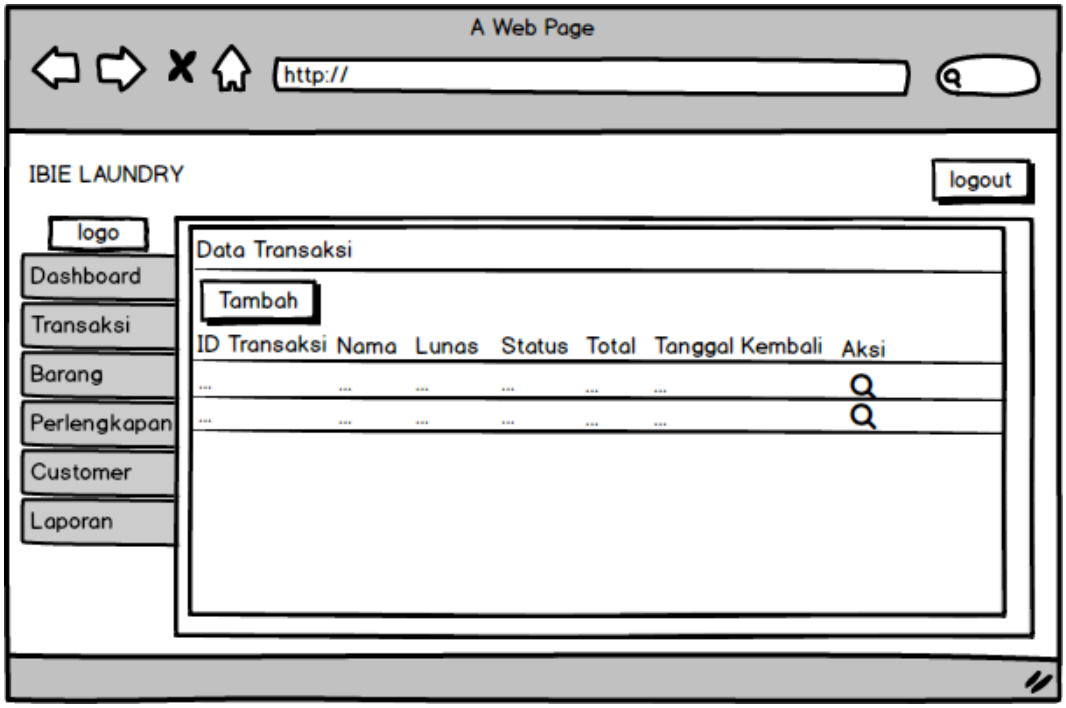
Gambar 4. 57 Rancangan halaman dashboard *manager*

3. Rancangan halaman transaksi

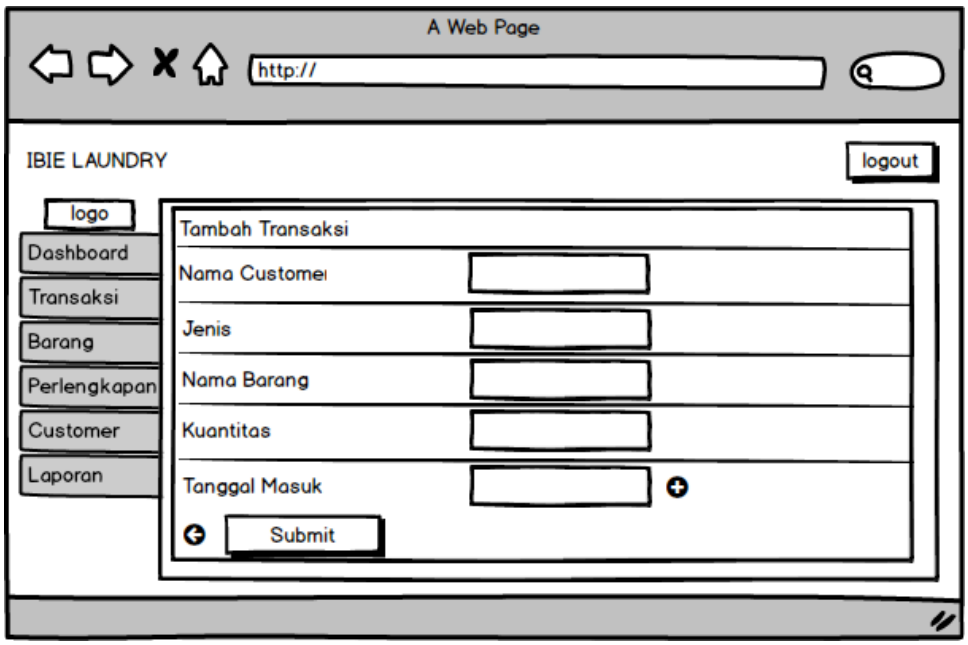
Halaman transaksi merupakan halaman yang berisi tentang data transaksi yang di kelola oleh *admin*. Pada halaman transaksi ini terdapat fitur tambah data, hapus data, lihat data, cetak data, update lunas, update status dan update tanggal kembali. Berbeda dengan halaman *admin*, halaman *manager* hanya mempunyai fitur untuk lihat data dan cetak data. Rancangan halaman transaksi dilihat pada Gambar 4.58, 4.59, 4.60, 4.61 dan 4.62.



Gambar 4. 58 Rancangan halaman transaksi *admin*

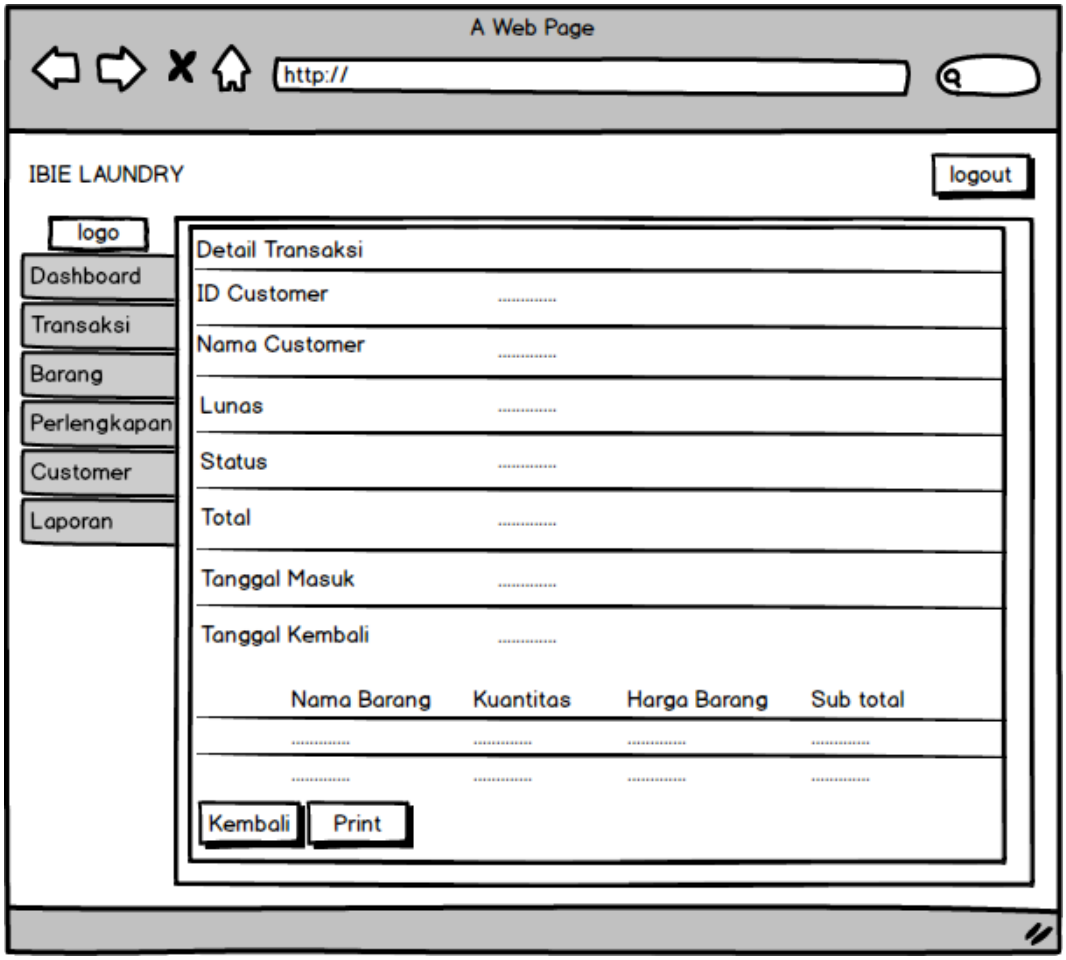


Gambar 4. 59 Rancangan halaman transaksi *manager*



Gambar 4. 60 Rancangan halaman tambah transaksi

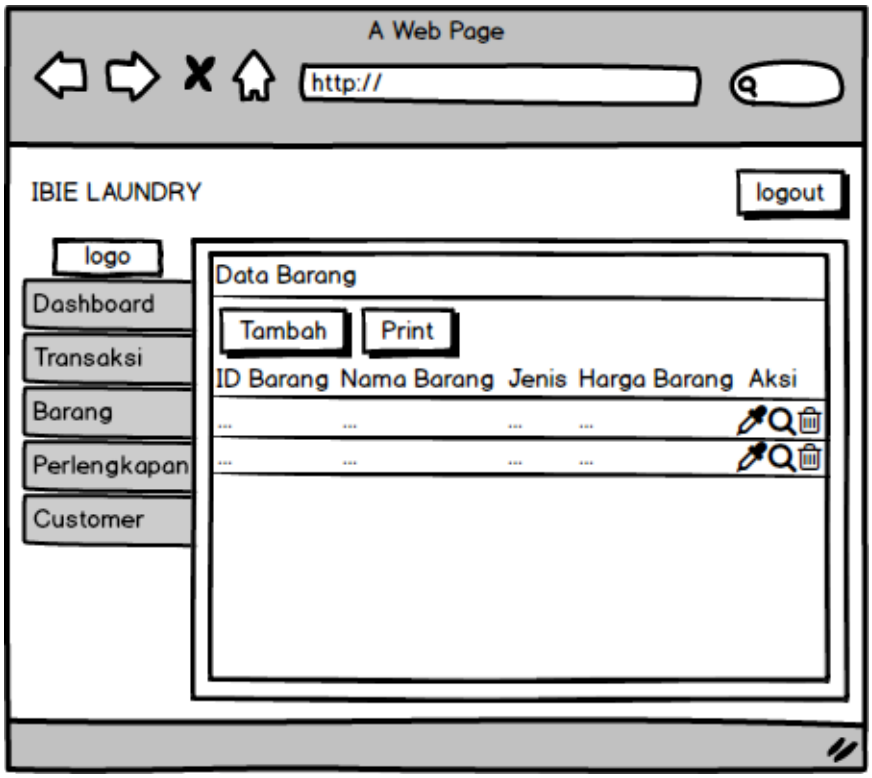
Gambar 4. 61 Rancangan halaman *detail* transaksi *admin*



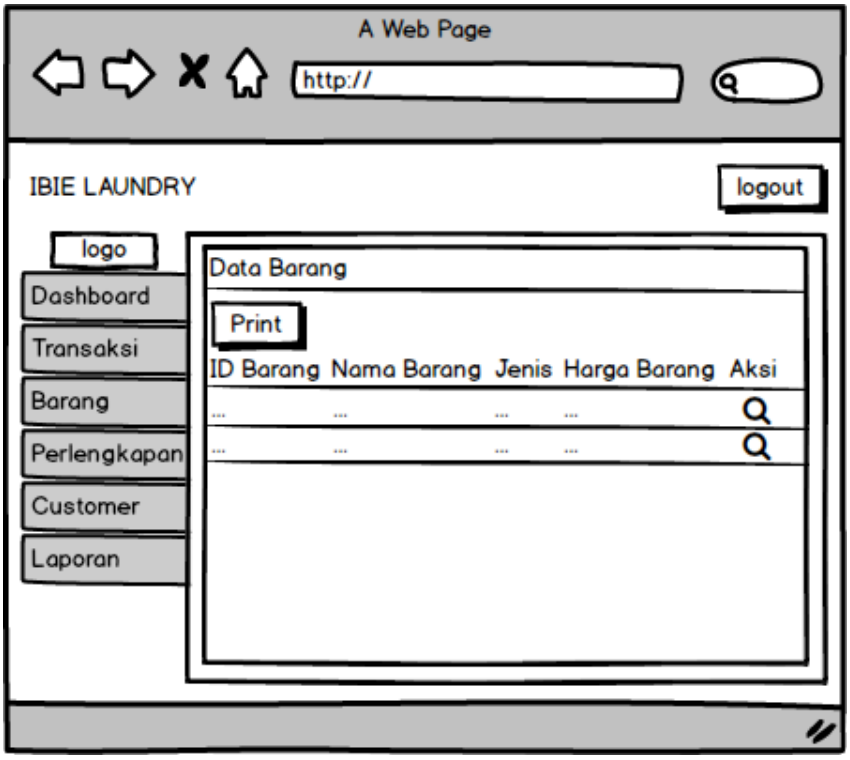
Gambar 4. 62 Rancangan halaman *detail* transaksi *manager*

4. Rancangan halaman barang

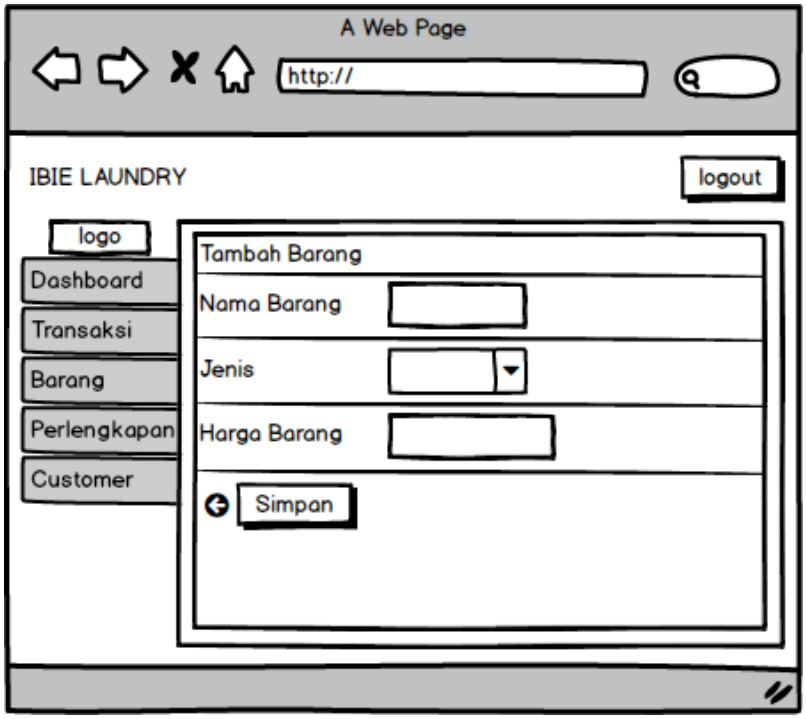
Halaman barang merupakan halaman yang berisi tentang data barang yang di didata oleh *admin*. Pada halaman barang ini terdapat fitur tambah data, ubah data, hapus data, lihat data, dan cetak data. Berbeda dengan halaman *admin*, halaman *manager* hanya mempunyai fitur untuk lihat data dan cetak data. Rancangan halaman transaksi dilihat pada Gambar 4.63, 4.64, 4.65, 4.67 dan 4.68.



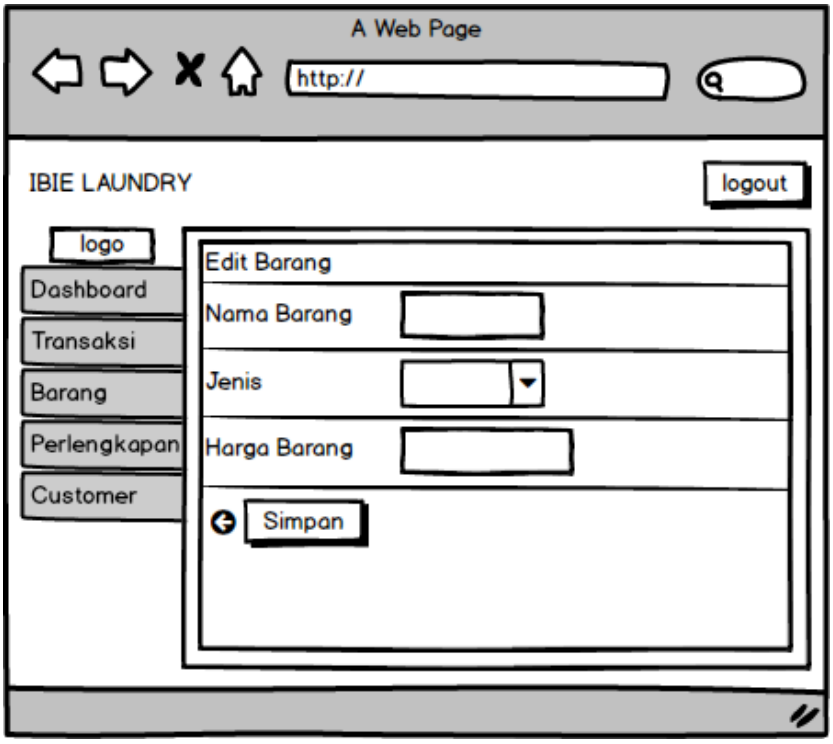
Gambar 4. 63 Rancangan halaman barang *admin*



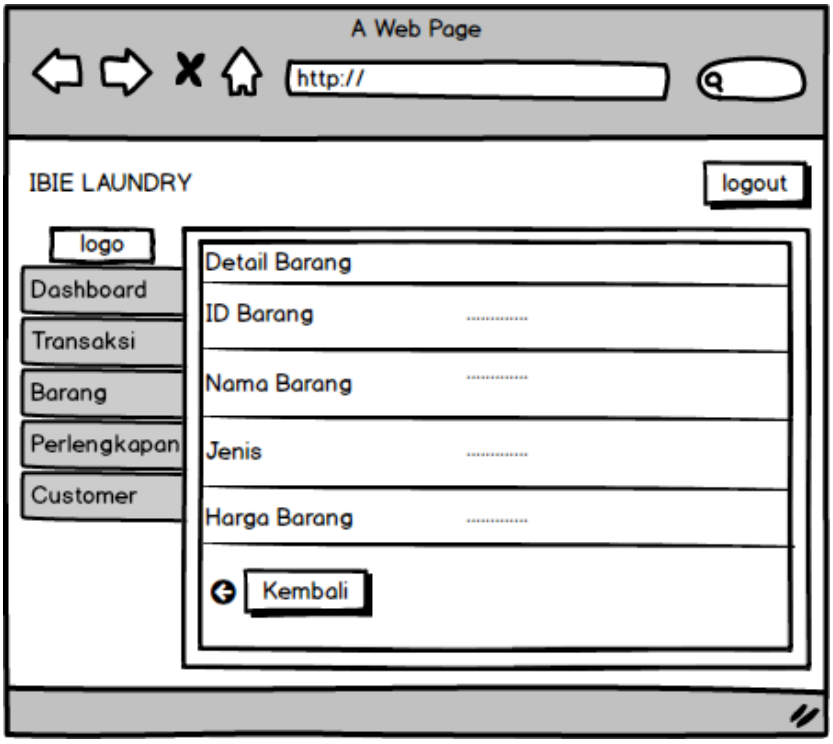
Gambar 4. 64 Rancangan halaman barang *manager*



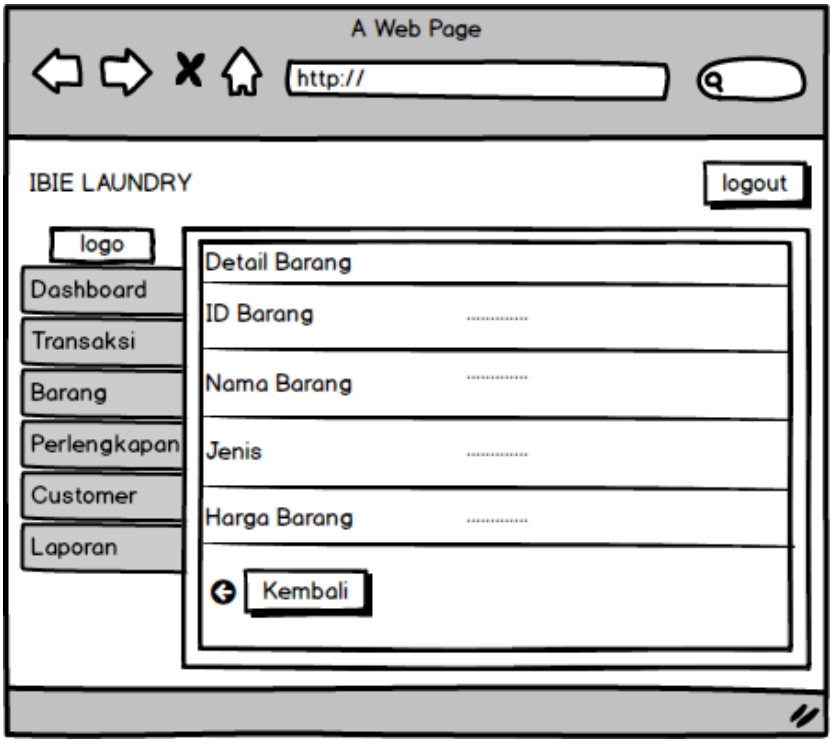
Gambar 4. 65 Rancangan halaman tambah barang



Gambar 4. 66 Rancangan halaman edit barang



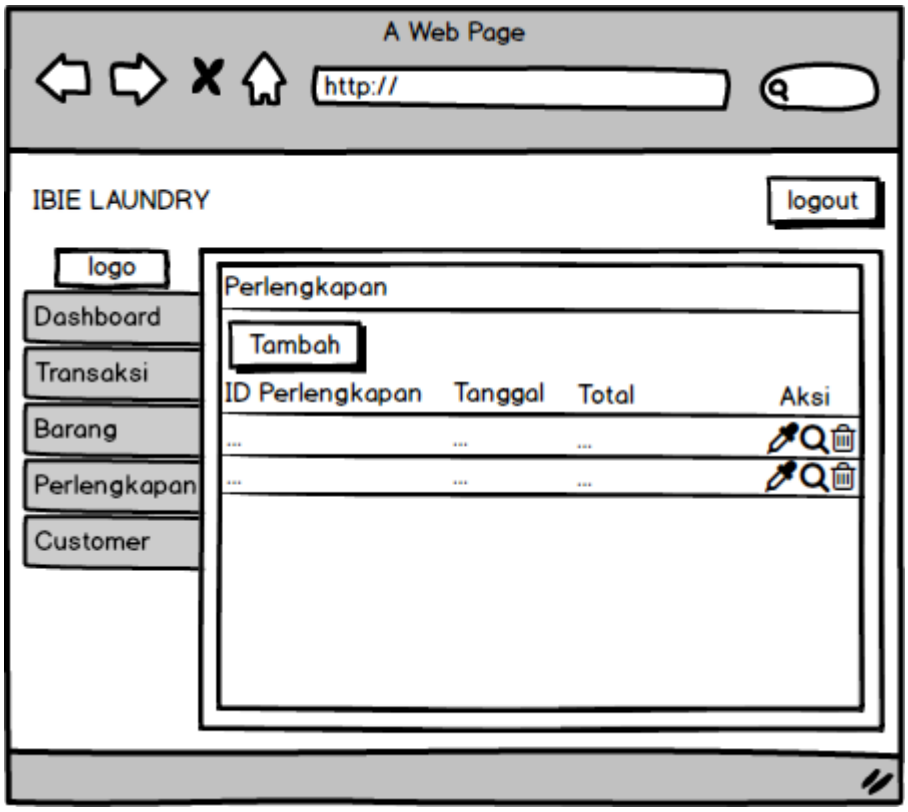
Gambar 4. 67 Rancangan halaman *detail* barang *admin*



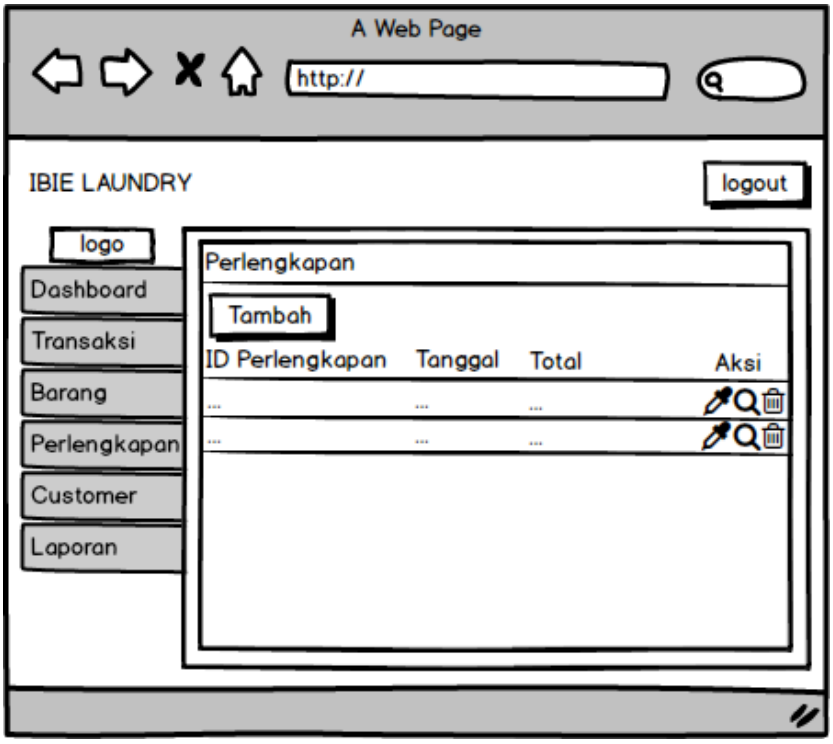
Gambar 4. 68 Rancangan halaman *detail* barang *manager*

5. **Rancangan halaman perlengkapan**

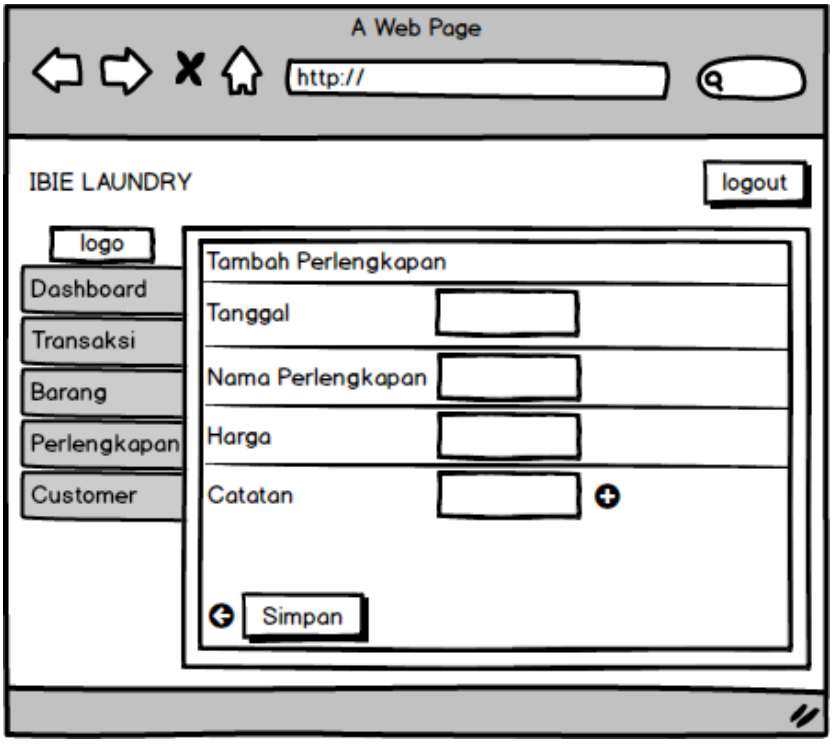
Halaman perlengkapan merupakan halaman yang berisi tentang pembelian perlengkapan yang di kelola oleh *admin* dan *manager*. Pada halaman perlengkapan ini terdapat fitur tambah data, hapus data, ubah data, lihat data dan cetak data. Rancangan halaman perlengkapan dilihat pada Gambar 4.69, 4.70, 4.71, 4.72, 4.73, 4.74, 4.75 dan 4.76.



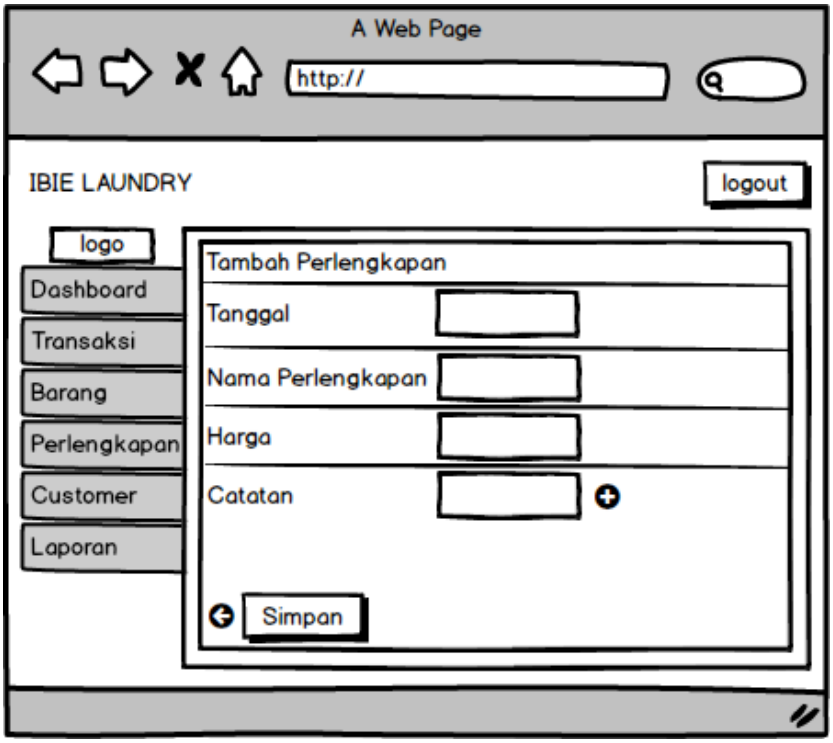
Gambar 4. 69 Rancangan halaman perlengkapan *admin*



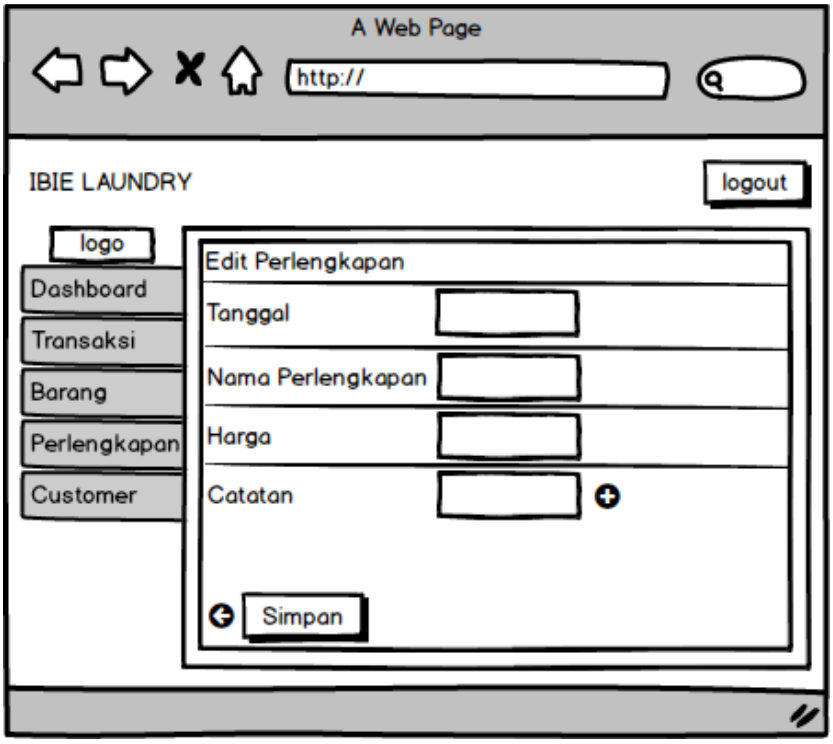
Gambar 4. 70 Rancangan halaman perlengkapan *manager*



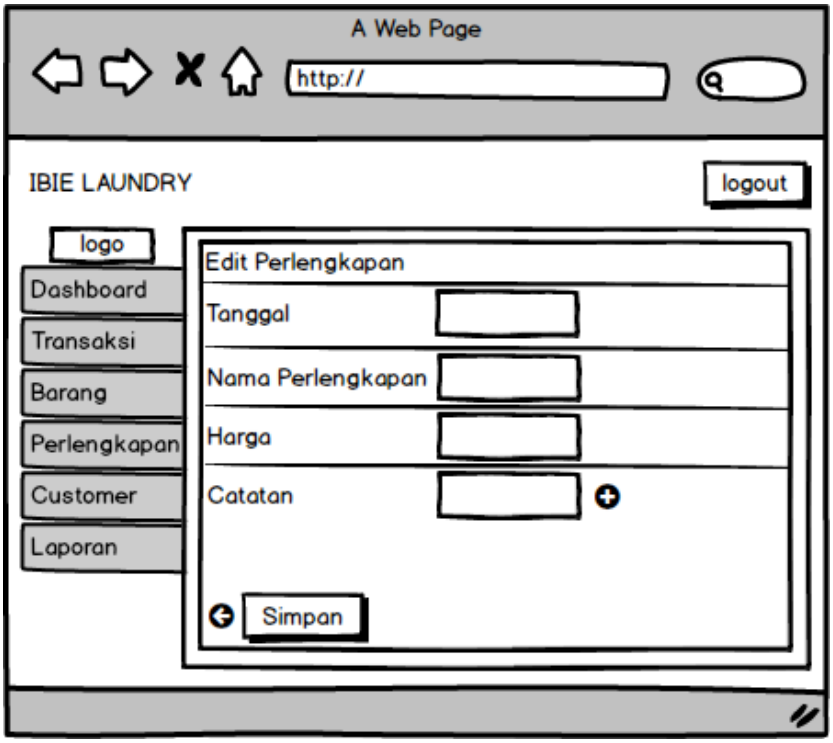
Gambar 4. 71 Rancangan halaman tambah perlengkapan *admin*



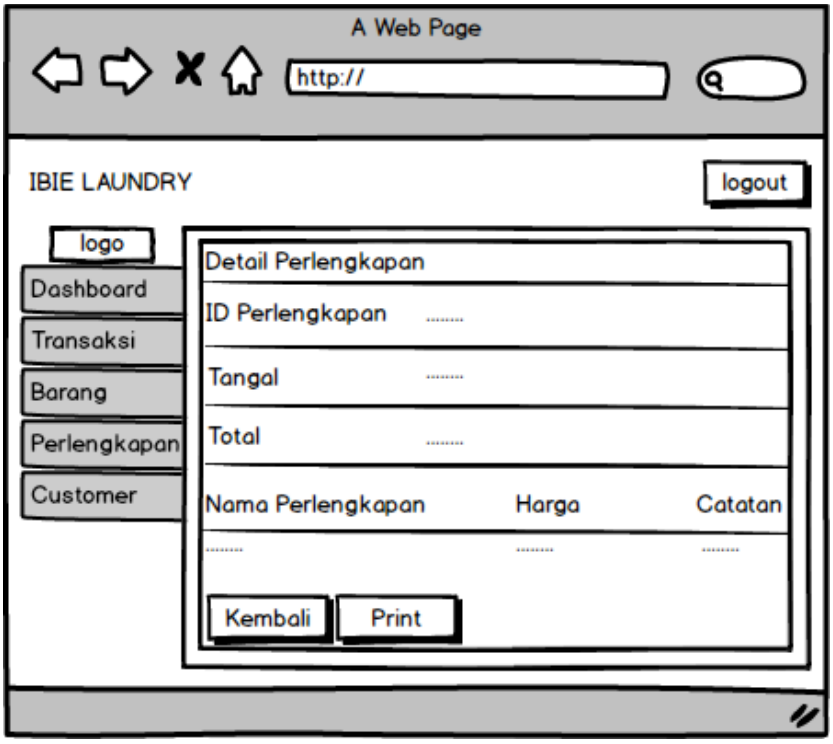
Gambar 4. 72 Rancangan halaman tambah perlengkapan *manager*



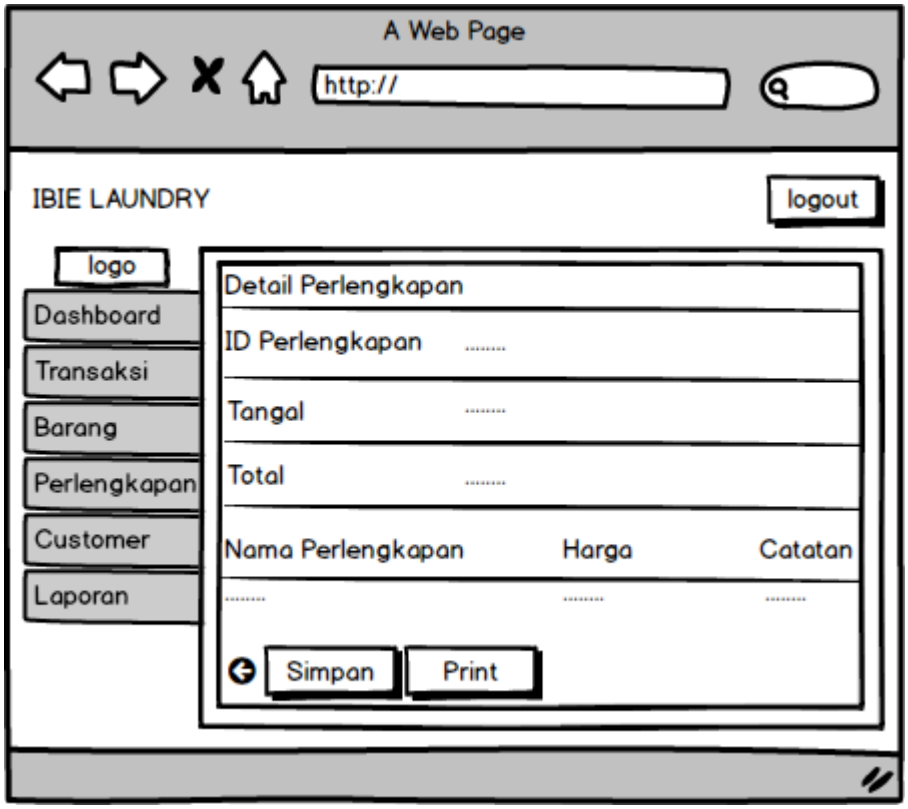
Gambar 4. 73 Rancangan halaman edit perlengkapan *admin*



Gambar 4. 74 Rancangan halaman edit perlengkapan *manager*



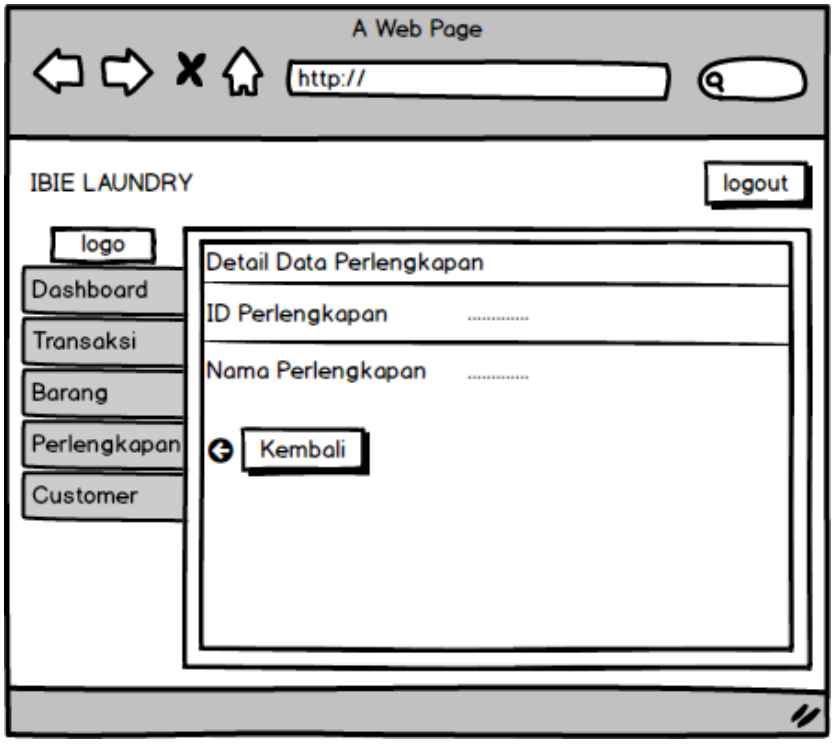
Gambar 4. 75 Rancangan halaman *detail* perlengkapan *admin*



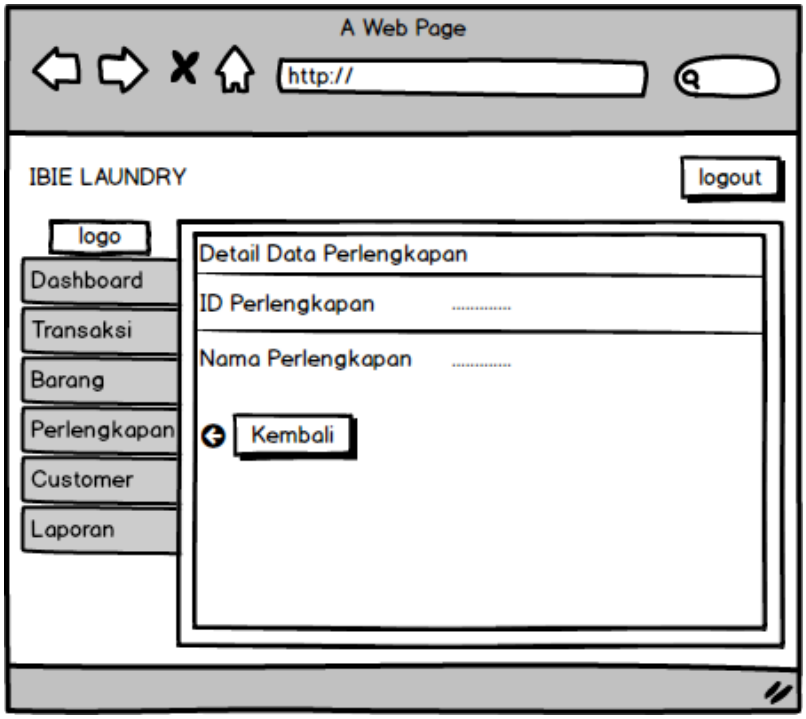
Gambar 4. 76 Rancangan halaman *detail* perlengkapan *manager*

6. **Rancangan halaman data perlengkapan**

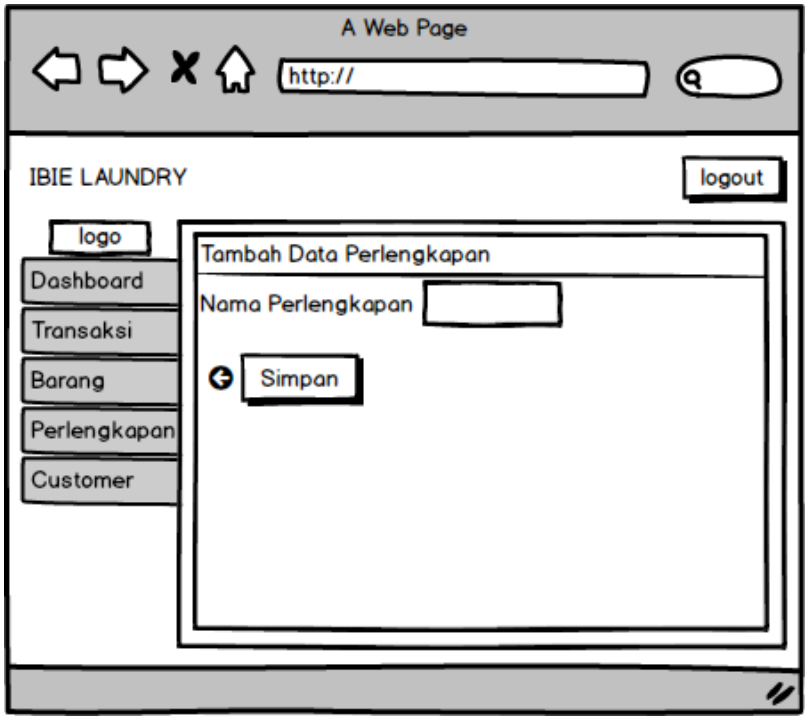
Halaman data perlengkapan merupakan halaman yang berisi tentang perlengkapan yang di didata oleh *admin* dan *manager*. Pada halaman data perlengkapan ini terdapat fitur tambah data, hapus data, ubah data, lihat data dan cetak data. Rancangan halaman perlengkapan dilihat pada Gambar 4.77, 4.78, 4.79, 4.80, 4.81, 4.82, 4.83 dan 4.84.



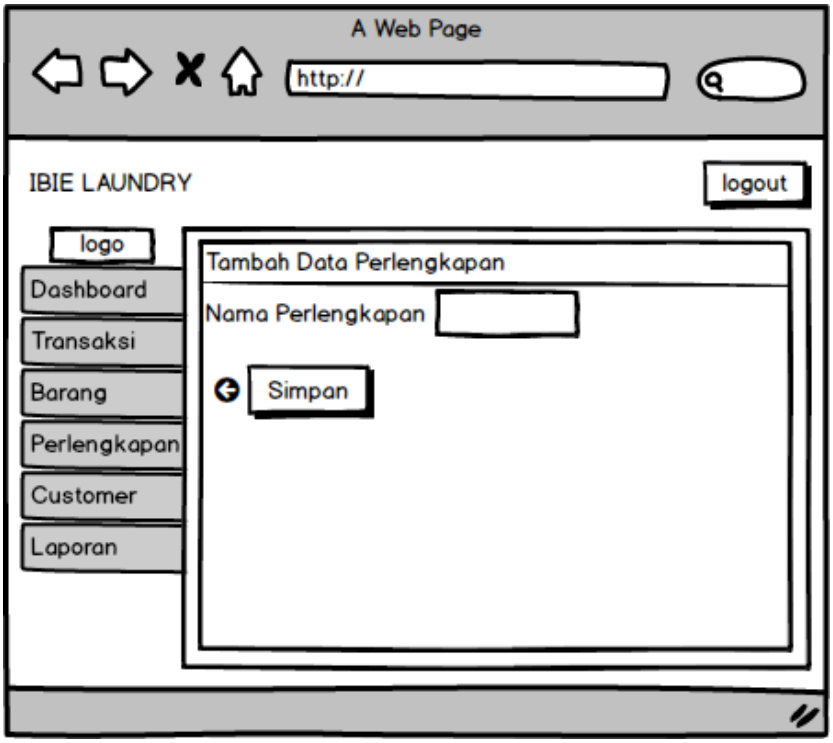
Gambar 4. 77 Rancangan halaman data perlengkapan *admin*



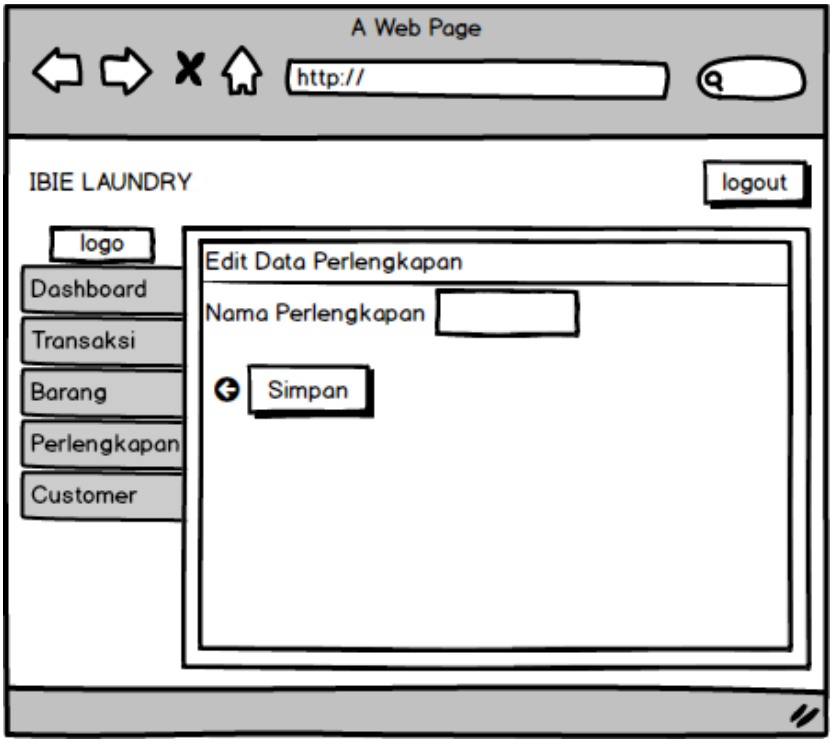
Gambar 4. 78 Rancangan halaman data perlengkapan *manager*



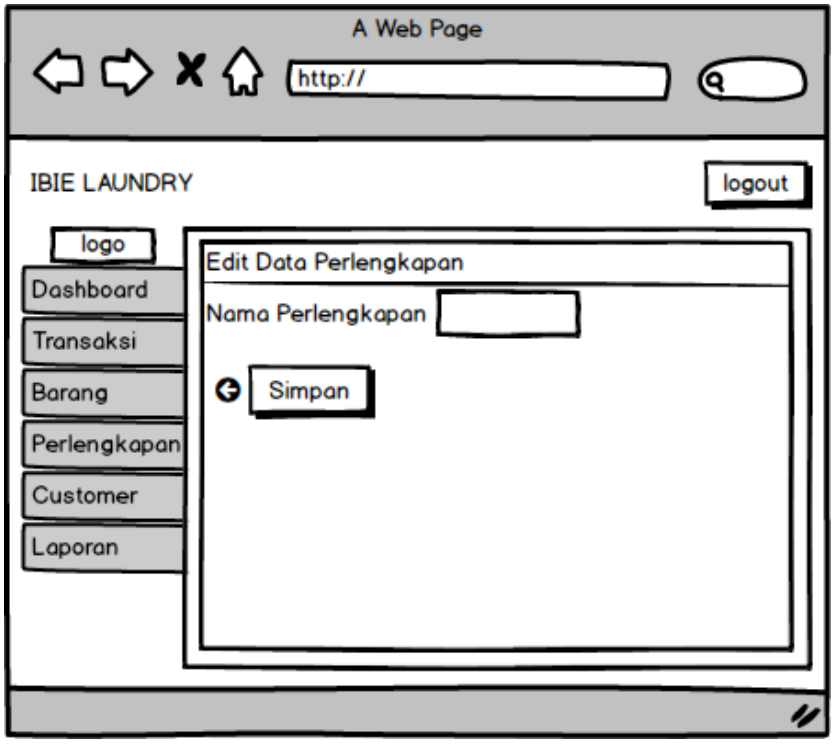
Gambar 4. 79 Rancangan halaman tambah data perlengkapan *admin*



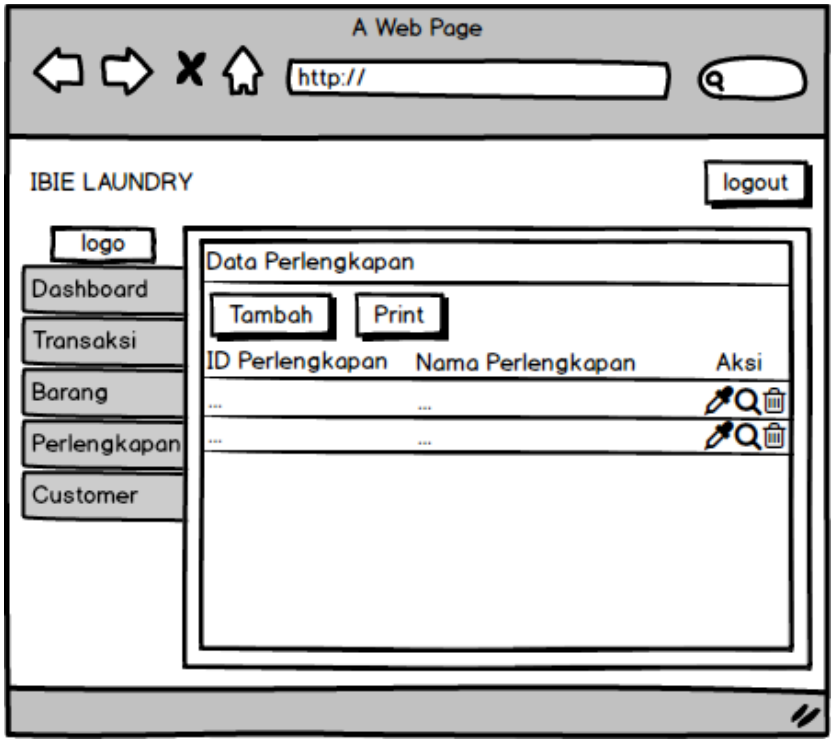
Gambar 4. 80 Rancangan halaman tambah data perlengkapan manage



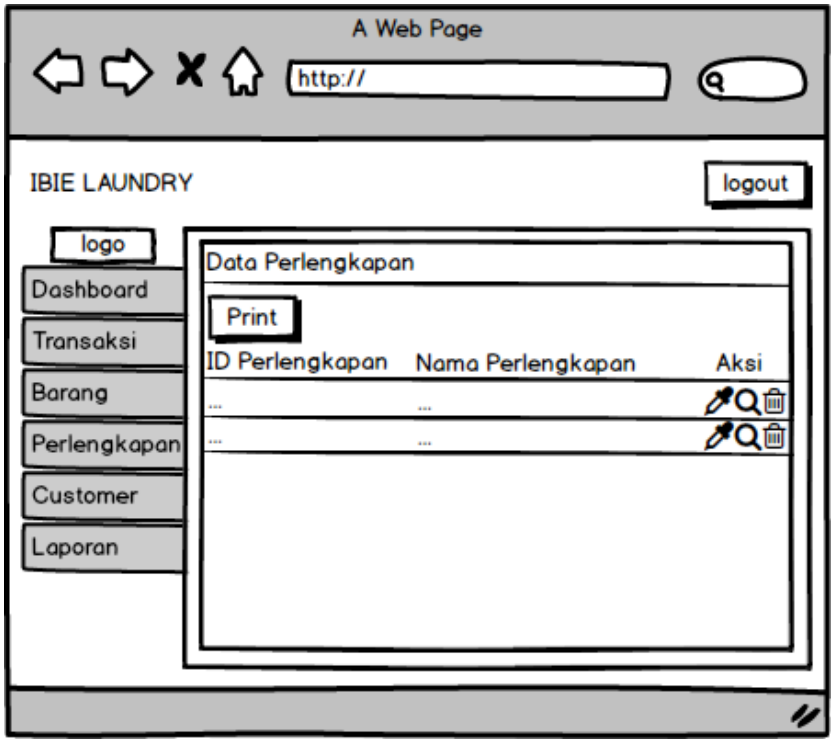
Gambar 4. 81 Rancangan halaman edit data perlengkapan *admin*



Gambar 4. 82 Rancangan halaman edit data perlengkapan *manager*



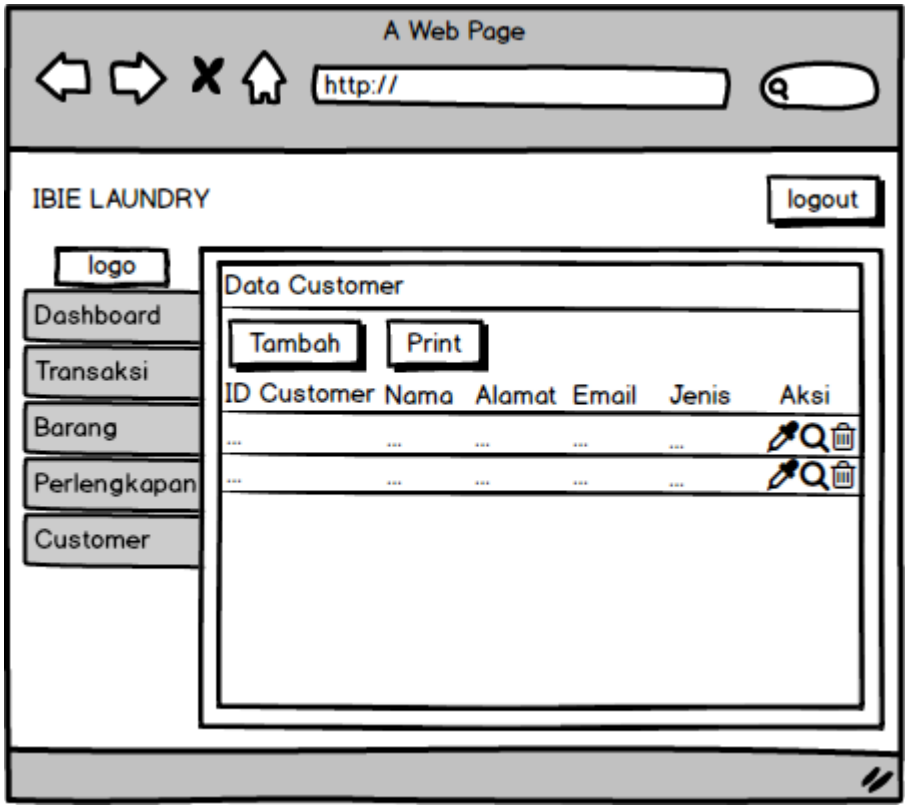
Gambar 4. 83 Rancangan halaman *detail* data perlengkapan admin



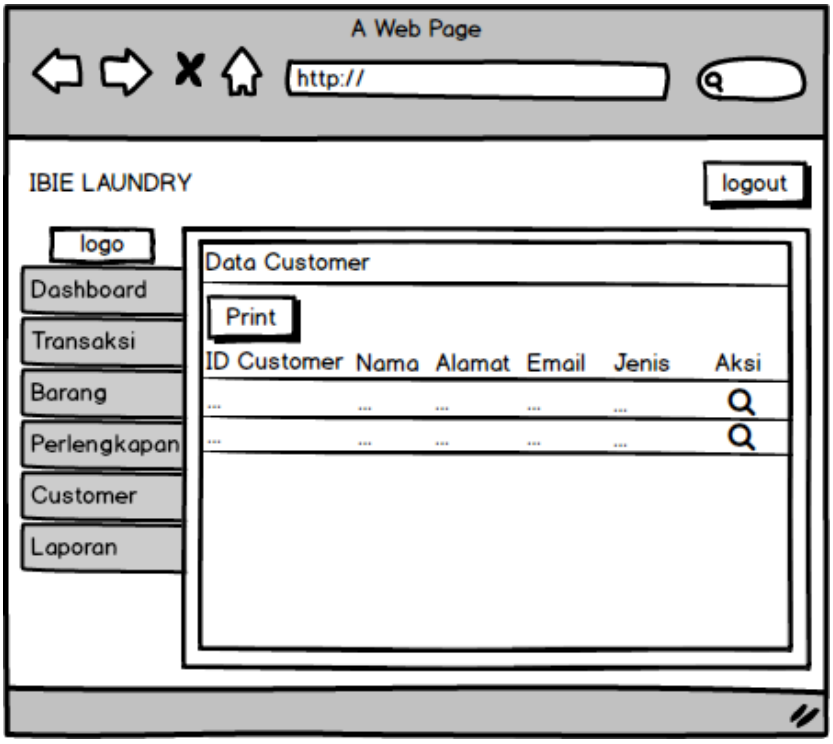
Gambar 4. 84 Rancangan halaman *detail* data perlengkapan *manager*

7. **Rancangan halaman *customer***

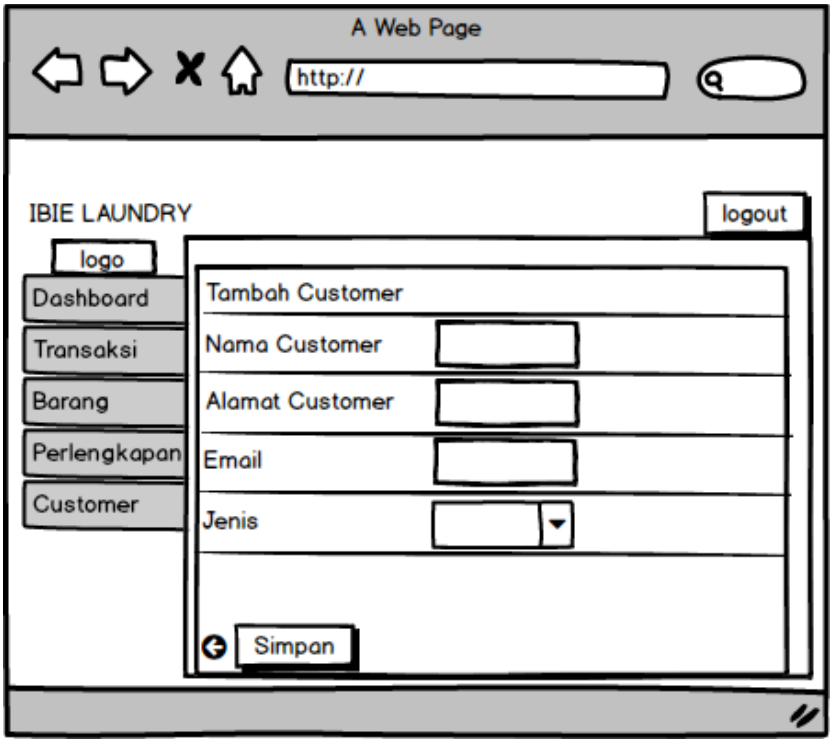
Halaman *customer* merupakan halaman yang berisi tentang *customer* yang di data oleh *admin*. Pada halaman *customer* ini terdapat fitur tambah data, ubah data, hapus data, lihat data, dan cetak data. Berbeda dengan halaman *admin*, halaman *manager* hanya mempunyai fitur untuk lihat data dan cetak data. Rancangan halaman *customer* dilihat pada Gambar 4.85, 4.86, 4.87, 4.88, 4.89 dan 4.90.



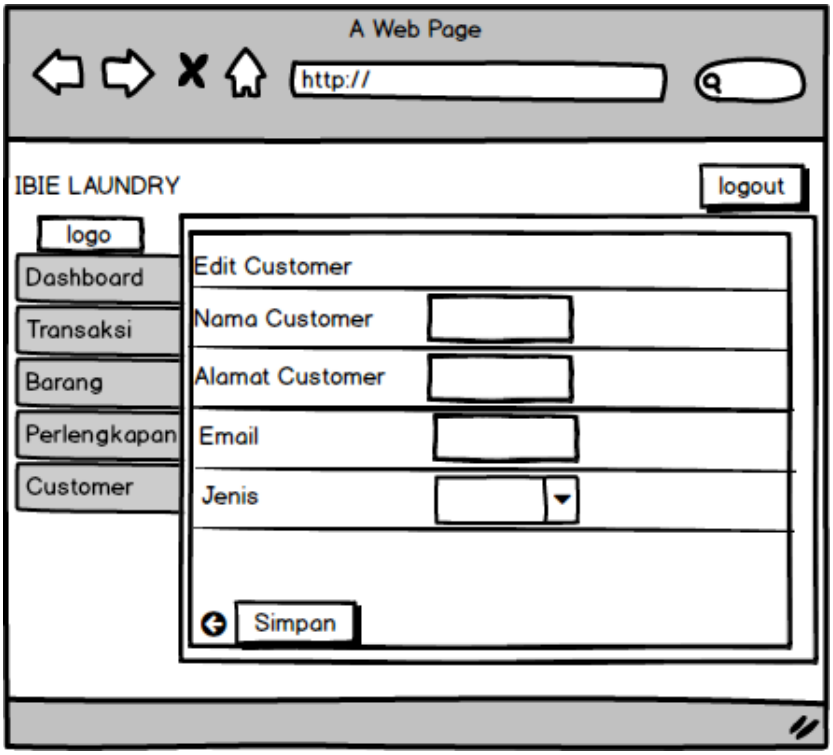
Gambar 4. 85 Rancangan halaman *customer admin*



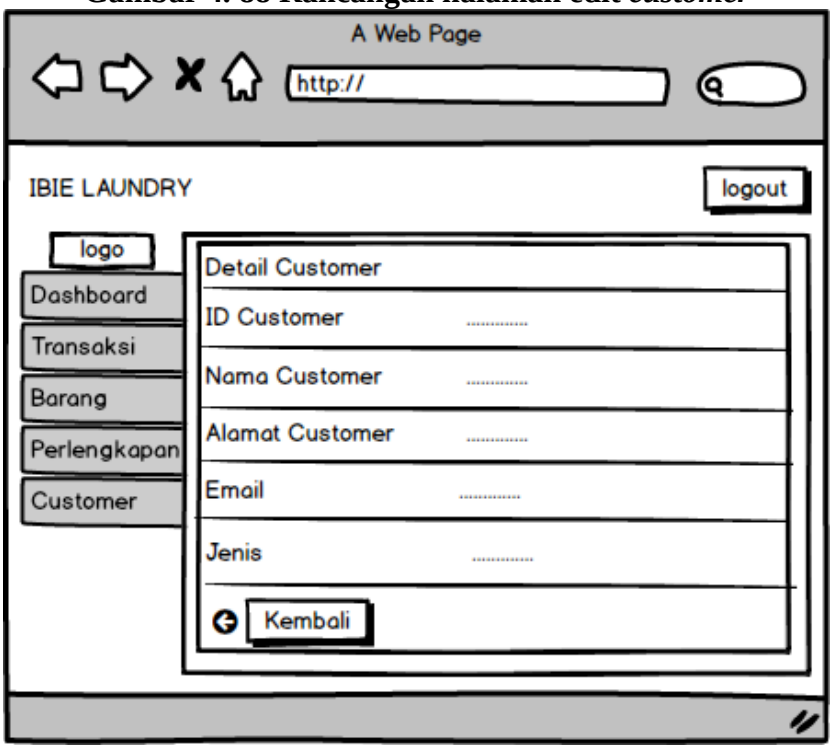
Gambar 4. 86 Rancangan halaman *customer manager*



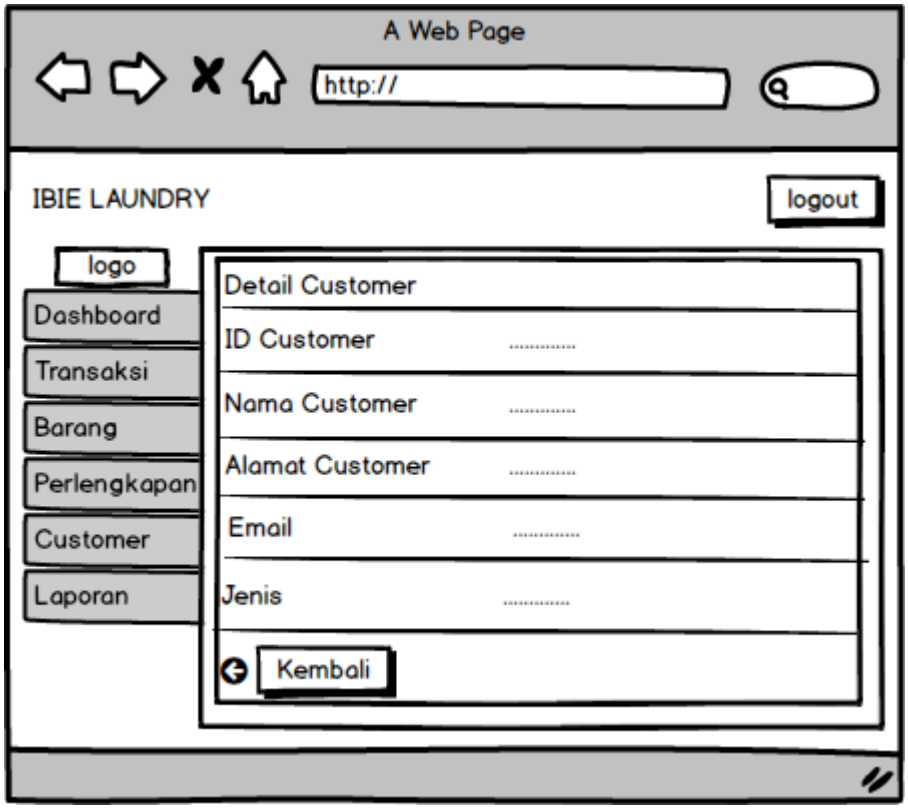
Gambar 4. 87 Rancangan halaman tambah *customer*



Gambar 4. 88 Rancangan halaman edit *customer*



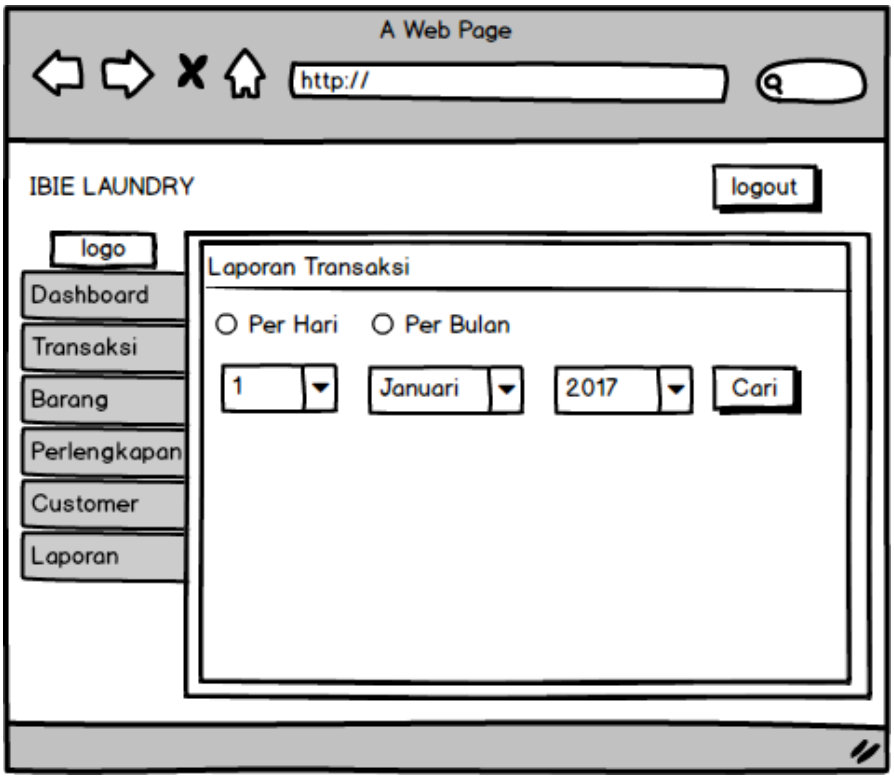
Gambar 4. 89 Rancangan halaman *detail customer admin*



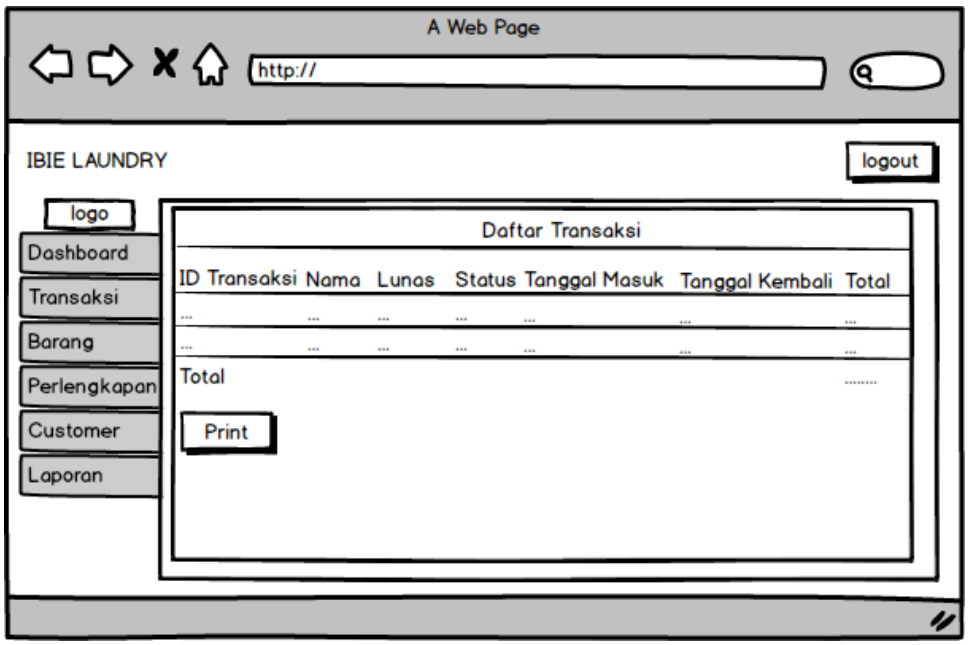
Gambar 4. 90 Rancangan halaman *detail customer manager*

8. Rancangan halaman laporan

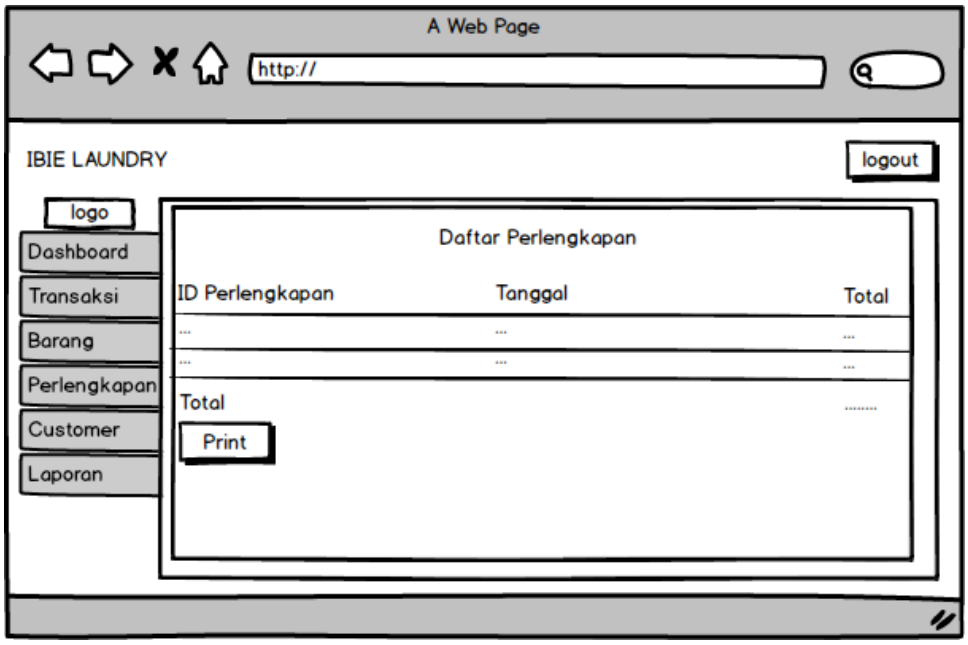
Halaman laporan merupakan halaman yang berisi tentang laporan *omzet*, laporan pengeluaran dan laporan *profit* yang di telah dikelola sebelumnya. Pada halaman laporan ini terdapat fitur memilih jangka waktu laporan, yaitu per-hari atau per-bulan dan terdapat fitur untuk cetak data. Rancangan halaman laporan dilihat pada Gambar 4.91, 4.92, 4.93 dan 4.94.



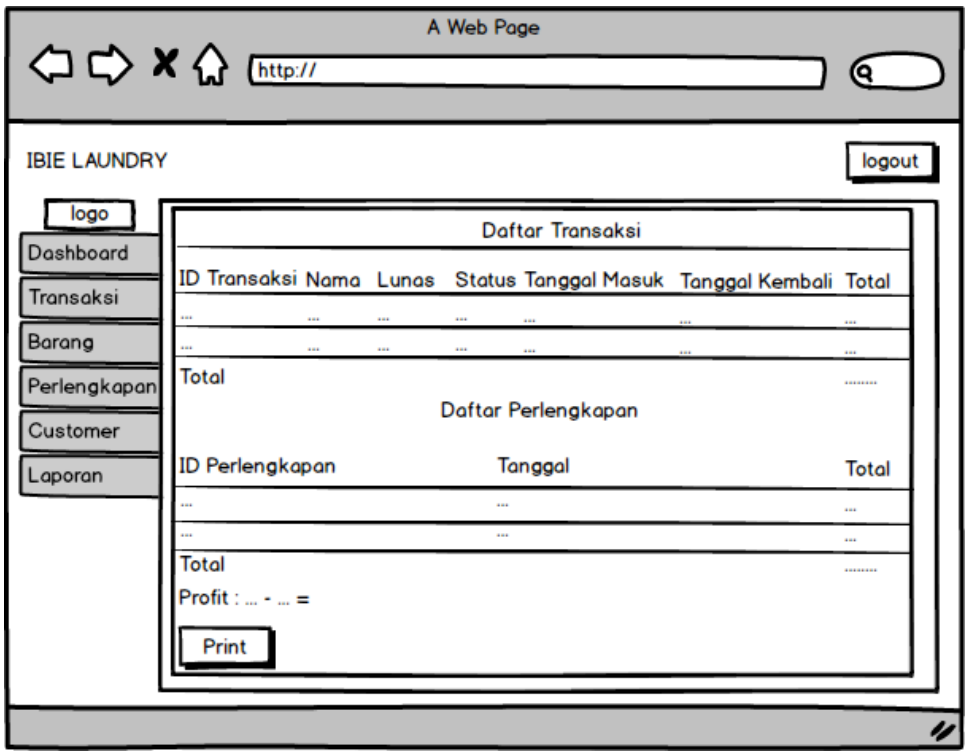
Gambar 4. 91 Rancangan halaman jangka waktu laporan



Gambar 4. 92 Rancangan halaman laporan *omzet*



Gambar 4. 93 Rancangan halaman laporan pengeluaran



Gambar 4. 94 Rancangan halaman laporan *profit*

BAB V
IMPLEMENTASI SISTEM

5.1 Spesifikasi Sistem

Spesifikasi perangkat lunak yang digunakan untuk membangun system informasi pengolahan transaksi laundry pada *Ibie Laundry & Wet Cleaning* sesuai dengan rancangan bab sebelumnya adalah sebagai berikut:

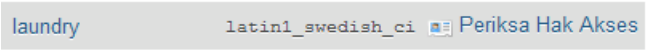
1. Sistem Operasi : Windows 7 Profesional 32 bit
2. Bahasa Pemrograman : PHP dan HTML
3. Framework : Code Igniter
4. Tool : Sublime Text
5. Local Server : Apache 3.2.1
6. DBMS : Mysql
7. Browser : Google Crome

Spesifikasi perangkat keras yang digunakan untuk membangun system informasi pengolahan transaksi laundry pada *Ibie Laundry & Wet Cleaning* antara lain:

1. Intel CORE(TM) i5-2410M @ 2.30 GHz
2. RAM 2 GB
3. HARDISK 500 GB

5.2 Implementasi Database

Dalam penelitian ini, penulis menggunakan database ‘laundry’ sebagai basisdata yang digunakan. Terdapat tabel, atribut dan relasi antar tabel dalam basisdata ‘laundry’. Implementasi basisdata laundry dapat dilihat pada Gambar 5.1



Gambar 5. 1 Implementasi Database laundry

Langkah selanjutnya adalah membuat tabel. Tabel-tabel yang akan dibuat adalah tabel barang, tabel *customer*, tabel data perlengkapan, tabel *detail* perlengkapan, tabel *detail* transaksi, tabel perlengkapan, tabel transaksi dan tabel user. Berikut merupakan sintaks SQL untuk membuat tabel-tabel tersebut:

1. Tabel User

Tabel user digunakan untuk menyimpan data pengguna sistem. Implementasi tabel user dapat dilihat pada Gambar 5.2

#	Nama	Jenis	Penyortiran	Atribut	Kosong	Bawaan	Ekstra
1	**id_username**	int(11)			Tidak	*Tidak ada*	AUTO_INCREMENT
2	**username**	varchar(20)	latin1_swedish_ci		Tidak	*Tidak ada*	
3	**password**	varchar(20)	latin1_swedish_ci		Tidak	*Tidak ada*	
4	**level**	varchar(20)	latin1_swedish_ci		Tidak	*Tidak ada*	

Gambar 5. 2 Implementasi tabel user

2. Tabel Barang

Tabel barang digunakan untuk menyimpan data barang. Implementasi tabel barang dapat dilihat pada Gambar 5.3

#	Nama	Jenis	Penyortiran	Atribut	Kosong	Bawaan	Ekstra
1	**id_barang**	int(11)			Tidak	*Tidak ada*	AUTO_INCREMENT
2	**nama_barang**	varchar(20)	latin1_swedish_ci		Tidak	*Tidak ada*	
3	**jenis**	varchar(20)	latin1_swedish_ci		Tidak	*Tidak ada*	
4	**harga_barang**	double			Tidak	*Tidak ada*	

Gambar 5. 3 Implementasi tabel barang

3. Tabel *Customer*

Tabel *customer* digunakan untuk menyimpan data *customer*. Implementasi tabel *customer* dapat dilihat pada Gambar 5.4

#	Nama	Jenis	Penyortiran	Atribut	Kosong	Bawaan	Ekstra
1	id_customer	int(11)			Tidak	*Tidak ada*	AUTO_INCREMENT
2	nama_customer	varchar(20)	latin1_swedish_ci		Tidak	*Tidak ada*	
3	alamat_customer	varchar(50)	latin1_swedish_ci		Tidak	*Tidak ada*	
4	email	varchar(20)	latin1_swedish_ci		Tidak	*Tidak ada*	
5	jenis	varchar(20)	latin1_swedish_ci		Tidak	*Tidak ada*	

Gambar 5. 4 Implementasi tabel *customer*

4. Tabel Data Perlengkapan

Tabel data perlengkapan digunakan untuk menyimpan data perlengkapan. Implementasi tabel data perlengkapan dapat dilihat pada Gambar 5.5

#	Nama	Jenis	Penyortiran	Atribut	Kosong	Bawaan	Ekstra
1	id_data_perlengkapan	int(11)			Tidak	*Tidak ada*	AUTO_INCREMENT
2	nama_perlengkapan	varchar(20)	latin1_swedish_ci		Tidak	*Tidak ada*	

Gambar 5. 5 Implementasi tabel data perlengkapan

5. Tabel *Detail* Perlengkapan

Tabel *detail* perlengkapan digunakan untuk menyimpan *detail* pembelian perlengkapan. Implementasi tabel *detail* perlengkapan dapat dilihat pada Gambar 5.6

#	Nama	Jenis	Penyortiran	Atribut	Kosong	Bawaan	Ekstra
1	id_detail_perlengkapan	int(11)			Tidak	*Tidak ada*	AUTO_INCREMENT
2	id_perlengkapan	int(11)			Tidak	*Tidak ada*	
3	nama_perlengkapan	varchar(20)	latin1_swedish_ci		Tidak	*Tidak ada*	
4	tanggal	datetime			Tidak	CURRENT_TIMESTAMP	
5	harga	double			Tidak	*Tidak ada*	
6	catatan	varchar(20)	latin1_swedish_ci		Tidak	*Tidak ada*	

Gambar 5. 6 Implementasi tabel *detail* perlengkapan

6. Tabel Perlengkapan

Tabel perlengkapan digunakan untuk menyimpan data pembelian perlengkapan. Implementasi tabel user perlengkapan dilihat pada Gambar 5.7

#	Nama	Jenis	Penyortiran	Atribut	Kosong	Bawaan	Ekstra
1	id_perlengkapan	int(11)			Tidak	*Tidak ada*	AUTO_INCREMENT
2	tanggal	datetime			Tidak	CURRENT_TIMESTAMP	
3	total	double			Tidak	*Tidak ada*	

Gambar 5. 7 Implementasi tabel perlengkapan

7. Tabel *Detail* Transaksi

Tabel *detail* transaksi digunakan untuk menyimpan *detail* transaksi. Implementasi tabel *detail* transaksi dapat dilihat pada Gambar 5.8

#	Nama	Jenis	Penyortiran	Atribut	Kosong	Bawaan	Ekstra
1	id_detail_transaksi	int(11)			Tidak	*Tidak ada*	AUTO_INCREMENT
2	id_transaksi	int(11)			Tidak	*Tidak ada*	
3	nama_barang	varchar(20)	latin1_swedish_ci		Tidak	*Tidak ada*	
4	kuantitas	int(3)			Tidak	*Tidak ada*	
5	harga_barang	double			Tidak	*Tidak ada*	
6	sub_total	double			Tidak	*Tidak ada*	

Gambar 5. 8 Implementasi tabel *detail* transaksi

8. Tabel Transaksi

Tabel transaksi digunakan untuk menyimpan data transaksi. Implementasi tabel transaksi dapat dilihat pada Gambar 5.9

#	Nama	Jenis	Penyortiran	Atribut	Kosong	Bawaan	Ekstra
1	id_transaksi	int(11)			Tidak	*Tidak ada*	AUTO_INCREMENT
2	nama_customer	varchar(20)	latin1_swedish_ci		Tidak	*Tidak ada*	
3	jenis	varchar(20)	latin1_swedish_ci		Tidak	*Tidak ada*	
4	total	double			Tidak	*Tidak ada*	
5	lunas	varchar(20)	latin1_swedish_ci		Tidak	Belum lunas	
6	status	varchar(20)	latin1_swedish_ci		Tidak	Sedang dikerjakan	
7	tanggal_masuk	datetime			Tidak	CURRENT_TIMESTAMP	
8	tanggal_kembali	datetime			Tidak	*Tidak ada*	

Gambar 5. 9 Implementasi tabel transaksi

5.3 Implementasi Antarmuka Proses

Implementasi antarmuka menggambarkan penjelasan mengenai hasil implementasi sistem yang terkait dengan cuplikan tampilan dan juga potongan *source code* dari masing-masing *form* atau proses pada sistem yang telah dibangun.

5.3.1 Koneksi basis data

Pada *framework* code igniter terdapat sebuah *file* yang digunakan untuk mengatur koneksi dari aplikasi ke basis data, *file* tersebut bernama *Database.php*. *Source code* untuk konfigurasi koneksi *database* mysql tersebut dapat dilihat pada Gambar 5.10.

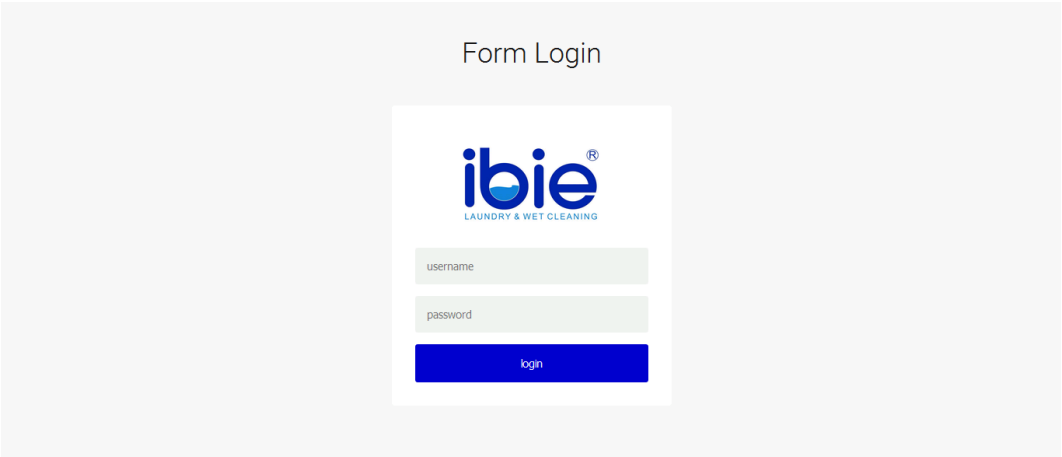
```
$active_group = 'default';
$active_record = TRUE;

$db['default']['hostname'] = 'localhost';
$db['default']['username'] = 'root';
$db['default']['password'] = '';
$db['default']['database'] = 'laundry';
$db['default']['dbdriver'] = 'mysqli';
$db['default']['dbprefix'] = '';
$db['default']['pconnect'] = TRUE;
$db['default']['db_debug'] = TRUE;
$db['default']['cache_on'] = FALSE;
$db['default']['cachedir'] = '';
$db['default']['char_set'] = 'utf8';
$db['default']['dbcollat'] = 'utf8_general_ci';
$db['default']['swap_pre'] = '';
$db['default']['autoinit'] = TRUE;
$db['default']['stricton'] = FALSE;
```

Gambar 5. 10 Source code koneksi database

5.3.2 Halaman login

Halaman *login* merupakan halaman awal ketika pengguna membuka sistem. Pada halaman ini pengguna memasukkan *username* dan *password* untuk dapat masuk ke sistem. Implementasi halaman login dapat dilihat pada Gambar 5.11.



Gambar 5. 11 Implementasi halaman login

Sedangkan pada Gambar 5.12 merupakan potongan kode program yang melakukan aksi menampilkan halaman login.

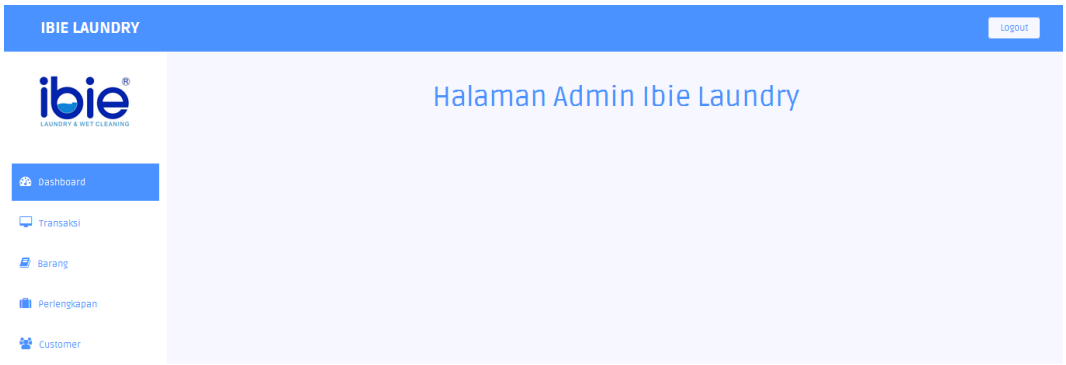
```
1  public function cek_login() {
2          $username =$this->input->post('username', TRUE);
3          $password = $this->input->post('password', TRUE);
4          $hasil = $this->m_user->cek_user($username, $password);
5
6          if ($hasil != FALSE) {
7              $session_data = array(
8                              'id' => $hasil[0]['id_username'],
9                              'username' => $hasil[0]['username'],
10                             'level' => $hasil[0]['level'],
11                             );
12
13             $this->session->set_userdata('logged_in', $session_data);
14             $userdata = $this->session->userdata('logged_in');
15             if($session_data['level']=='admin'){
16                 redirect('admin/c_admin');
17             }
18             else {
19                 redirect('manager/c_manager');
20             }
21         }
22         else {
23             echo "<script>alert('Gagal login: Cek username, password!');history.go(-1);</script>";
24         }
25     }
```

Gambar 5. 12 Potongan kode login

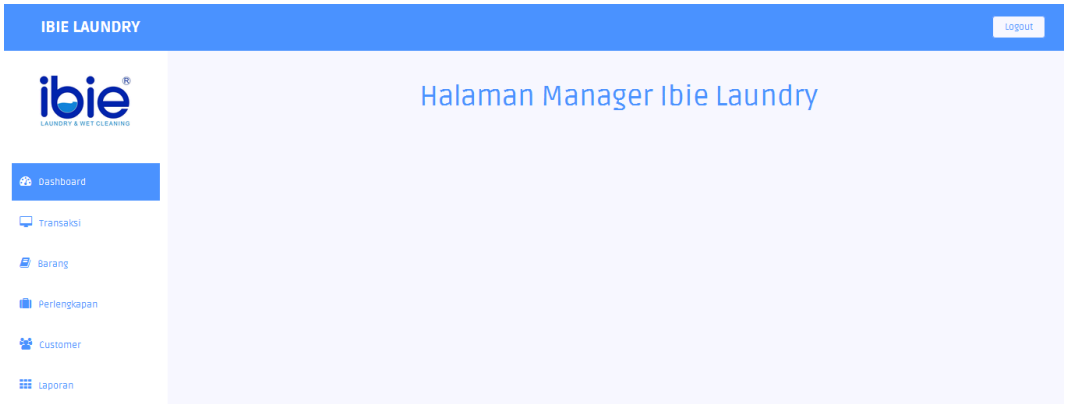
Untuk keterangan potongan kode diatas, pada baris ke 15 sampai 19 sistem memisahkan level pengguna. Apabila level *admin*, maka akan diarahkan ke controller *admin*, sedangkan jika level pengguna adalah *manager*, maka akan ke arahkan ke controller *manager*.

5.3.3 Halaman dashboard

Halaman dashboard merupakan halaman yang pertama kali muncul pada saat pengguna berhasil masuk kedalam sistem. Implementasi halaman dashboard *admin* dan *manager* dapat dilihat pada Gambar 5.13 dan 5.14



Gambar 5. 13 Implementasi halaman dashboard *admin*



Gambar 5. 14 Implementasi halaman dashboard *manager*

pada Gambar 5.15 dan 5.16 merupakan potongan kode untuk menampilkan halaman dashboard *admin* dan *manager*

```
1  public function index() {
2          $data['active'] = 'dashboard';
3           $data['nama_pengguna'] = $this ->session->userdata('id_user');
4          $this->template->load('sys_template','admin/index', $data);
5      }
```

Gambar 5. 15 Potongan kode halaman dashboard *admin*

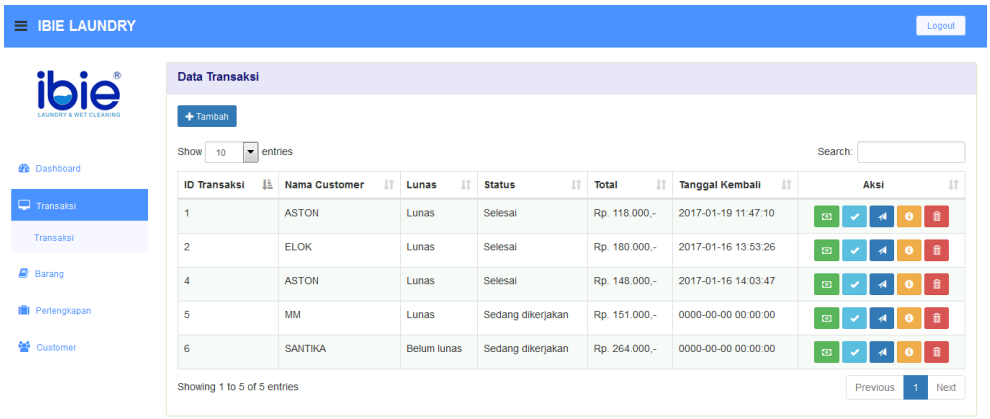
```
1  public function index() {
2          $data['active'] = 'dashboard';
3          $data['nama_pengguna'] = $this ->session->userdata('id_user');
4          $this->template->load('sys_template_manager','manager/index', $data);
5      }
```

Gambar 5. 16 Potongan kode halaman dashboard *manager*

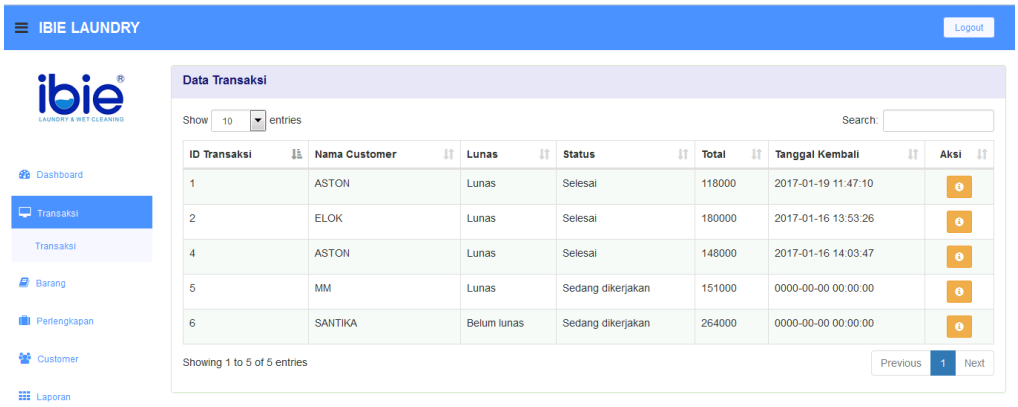
Untuk keterangan gambar diatas, merupakan kode berhasil mengakses halaman *admin*. Pada baris 3 terdapat judul halaman yang berarti pengguna sedang berada di halaman *admin* atau *manager*. Sedangkan untuk baris 4 merupakan kode untuk mengarahkan ke halaman dashboard *admin* atau *manager*.

5.3.4 Halaman transaksi

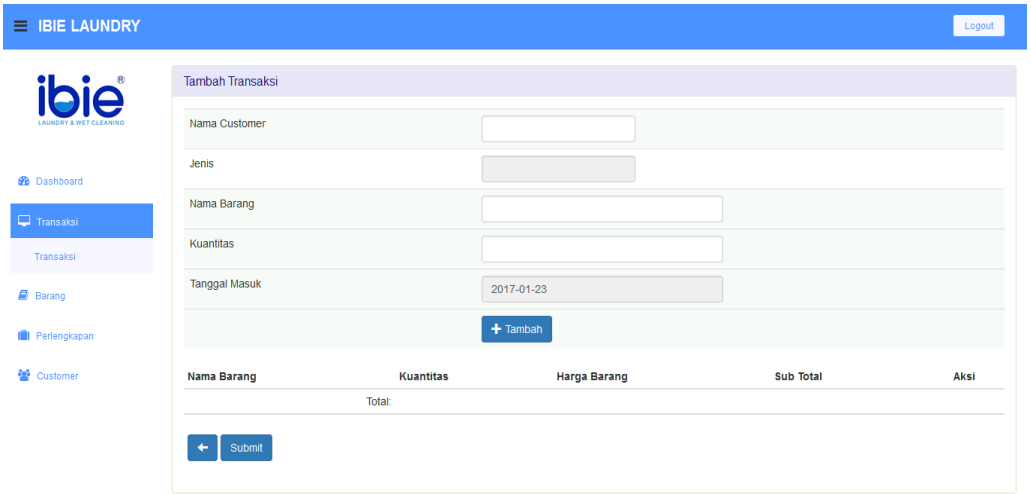
Halaman transaksi adalah halaman yang digunakan untuk mengelola data transaksi, yang berhak mengelola sistem ini adalah *admin*. Halaman ini meliputi tambah data transaksi, hapus data transaksi, *detail* transaksi, cetak transaksi dan kirim transaksi. Implementasi halaman transaksi *admin* dan *manager* dapat dilihat pada Gambar 5.17 dan 5.18, implementasi halaman tambah transaksi dapat dilihat pada Gambar 5.19, implementasi halaman *detail* transaksi *admin* dan *manager* dapat dilihat pada Gambar 5.20 dan 5.21, implementasi halaman cetak transaksi dapat dilihat pada Gambar 5.22 dan implementasi halaman kirim transaksi dapat dilihat pada Gambar 5.23.



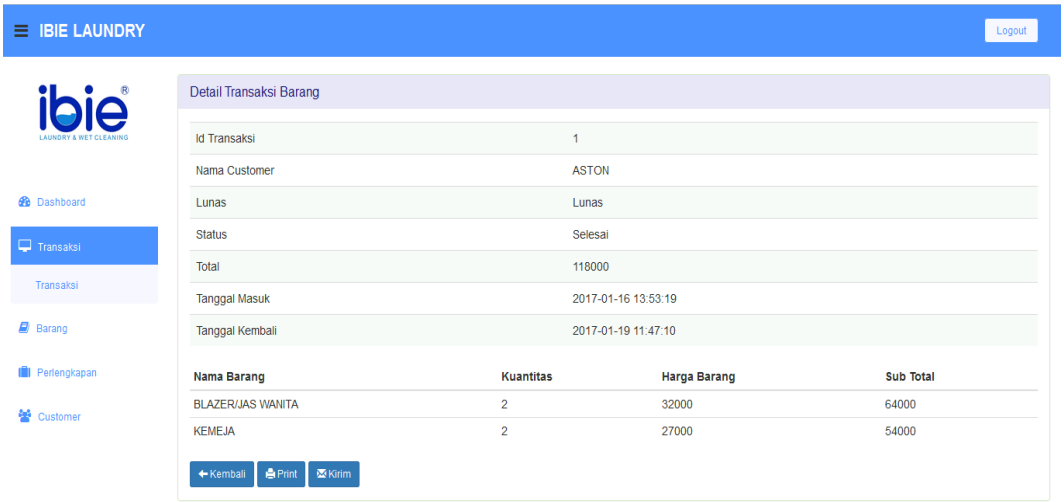
Gambar 5. 17 Implementasi halaman transaksi *admin*



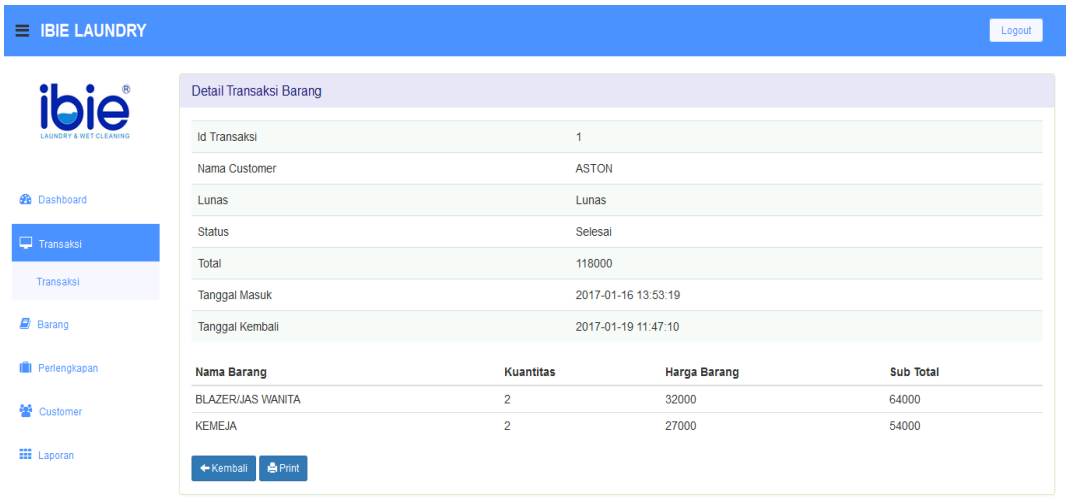
Gambar 5. 18 Implementasi halaman transaksi *manager*



Gambar 5. 19 Implementasi halaman tambah transaksi

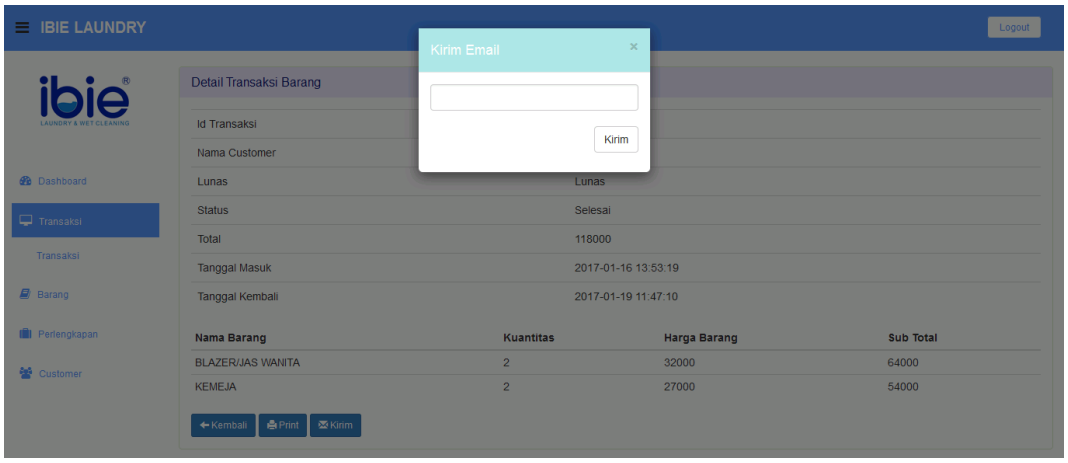


Gambar 5. 20 Implementasi halaman *detail* transaksi *admin*



Gambar 5. 21 Implementasi halaman *detail* transaksi *manager*

Gambar 5. 22 Implementasi halaman cetak transaksi



Gambar 5. 23 Implementasi halaman kirim transaksi

pada Gambar 5.24 dan 5.25 merupakan potongan kode yang melakukan aksi menampilkan halaman transaksi *admin* dan *manager*.

```
1  public function transaksi() {
2          $data['active'] = 'transaksi';
3          if ($this->session->userdata['logged_in']['level'] != "admin") {
4              redirect('admin/c_transaksi');
5          }
6          $data['title'] = 'Tabel Transaksi';
7          $data['quser'] = $this->m_transaksi->get_alltransaksi();
8          $this->template->load('sys_template','transaksi/v_user', $data);
9      }
```

Gambar 5. 24 Potongan kode halaman transaksi *admin*

Untuk keterangan gambar diatas, baris ke 7 merupakan kode untuk memanggil halaman transaksi menggunakan mode m_transaksi. Sedangkan kode pada baris ke 8 merupakan kode untuk mengarahkan ke halaman tabel data transaksi.

```
1  public function transaksi() {
2          $data['active'] = 'transaksi';
3          if ($this->session->userdata['logged_in']['level'] != "manager") {
4              redirect('manager/c_transaksi');
5          }
6          $data['title'] = 'Tabel Transaksi';
7          $data['quser'] = $this->m_transaksi->get_alltransaksi();
8          $this->template->load('sys_template_manager','transaksi/v_user_manager', $data);
9      }
```

Gambar 5. 25 Potongan kode halaman transaksi *manager*

Sama seperti potongan kode halaman sebelumnya, yaitu halaman transaksi *admin*, yang membedakan pada potongan kode ini adalah pada baris ke 8 menuju ke tampilan/view khusus *manager*.

Pada Gambar 5.26 merupakan potongan kode untuk halaman tambah transaksi. Halaman ini hanya dapat diakses oleh *admin* saja.

```
1   public function form() {
2       $data['active'] = 'transaksi';
3       $mau_ke = $this->uri->segment(4);
4       $id = $this->uri->segment(5);
5       if ($mau_ke == "add") {
6           $data['title'] = 'Tambah Transaksi';
7           $data['aksi'] = 'aksi_add';
8           $data['id_transaksi'] = $this->m_transaksi->getIdtransaksi();
9           $this->template->load('sys_template','transaksi/v_form', $data);
10      }
11      elseif ($mau_ke == "aksi_add") {
12          $data = array(
13              'nama_customer'  => ($this->input->post('nama_customer')),
14              'jenis'  => ($this->input->post('jenis')),
15              'total'  => ($this->input->post('total')),
16              'tanggal_masuk'  => (date('Y-m-d H:i:s'))
17          );
18          $this->m_transaksi->get_insert($data, "transaksi");
19      }
20      elseif ($mau_ke == "aksi_add2"){
21      $this->db->select_MAX('id_transaksi');
22      $data = $this->db->get('transaksi');
23      foreach($data->result() as $row){
24      $hasil=$row->id_transaksi;
25      }
26          $detail = array(
27              'id_transaksi' => ($hasil),
28              'nama_barang'  => ($this->input->post('nama_barang')),
29              'kuantitas'  => ($this->input->post('kuantitas')),
30              'harga_barang'  => ($this->input->post('harga_barang')),
31              'sub_total'  => ($this->input->post('sub_total'))
32          );
33          $this->m_transaksi->get_insert($detail, 'detail_transaksi');
34      }
```

Gambar 5. 26 Potongan kode halaman tambah transaksi

Untuk keterangan gambar diatas, pada baris ke 3 dan 4 digunakan untuk mengambil variabel url. kemudian, dimulai dari baris ke 5 digunakan untuk mengarahkan fungsi form sesuai dengan uri segmentnya. Untuk kode yang dimulai dari baris ke 11 digunakan untuk menginputkan data transaksi, dan untuk kode yang dimulai dari baris ke 20 digunakan untuk menginputkan *detail* transaksi.

Kemudian untuk potongan kode halaman *detail* transaksi *admin* dan *manager* dapat dilihat pada Gambar 5.27 dan 5.28

```
1   public function detail($id) {
2           $data['active'] = 'transaksi';
3           $data['title'] = 'Detail transaksi';
4           $data['quser'] = $this->m_transaksi->get_transaksi_byid($id);
5           $this->template->load('sys_template','transaksi/v_detail', $data);
6       }
```

Gambar 5. 27 Potongan kode halaman *detail* transaksi *admin*

Untuk keterangan gambar diatas, baris ke 4 merupakan kode untuk memanggil halaman transaksi menggunakan kode m_transaksi. Sedangkan kode pada baris ke 5 merupakan kode untuk mengarahkan ke halaman *detail* transaksi.

```
1  public function detail($id) {
2          $data['active'] = 'transaksi';
3          $data['title'] = 'Detail transaksi';
4          $data['quser'] = $this->m_transaksi->get_transaksi_byid($id);
5          $this->template->load('sys_template_manager','transaksi/v_detail_manager', $data);
6      }
```

Gambar 5. 28 Potongan kode halaman *detail* transaksi *manager*

Sama seperti potongan kode halaman sebelumnya, yaitu halaman *detail* transaksi *admin*, yang membedakan pada potongan kode ini adalah pada baris ke 5 menuju ke tampilan/view khusus *manager*.

Pada Gambar 5.29 dan 5.30 merupakan potongan kode untuk halaman cetak transaksi *admin* dan halaman cetak transaksi *manager*.

```
1  public function detail_print($id) {
2          $data['active'] = 'transaksi';
3          $data['title'] = 'Detail transaksi';
4          $data['quser'] = $this->m_transaksi->get_transaksi_byid($id);
5          $this->template->load('pure_template','transaksi/v_detail_print', $data);
6      }
```

Gambar 5. 29 Potongan kode halaman cetak transaksi *admin*

Untuk keterangan gambar diatas, baris ke 4 merupakan kode untuk memanggil halaman transaksi menggunakan kode m_transaksi. Sedangkan kode pada baris ke 5 merupakan kode untuk mengarahkan ke halaman *detail* transaksi, berbeda dengan *detail* transaksi, halaman ini menuju ke tampilan pure template yang siap untuk dicetak.

```
1  public function detail_print($id) {
2          $data['active'] = 'transaksi';
3          $data['title'] = 'Detail transaksi';
4          $data['quser'] = $this->m_transaksi->get_transaksi_byid($id);
5          $this->template->load('pure_template','transaksi/v_detail_manager_print', $data);
6      }
```

Gambar 5. 30 Potongan kode halaman cetak transaksi *manager*

Sama seperti potongan kode halaman sebelumnya, yaitu halaman cetak transaksi *admin*, yang membedakan pada potongan kode ini adalah pada baris ke 5

menuju ke tampilan/view khusus *manager* dengan tampilan pure template yang siap untuk dicetak.

Pada Gambar 5.31, 5.32 dan 5.33 merupakan potongan kode untuk kirim transaksi. Fitur kirim transaksi menggunakan *email* ini menggunakan plugin phpmailer.

```
1   public function sendmail($id){
2           require 'vendor/phpmailer/phpmailer/PHPMailerAutoload.php';
3           $penerima = $this->input->post('email');
4           $a = $this->m_transaksi->get_transaksi_byid($id);
5           foreach ($a as $rowdata) {
6               $id_transaksi = $rowdata->id_transaksi;
7               $nama_customer = $rowdata->nama_customer;
8               $lunas = $rowdata->lunas;
9               $status = $rowdata->status;
10              $total = $rowdata->total;
11              $tanggal_masuk = $rowdata->tanggal_masuk;
12              $tanggal_kembali = $rowdata->tanggal_kembali;
13          }
14          $msg = '<html><body><table width=100%><tr>
15                                      <td>Id Transaksi</td>
16                                      <td>'.$id_transaksi.'</td>
17                                  </tr><tr>
18                                      <td>Nama Customer</td>
19                                      <td>'.$nama_customer.'</td>
20                                  </tr><tr>
21                                      <td>Lunas</td>
22                                      <td>'.$lunas.'</td>
23                                  </tr><tr>
24                                      <td>Status</td>
25                                      <td>'.$status.'</td>
26                                  </tr><tr>
27                                      <td>Total</td>
28                                      <td>'.$total.'</td>
29                                  </tr><tr>
30                                      <td>Tanggal Masuk</td>
31                                      <td>'.$tanggal_masuk.'</td>
32                                  </tr><tr>
33                                      <td>Tanggal Kembali</td>
34                                      <td>'.$tanggal_kembali.'</td>
35                                  </tr></table>
```

Gambar 5. 31 Potongan kode halaman kirim transaksi

Untuk keterangan gambar diatas, pada baris ke 3 digunakan untuk menampilkan list *email customer* yang sudah terdaftar. Kemudian pada baris ke 4 digunakan untuk memanggil halaman transaksi menggunakan mode m_transaksi. Untuk potongan kode yang dimulai dari baris ke 14 digunakan untuk menampilkan data transaksi yang akan di tampilkan pada *email customer*.

```
37          <table width=100%> <thead>
38                  <tr>
39                      <th> Nama Barang </th>
40                      <th> Kuantitas </th>
41                      <th> Harga Barang </th>
42                      <th> Sub Total </th>
43                  </tr>
44              </thead>
45              <tbody>';
46              foreach ($a as $rowdata) {
47              $msg .= '<tr align="center">
48                          <td>'.$rowdata->nama_barang.'</td>
49                          <td>'.$rowdata->kuantitas.'</td>
50                          <td>'.$rowdata->harga_barang.'</td>
51                          <td>'.$rowdata->sub_total.'</td>
52                      </tr>';
53              }
54          $msg .= ' </tbody></table></body></html>';
55          $mail = new PHPMailer;
56          $mail->isSMTP();
57          $mail->Host = 'smtp.gmail.com';
58          $mail->SMTPAuth = true;
59          $mail->Username =
60          $mail->Password =
61          $mail->SMTPSecure = 'ssl';
62          $mail->Port = 465;
63          $mail->setFrom(                    , 'Ibie Laundry');
64          $mail->addAddress($penerima);
65          $mail->isHTML(true);
66          $mail->MsgHTML($msg);
67          $mail->Subject = 'Data Transaksi Ibie Laundry';
68          if(!$mail->send()) {
69              echo 'Message could not be sent.';
70              echo 'Mailer Error: ' . $mail->ErrorInfo;
71          } else {
```

Gambar 5. 32 Potongan kode halaman kirim transaksi (lanjutan)

Kemudian dimulai dari baris ke 55 merupakan konfigurasi untuk mengaktifkan koneksi *email* menggunakan gmail.

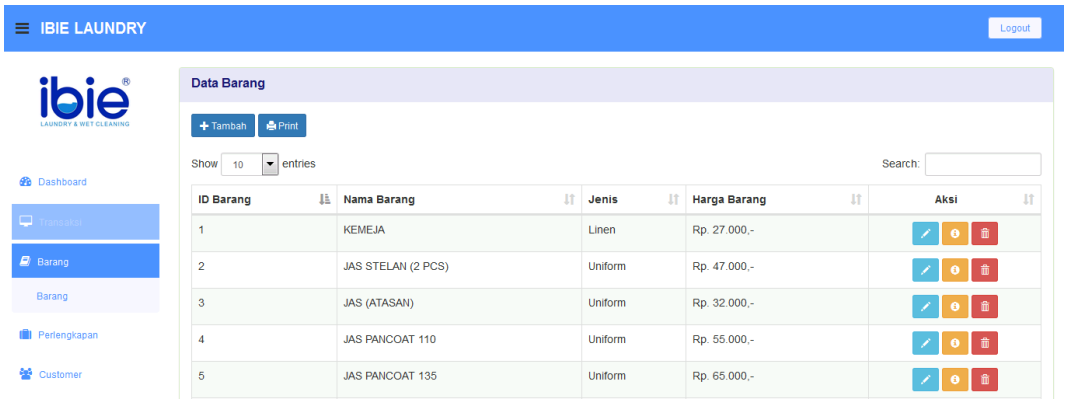
```
72              echo "<script>alert('Email berhasil terkirim');history.go(-1);</script>";
73              redirect('admin/c_transaksi/transaksi');
74          }
75      }
```

Gambar 5. 33 Potongan kode halaman kirim transaksi (lanjutan)

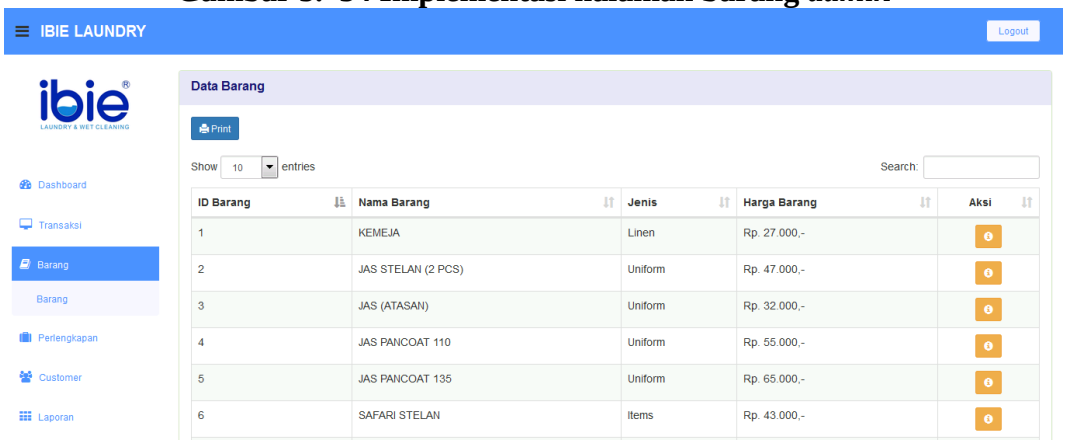
5.3.5 Halaman barang

Halaman barang adalah halaman yang digunakan untuk mengelola data barang, yang berhak mengelola sistem ini adalah *admin*. Halaman ini meliputi tambah data barang, ubah data barang, hapus data barang, *detail* barang, dan cetak barang. Implementasi halaman baramg *admin* dan *manager* dapat dilihat pada Gambar 5.34 dan 5.35, implementasi halaman tambah data barang dapat dilihat

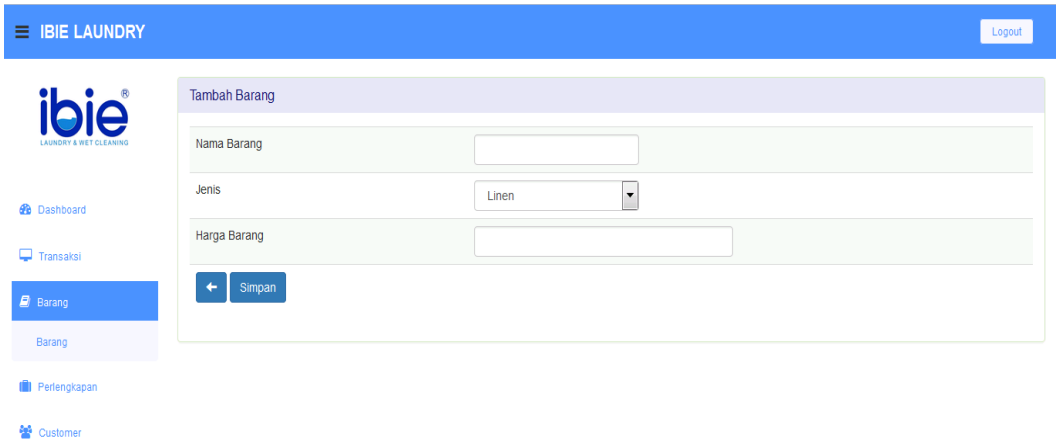
pada Gambar 5.36, implementasi halaman ubah data barang dapat dilihat pada Gambar 5.37, implementasi halaman *detail* data barang *admin* dan *manager* dapat dilihat pada Gambar 5.38 dan 5.39, dan implementasi halaman cetak data barang dapat dilihat pada Gambar 5.40.



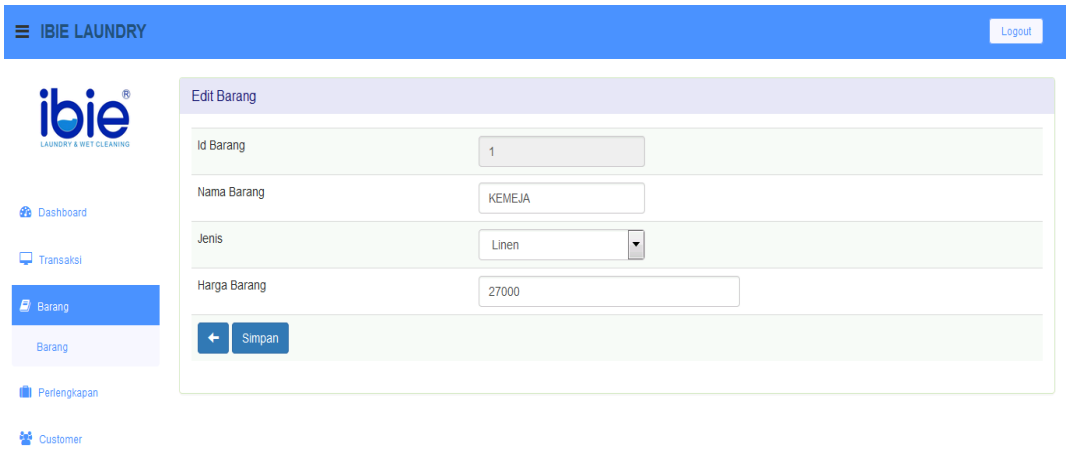
Gambar 5. 34 Implementasi halaman barang *admin*



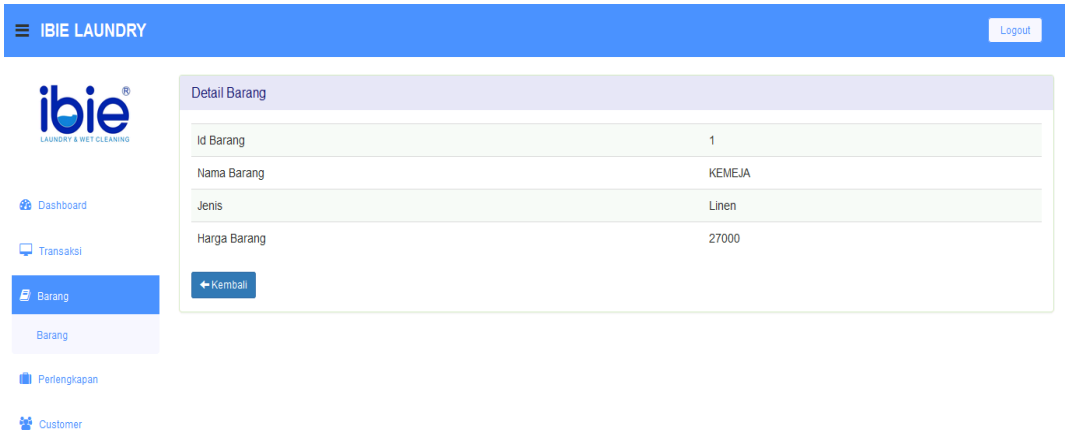
Gambar 5. 35 Implementasi halaman barang *manager*



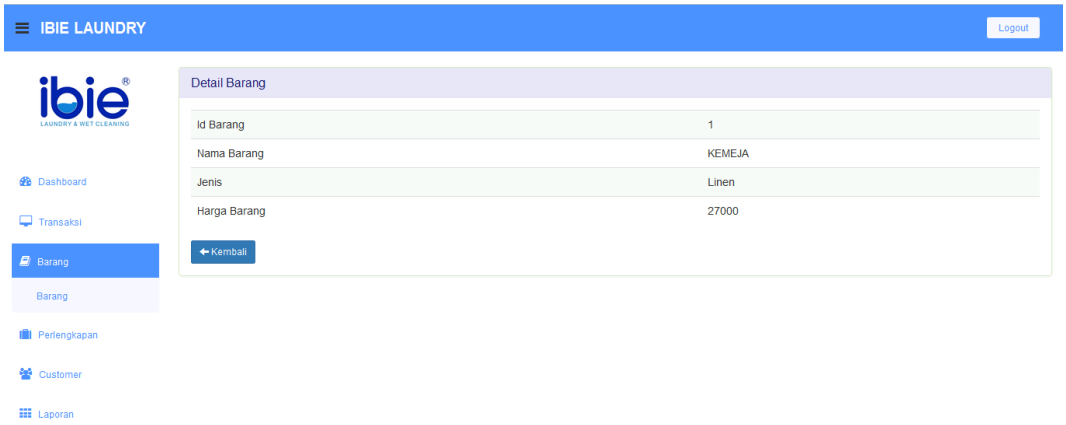
Gambar 5. 36 Implementasi halaman tambah barang



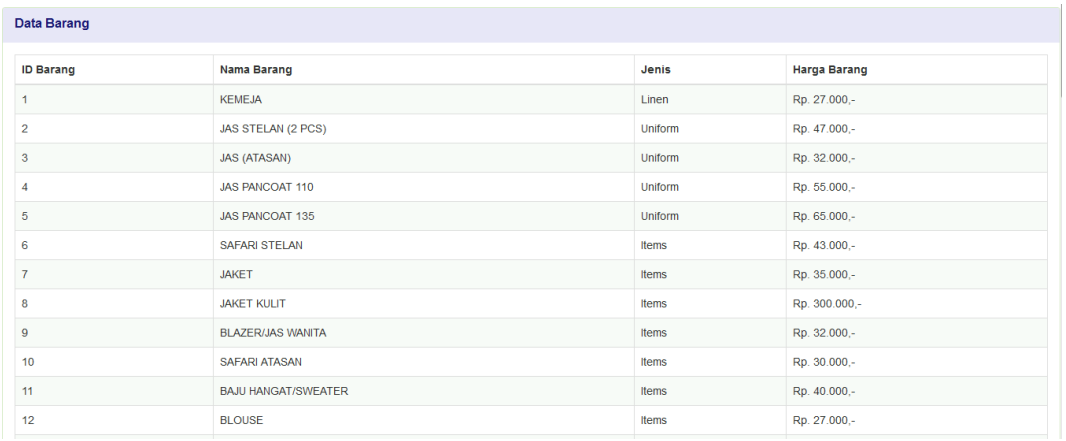
Gambar 5. 37 Implementasi halaman ubah barang



Gambar 5. 38 Implementasi halaman *detail* barang *admin*



Gambar 5. 39 Implementasi halaman *detail* barang *manager*



Data Barang

ID Barang	Nama Barang	Jenis	Harga Barang
1	KEMEJA	Linen	Rp. 27.000,-
2	JAS STELAN (2 PCS)	Uniform	Rp. 47.000,-
3	JAS (ATASAN)	Uniform	Rp. 32.000,-
4	JAS PANCOAT 110	Uniform	Rp. 55.000,-
5	JAS PANCOAT 135	Uniform	Rp. 65.000,-
6	SAFARI STELAN	Items	Rp. 43.000,-
7	JAKET	Items	Rp. 35.000,-
8	JAKET KULIT	Items	Rp. 300.000,-
9	BLAZER/JAS WANITA	Items	Rp. 32.000,-
10	SAFARI ATASAN	Items	Rp. 30.000,-
11	BAJU HANGAT/SWEATER	Items	Rp. 40.000,-
12	BLOUSE	Items	Rp. 27.000,-

Gambar 5. 40 Implementasi halaman cetak barang

pada Gambar 5.41 dan 5.42 merupakan potongan kode yang melakukan aksi menampilkan halaman barang *admin* dan *manager*.

```
1  public function barang() {
2          $data['active'] = 'barang';
3          if ($this->session->userdata['logged_in']['level'] != "admin") {
4              redirect('admin/c_barang');
5          }
6          $data['title'] = 'Tabel Barang';
7          $data['quser'] = $this->m_barang->get_allBarang();
8          $this->template->load('sys_template','barang/v_user', $data);
9      }
```

Gambar 5. 41 Potongan kode halaman barang *admin*

Untuk keterangan gambar diatas, baris ke 7 merupakan kode untuk memanggil halaman barang menggunakan mode m_barang. Sedangkan kode pada baris ke 8 merupakan kode untuk mengarahkan ke halaman tabel data barang.

```
1  public function barang() {
2          $data['active'] = 'barang';
3          if ($this->session->userdata['logged_in']['level'] != "manager") {
4              redirect('manager/c_barang');
5          }
6          $data['title'] = 'Tabel Barang';
7          $data['quser'] = $this->m_barang->get_allBarang();
8          $this->template->load('sys_template_manager','barang/v_user_manager', $data);
9      }
```

Gambar 5. 42 Potongan kode halaman barang *manager*

Sama seperti potongan kode halaman sebelumnya, yaitu halaman transaksi *admin*, yang membedakan pada potongan kode ini adalah pada baris ke 8 menuju ke tampilan/view khusus *manager*.

Pada Gambar 5.43 merupakan potongan kode untuk halaman tambah barang. Halaman ini hanya dapat diakses oleh *admin* saja.

```
1  public function form() {
2          $data['active'] = 'barang';
3          $mau_ke = $this->uri->segment(4);
4          $id = $this->uri->segment(5);
5          if ($mau_ke == "add") {
6              $data['title'] = 'Tambah Barang';
7              $data['aksi'] = 'aksi_add';
8              $data['id_barang'] = $this->m_barang->getIdBarang();
9              $this->template->load('sys_template','barang/v_form', $data);
10         }
11         elseif ($mau_ke == "aksi_add") {
12             $data = array(
13                 'id_barang'  => ($this->input->post('id_barang')),
14                 'nama_barang' => ($this->input->post('nama_barang')),
15                 'jenis' => ($this->input->post('jenis')),
16                 'harga_barang'  => ($this->input->post('harga_barang'))
17             );
18             $this->m_barang->get_insert($data);
19             $this->session->set_flashdata("Pesan", "<div class=\"alert alert-success\" id=\"alert\">
20                         <i class=\"glyphicon glyphicon-ok\"></i> Data berhasil di Simpan</div>");
21             redirect('admin/c_barang/barang');
22         }
23 }
```

Gambar 5. 43 Potongan kode halaman tambah barang

Untuk keterangan gambar diatas, pada baris ke 3 dan 4 digunakan untuk mengambil variabel url. kemudian, dimulai dari baris ke 5 digunakan untuk mengarahkan fungsi form sesuai dengan uri segmentnya. Untuk kode yang dimulai dari baris ke 11 digunakan untuk menginputkan data barang.

Kemudian untuk potongan kode halaman ubah data barang dapat dilihat pada Gambar 5.44

```
1  public function form() {
2          $data['active'] = 'barang';
3          $mau_ke = $this->uri->segment(4);
4          $id = $this->uri->segment(5);
5          if ($mau_ke == "edit") {
6              $data['qdata'] = $this->m_barang->get_barang_byid($id);
7              $data['title'] = 'Edit Barang';
8              $data['aksi']  = 'aksi_edit';
9              $data['id_barang'] = $this->m_barang->getIdBarang();
10             $this->template->load('sys_template','barang/v_form', $data);
11         } elseif ($mau_ke == "aksi_edit") {
12             $data = array(
13                 'id_barang'  => ($this->input->post('id_barang')),
14                 'nama_barang'  => ($this->input->post('nama_barang')),
15                 'jenis' => ($this->input->post('jenis')),
16                 'harga_barang'  => ($this->input->post('harga_barang'))
17             );
18             $hasil = $this->m_barang->get_update($id, $data);
19             $this->session->set_flashdata("Pesan", "<div class=\"alert alert-success\" id=\"alert\">
20                         <i class=\"glyphicon glyphicon-ok\"></i> Data berhasil di Update</div>");
21             redirect('admin/c_barang/barang');
22         }
23 }
```

Gambar 5. 44 Potongan kode halaman ubah barang

Untuk keterangan gambar diatas, data barang yang ingin diubah berdasarkan dengan id_barang yang dipilih sesuai pada kode baris ke 6. Setelah id_barang sudah dipilih maka sistem akan mengarahkan ke halaman update barang seperti pada baris ke 18.

Kemudian untuk potongan kode halaman *detail* barang *admin* dan *manager* dapat dilihat pada Gambar 5.45 dan 5.46

```
1  public function detail($id) {
2          $data['active'] = 'barang';
3          $data['title'] = 'Detail Barang';
4          $data['quser'] = $this->m_barang->get_barang_byid($id);
5          $this->template->load('sys_template','barang/v_detail', $data);
6      }
```

Gambar 5. 45 Potongan kode halaman *detail* barang *admin*

Untuk keterangan gambar diatas, baris ke 4 merupakan kode untuk memanggil halaman barang menggunakan kode m_barang. Sedangkan kode pada baris ke 5 merupakan kode untuk mengarahkan ke halaman *detail* barang.

```
1  public function detail($id) {
2          $data['active'] = 'barang';
3          $data['title'] = 'Detail Barang';
4          $data['quser'] = $this->m_barang->get_barang_byid($id);
5          $this->template->load('sys_template_manager','barang/v_detail_manager', $data);
6      }
```

Gambar 5. 46 Potongan kode halaman *detail* barang *manager*

Sama seperti potongan kode halaman sebelumnya, yaitu halaman *detail* barang *admin*, yang membedakan pada potongan kode ini adalah pada baris ke 5 menuju ke tampilan/view khusus *manager*.

Pada Gambar 5.47 dan 5.48 merupakan potongan kode untuk halaman cetak data barang *admin* dan halaman cetak data barang *manager*.

```
1  public function print_barang() {
2          $data['active'] = 'barang';
3          if ($this->session->userdata['logged_in']['level'] != "admin") {
4              redirect('admin/c_barang');
5          }
6          $data['title'] = 'Tabel Barang';
7          $data['quser'] = $this->m_barang->get_allBarang();
8          $this->template->load('pure_template','barang/v_user_print', $data);
9      }
```

Gambar 5. 47 Potongan kode halaman cetak barang *admin*

Untuk keterangan gambar diatas, baris ke 7 merupakan kode untuk memanggil halaman barang menggunakan kode m_barang. Sedangkan kode pada baris ke 8 merupakan kode untuk mengarahkan ke halaman *detail* barang, berbeda dengan *detail* barang, halaman ini menuju ke tampilan pure template yang siap untuk dicetak.

```
1  public function barang_print() {
2          $data['active'] = 'barang';
3          if ($this->session->userdata['logged_in']['level'] != "manager") {
4              redirect('manager/c_barang');
5          }
6          $data['title'] = 'Tabel Barang';
7          $data['quser'] = $this->m_barang->get_allBarang();
8          $this->template->load('pure_template','barang/v_user_manager_print', $data);
9      }
```

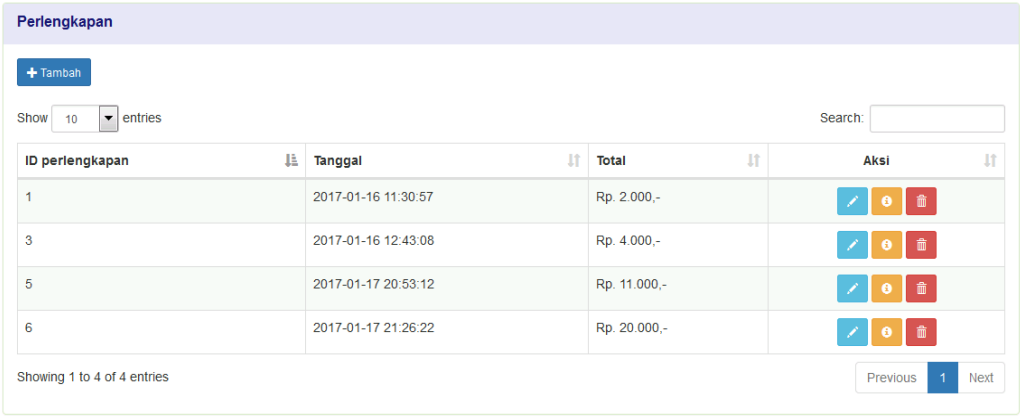
Gambar 5. 48 Potongan kode halaman cetak barang *manager*

Sama seperti potongan kode halaman sebelumnya, yaitu halaman cetak barang *admin*, yang membedakan pada potongan kode ini adalah pada baris ke 5 menuju ke tampilan/view khusus *manager* dengan tampilan pure template yang siap untuk dicetak.

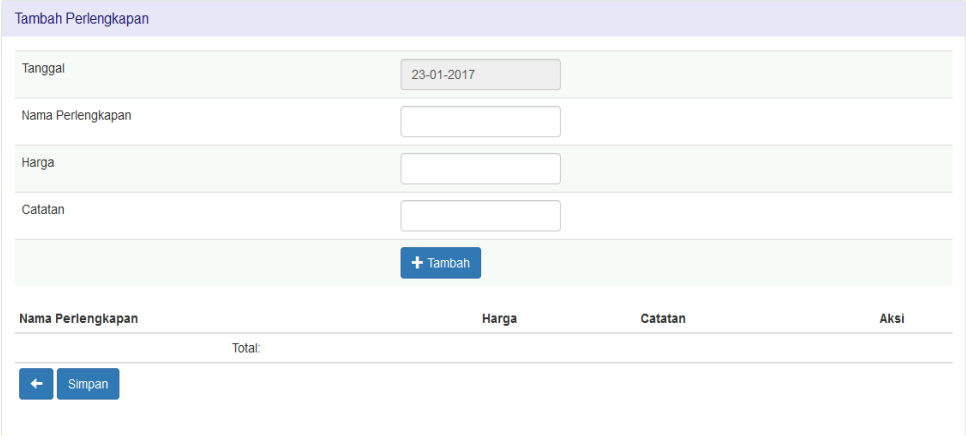
5.3.6 Halaman perlengkapan

Halaman perlengkapan adalah halaman yang digunakan untuk mengelola pembelian perlengkapan, halaman ini dapat diakses oleh *admin* dan *manager*.

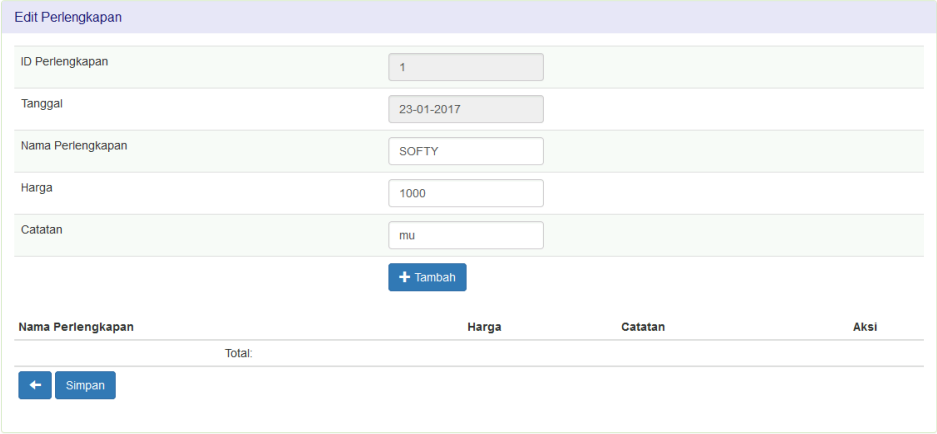
Halaman ini meliputi tambah perlengkapan, ubah perlengkapan, hapus perlengkapan, *detail* perlengkapan, dan cetak perlengkapan. Implementasi halaman perlengkapan dapat dilihat pada Gambar 5.49, implementasi halaman tambah perlengkapan dapat dilihat pada Gambar 5.50, implementasi halaman ubah perlengkapan dapat dilihat pada Gambar 5.51, implementasi halaman *detail* perlengkapan dapat dilihat pada Gambar 5.52 dan implementasi halaman cetak perlengkapan dapat dilihat pada Gambar 5.53.



Gambar 5. 49 Implementasi halaman perlengkapan



Gambar 5. 50 Implementasi halaman tambah perlengkapan



Gambar 5. 51 Implementasi halaman ubah perlengkapan

Gambar 5. 52 Implementasi halaman *detail* perlengkapan

Gambar 5. 53 Implementasi halaman cetak perlengkapan

pada Gambar 5.54 merupakan potongan kode yang melakukan aksi menampilkan halaman perlengkapan.

```
1   public function perlengkapan() {
2           $data['active'] = 'perlengkapan';
3           if ($this->session->userdata['logged_in']['level'] != "admin") {
4               redirect('admin/c_perlengkapan');
5           }
6           $data['title'] = 'Tabel Perlengkapan';
7           $data['quser'] = $this->m_perlengkapan->get_allPerlengkapan();
8           $this->template->load('sys_template','perlengkapan/v_user', $data);
9       }
```

Gambar 5. 54 Potongan kode halaman perlengkapan

Untuk keterangan gambar diatas, baris ke 7 merupakan kode untuk memanggil halaman perlengkapan menggunakan kode m_perlengkapan. Sedangkan kode pada baris ke 8 merupakan kode untuk mengarahkan ke halaman tabel perlengkapan. Untuk kode halaman perlengkapan *manager* tidak jauh beda dengan potongan kode sebelumnya, hanya perbedaannya pada baris ke 8 menuju ke tampilan/view khusus *manager*.

Pada Gambar 5.55 merupakan potongan kode untuk halaman tambah perlengkapan.

```
public function form() {
        $data['active'] = 'perlengkapan';
        $mau_ke = $this->uri->segment(4);
        $id = $this->uri->segment(5);
        if ($mau_ke == "add") {
            $data['title'] = 'Tambah Perlengkapan';
            $data['aksi'] = 'aksi_add';
            $data['id_perlengkapan'] = $this->m_perlengkapan->getIdPerlengkapan();
            $this->template->load('sys_template','perlengkapan/v_form', $data);
        } elseif ($mau_ke == "aksi_add") {
            $data = array(
                'tanggal' => (date('Y-m-d H:i:s')),
                'total' => ($this->input->post('total'))
            );
            $this->m_perlengkapan->get_insert($data, "perlengkapan");
        } elseif ($mau_ke == "aksi_add2") {
            $this->db->select_MAX('id_perlengkapan');
            $data = $this->db->get('perlengkapan');
            foreach($data->result() as $row){
                $hasil=$row->id_perlengkapan;
            }
            $detail = array(
                'id_perlengkapan'  => ($hasil),
                'tanggal' => (date('Y-m-d H:i:s')),
                'nama_perlengkapan' => ($this->input->post('nama_perlengkapan')),
                'harga' => ($this->input->post('harga')),
                'catatan' => ($this->input->post('catatan'))
            );
            $this->m_perlengkapan->get_insert($detail, 'detail_perlengkapan');
        }
```

Gambar 5. 55 Potongan kode halaman tambah perlengkapan

Untuk keterangan gambar diatas, pada baris ke 3 dan 4 digunakan untuk mengambil variabel url. kemudian, dimulai dari baris ke 5 digunakan untuk mengarahkan fungsi form sesuai dengan uri segmentnya. Untuk kode yang dimulai dari baris ke 11 digunakan untuk menginputkan data perlengkapan, dan untuk kode yang dimulai dari baris ke 16 digunakan untuk menginputkan *detail* perlengkapan. Untuk kode halaman tambah perlengkapan *manager* tidak jauh beda dengan potongan kode sebelumnya, hanya perbedaannya pada baris ke 9 menuju ke tampilan/view khusus *manager*.

Kemudian untuk potongan kode halaman ubah perlengkapan dapat dilihat pada Gambar 5.56

```
1   public function form() {
2           $data['active'] = 'perlengkapan';
3           $mau_ke = $this->uri->segment(4);
4           $id = $this->uri->segment(5);
5           if ($mau_ke == "edit") {
6               $data['qdata'] = $this->m_perlengkapan->get_perlengkapan_byid($id);
7               $data['title'] = 'Edit Perlengkapan';
8               $data['aksi']  = 'aksi_edit';
9               $data['id_perlengkapan'] = $this->m_perlengkapan->getIdPerlengkapan();
10              $this->template->load('sys_template','perlengkapan/v_form', $data);
11          } elseif ($mau_ke == "aksi_edit") {
12              $data = array(
13                  'id_perlengkapan'  => ($this->input->post('id_perlengkapan')),
14                  'tanggal' => ($this->input->post('tanggal')),
15                  'total' => ($this->input->post('total'))
16              );
17              $hasil = $this->m_perlengkapan->get_update($id, $data);
18              $this->session->set_flashdata("Pesan", "<div class=\"alert alert-success\" id=\"alert\">
19                          <i class=\"glyphicon glyphicon-ok\"></i> Data berhasil di Update</div>");
20              redirect('admin/c_perlengkapan/perlengkapan');
21          }
22      }
```

Gambar 5. 56 Potongan kode halaman ubah perlengkapan

Untuk keterangan gambar diatas, pembelian perlengkapan yang ingin diubah berdasarkan dengan id_perlengkapan yang dipilih sesuai pada kode baris ke 6. Setelah id_perlengkapan sudah dipilih maka sistem akan mengarahkan ke halaman update perlengkapan seperti pada baris ke 17. Untuk kode halaman ubah perlengkapan *manager* tidak jauh beda dengan potongan kode sebelumnya, hanya perbedaannya pada baris ke 10 dan 20 menuju ke tampilan/view khusus *manager*.

Kemudian untuk potongan kode halaman *detail* perlengkapan dapat dilihat pada Gambar 5.57.

```
1  public function detail($id) {
2          $data['active'] = 'perlengkapan';
3          $data['title'] = 'Detail Perlengkapan';
4          $data['quser'] = $this->m_perlengkapan->get_perlengkapan_byid($id);
5          $this->template->load('sys_template','perlengkapan/v_detail', $data);
6      }
```

Gambar 5. 57 Potongan kode halaman *detail* perlengkapan

Untuk keterangan gambar diatas, baris ke 4 merupakan kode untuk memanggil halaman perlengkapan menggunakan kode m_perlengkapan. Sedangkan kode pada baris ke 5 merupakan kode untuk mengarahkan ke halaman *detail* perlengkapan. Untuk kode halaman *detail* perlengkapan *manager* tidak jauh beda dengan potongan kode sebelumnya, hanya perbedaannya pada baris ke 5 menuju ke tampilan/view khusus *manager*.

Pada Gambar 5.58 merupakan potongan kode untuk halaman cetak perlengkapan.

```
1  public function detail_print($id) {
2          $data['active'] = 'perlengkapan';
3          $data['title'] = 'Detail Perlengkapan';
4          $data['quser'] = $this->m_perlengkapan->get_perlengkapan_byid($id);
5          $this->template->load('pure_template','perlengkapan/v_detail_print', $data);
6      }
```

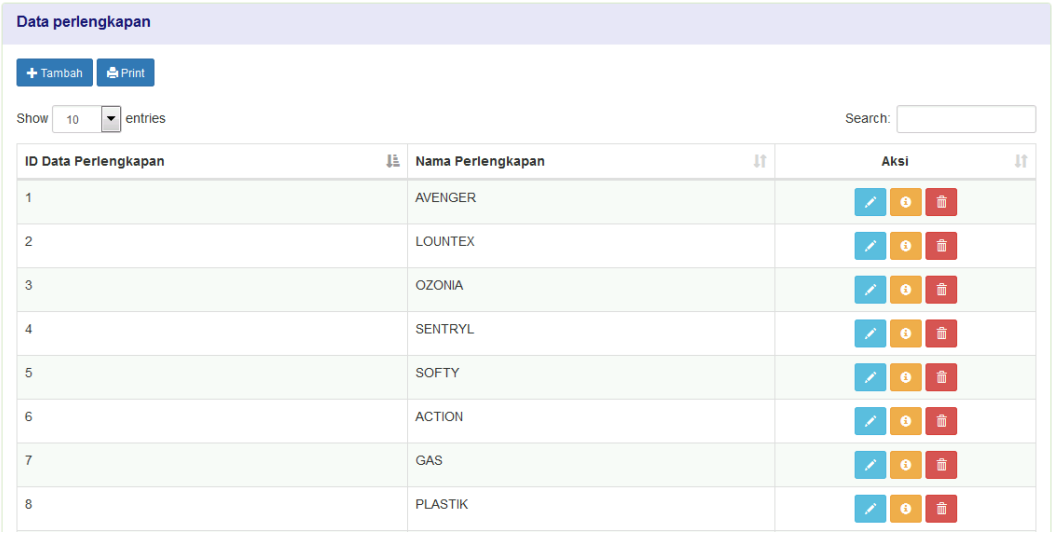
Gambar 5. 58 Potongan kode halaman cetak perlengkapan

Untuk keterangan gambar diatas, baris ke 4 merupakan kode untuk memanggil halaman perlengkapan menggunakan kode m_perlengkapan. Sedangkan kode pada baris ke 5 merupakan kode untuk mengarahkan ke halaman *detail* perlengkapan, berbeda dengan *detail* perlengkapan, halaman ini menuju ke tampilan pure template yang siap untuk dicetak. Untuk kode halaman cetak perlengkapan *manager* tidak jauh beda dengan potongan kode sebelumnya, hanya perbedaannya pada baris ke 5 menuju ke tampilan/view khusus *manager*.

5.3.7 Halaman data perlengkapan

Halaman data perlengkapan adalah halaman yang digunakan untuk mengelola data perlengkapan, halaman ini dapat diakses oleh *admin* dan *manager*. Halaman ini meliputi tambah data perlengkapan, ubah data perlengkapan, hapus data perlengkapan, *detail* data perlengkapan, dan cetak data perlengkapan. Implementasi halaman data perlengkapan dapat dilihat pada Gambar 5.59,

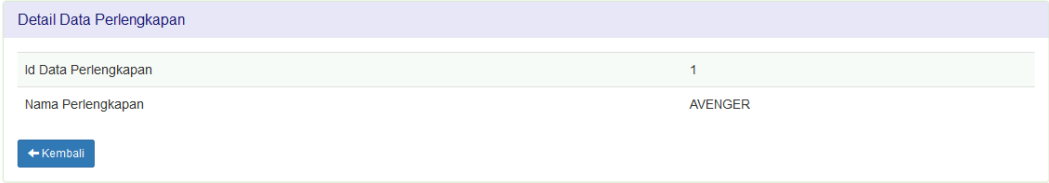
implementasi halaman tambah data perlengkapan dapat dilihat pada Gambar 5.60, implementasi halaman ubah data perlengkapan dapat dilihat pada Gambar 5.61, implementasi halaman *detail* data perlengkapan dapat dilihat pada Gambar 5.62 dan implementasi halaman cetak data perlengkapan dapat dilihat pada Gambar 5.63.



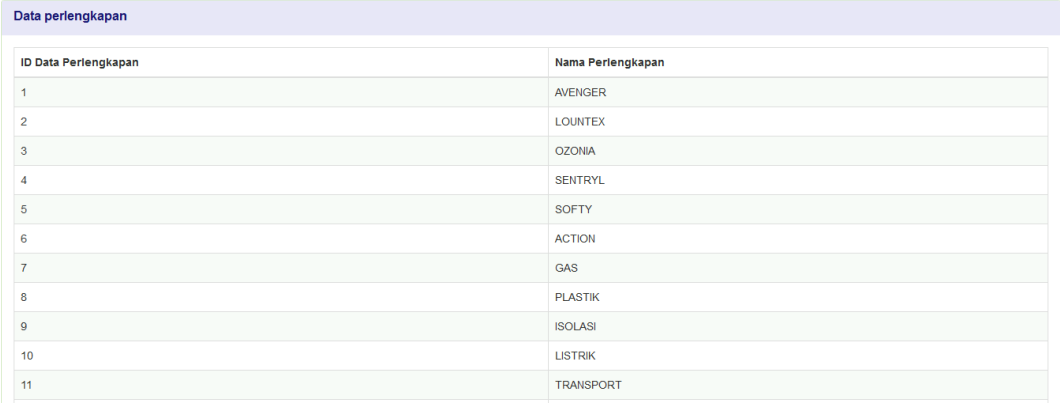
Gambar 5. 59 Implementasi halaman data perlengkapan

Gambar 5. 60 Implementasi halaman tambah data perlengkapan

Gambar 5. 61 Implementasi halaman ubah data perlengkapan



Gambar 5. 62 Implementasi halaman *detail* data perlengkapan



Data perlengkapan

ID Data Perlengkapan	Nama Perlengkapan
1	AVENGER
2	LOUNTEX
3	OZONIA
4	SENTRYL
5	SOFTY
6	ACTION
7	GAS
8	PLASTIK
9	ISOLASI
10	LISTRIK
11	TRANSPORT

Gambar 5. 63 Implementasi halaman cetak data perlengkapan

pada Gambar 5.64 merupakan potongan kode yang melakukan aksi menampilkan halaman data perlengkapan.

```
1  public function data_perlengkapan() {
2          $data['active'] = 'data_perlengkapan';
3          if ($this->session->userdata['logged_in']['level'] != "admin") {
4              redirect('admin/c_data_perlengkapan');
5          }
6          $data['title'] = 'Tabel Data Perlengkapan';
7          $data['quser'] = $this->m_data_perlengkapan->get_allDataPerlengkapan();
8          $this->template->load('sys_template','data_perlengkapan/v_user', $data);
9      }
```

Gambar 5. 64 Potongan kode halaman data perlengkapan

Untuk keterangan gambar diatas, baris ke 7 merupakan kode untuk memanggil halaman data perlengkapan menggunakan kode m_data_perlengkapan. Sedangkan kode pada baris ke 8 merupakan kode untuk mengarahkan ke halaman tabel data perlengkapan. Untuk kode halaman data perlengkapan *manager* tidak jauh beda dengan potongan kode sebelumnya, hanya perbedaannya pada baris ke 8 menuju ke tampilan/view khusus *manager*.

Pada Gambar 5.65 merupakan potongan kode untuk halaman tambah data perlengkapan.

```
1  public function form() {
2          $data['active'] = 'data_perlengkapan';
3          $mau_ke = $this->uri->segment(4);
4          $id = $this->uri->segment(5);
5          if ($mau_ke == "add") {
6              $data['title'] = 'Tambah Data Perlengkapan';
7              $data['aksi'] = 'aksi_add';
8              $data['id_data_perlengkapan'] = $this->m_data_perlengkapan->getIdDataPerlengkapan();
9              $this->template->load('sys_template','data_perlengkapan/v_form', $data);
10         }
11         elseif ($mau_ke == "aksi_add") {
12             $data = array(
13                 'id_data_perlengkapan'  => ($this->input->post('id_data_perlengkapan')),
14                 'nama_perlengkapan' => ($this->input->post('nama_perlengkapan'))
15             );
16             $this->m_data_perlengkapan->get_insert($data);
17             $this->session->set_flashdata("Pesan", "<div class=\"alert alert-success\" id=\"alert\">
18                         <i class=\"glyphicon glyphicon-ok\"></i> Data berhasil di Simpan</div>");
19             redirect('admin/c_data_perlengkapan/data_perlengkapan');
20         }
21 }
```

Gambar 5. 65 Potongan kode halaman tambah data perlengkapan

Untuk keterangan gambar diatas, pada baris ke 3 dan 4 digunakan untuk mengambil variabel url. kemudian, dimulai dari baris ke 5 digunakan untuk mengarahkan fungsi form sesuai dengan uri segmentnya. Untuk kode yang dimulai dari baris ke 11 digunakan untuk menginputkan data perlengkapan. Untuk kode halaman tambah data perlengkapan *manager* tidak jauh beda dengan potongan kode sebelumnya, hanya perbedaannya pada baris ke 9 menuju ke tampilan/view khusus *manager*.

Kemudian untuk potongan kode halaman ubah data perlengkapan dapat dilihat pada Gambar 5.66

```
public function form() {
        $data['active'] = 'data_perlengkapan';
        $mau_ke = $this->uri->segment(4);
        $id = $this->uri->segment(5);
        if ($mau_ke == "edit") {
            $data['qdata'] = $this->m_data_perlengkapan->get_data_perlengkapan_byid($id);
            $data['title'] = 'Edit Data Perlengkapan';
            $data['aksi']  = 'aksi_edit';
            $data['id_data_perlengkapan'] = $this->m_data_perlengkapan->getIdDataPerlengkapan();
            $this->template->load('sys_template','data_perlengkapan/v_form', $data);
        }
        elseif ($mau_ke == "aksi_edit") {
            $data = array(
                'id_data_perlengkapan'  => ($this->input->post('id_data_perlengkapan')),
                'nama_perlengkapan' => ($this->input->post('nama_perlengkapan'))
            );
            $hasil = $this->m_data_perlengkapan->get_update($id, $data);
            $this->session->set_flashdata("Pesan", "<div class=\"alert alert-success\" id=\"alert\">
                        <i class=\"glyphicon glyphicon-ok\"></i> Data berhasil di Update</div>");
            redirect('admin/c_data_perlengkapan/data_perlengkapan');
        }
}
```

Gambar 5. 66 Potongan kode halaman ubah data perlengkapan

Untuk keterangan gambar diatas, data perlengkapan yang ingin diubah berdasarkan dengan id_data_perlengkapan yang dipilih sesuai pada kode baris ke 6. Setelah id_data_perlengkapan sudah dipilih maka sistem akan mengarahkan ke halaman update data perlengkapan seperti pada baris ke 17. Untuk kode halaman ubah data perlengkapan *manager* tidak jauh beda dengan potongan kode sebelumnya, hanya perbedaannya pada baris ke 10 dan 20 menuju ke tampilan/view khusus *manager*.

Kemudian untuk potongan kode halaman *detail* data perlengkapan dapat dilihat pada Gambar 5.67.

```
1  public function detail($id) {
2          $data['active'] = 'data_perlengkapan';
3          $data['title'] = 'Detail Data Perlengkapan';
4          $data['quser'] = $this->m_data_perlengkapan->get_data_perlengkapan_byid($id);
5          $this->template->load('sys_template','data_perlengkapan/v_detail', $data);
6      }
```

Gambar 5. 67 Potongan kode halaman *detail* data perlengkapan

Untuk keterangan gambar diatas, baris ke 4 merupakan kode untuk memanggil halaman data perlengkapan menggunakan kode m_data_perlengkapan. Sedangkan kode pada baris ke 5 merupakan kode untuk mengarahkan ke halaman *detail* data perlengkapan. Untuk kode halaman *detail* data perlengkapan *manager* tidak jauh beda dengan potongan kode sebelumnya, hanya perbedaannya pada baris ke 5 menuju ke tampilan/view khusus *manager*.

Pada Gambar 5.68 merupakan potongan kode untuk halaman cetak data perlengkapan.

```
1  public function data_perlengkapan_print() {
2          $data['active'] = 'data_perlengkapan';
3          if ($this->session->userdata['logged_in']['level'] != "admin") {
4              redirect('admin/c_data_perlengkapan');
5          }
6          $data['title'] = 'Tabel Data Perlengkapan';
7          $data['quser'] = $this->m_data_perlengkapan->get_allDataPerlengkapan();
8          $this->template->load('pure_template','data_perlengkapan/v_user_print', $data);
9      }
```

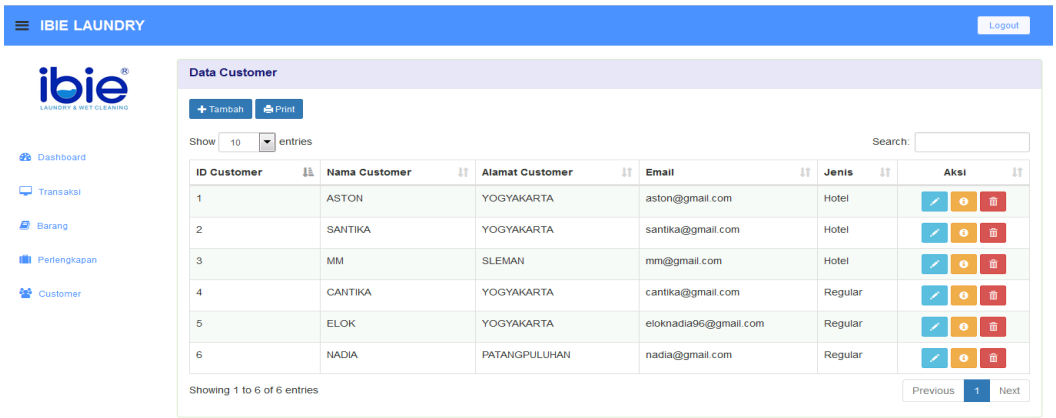
Gambar 5. 68 Potongan kode halaman cetak data perlengkapan

Untuk keterangan gambar diatas, baris ke 7 merupakan kode untuk memanggil halaman data perlengkapan menggunakan kode m_data_perlengkapan. Sedangkan kode pada baris ke 8 merupakan kode untuk mengarahkan ke halaman

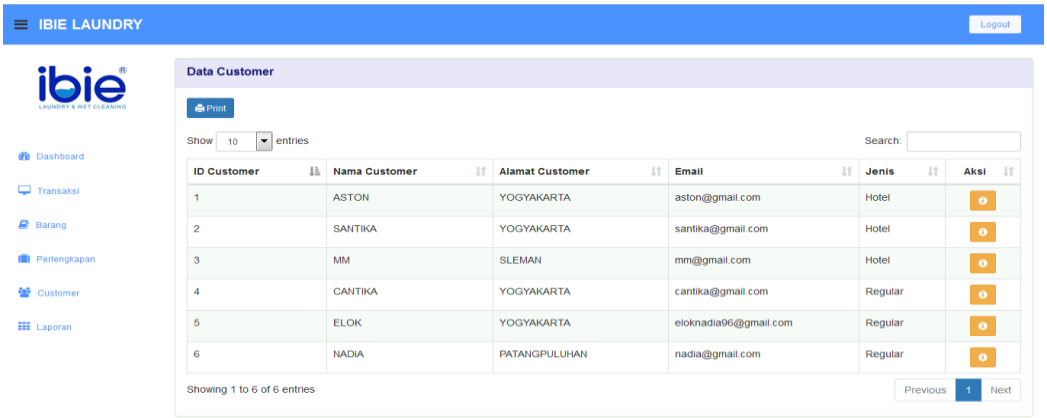
detail data perlengkapan, berbeda dengan *detail* data perlengkapan, halaman ini menuju ke tampilan pure template yang siap untuk dicetak. Untuk kode halaman cetak data perlengkapan *manager* tidak jauh beda dengan potongan kode sebelumnya, hanya perbedaannya pada baris ke 8 menuju ke tampilan/view khusus *manager*.

5.3.8 Halaman *customer*

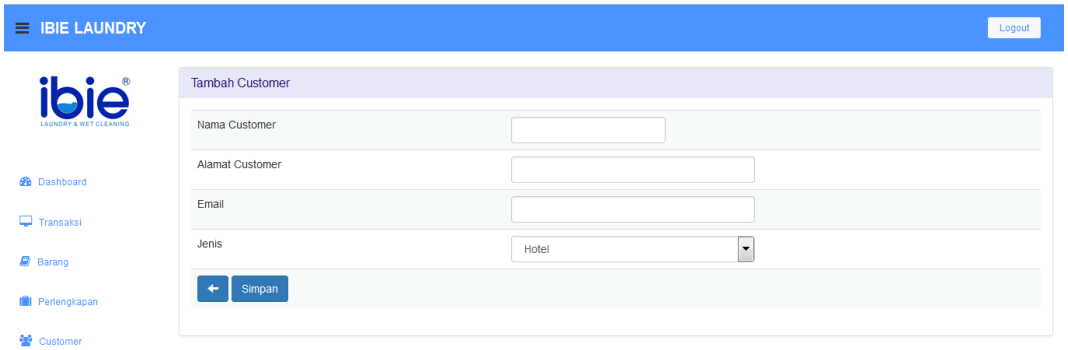
Halaman *customer* adalah halaman yang digunakan untuk mengelola data *customer*, yang berhak mengelola sistem ini adalah *admin*. Halaman ini meliputi tambah *customer*, ubah *customer*, hapus *customer*, *detail customer*, dan cetak *customer*. Implementasi halaman *customer admin* dan *manager* dapat dilihat pada Gambar 5.69 dan 5.70, implementasi halaman tambah *customer* dapat dilihat pada Gambar 5.71, implementasi halaman ubah *customer* dapat dilihat pada Gambar 5.72, implementasi halaman *detail customer admin* dan *manager* dapat dilihat pada Gambar 5.73 dan 5.74 dan implementasi halaman cetak *customer* dapat dilihat pada Gambar 5.75.



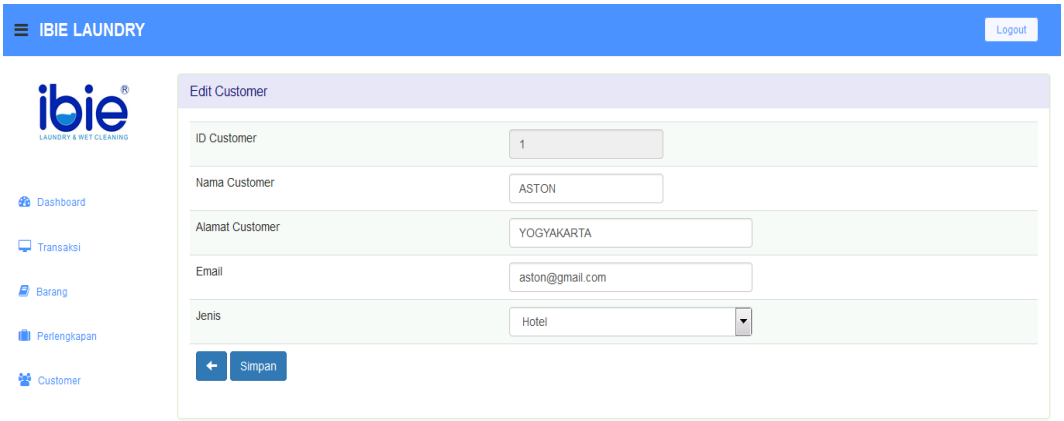
Gambar 5. 69 Implementasi halaman *customer admin*



Gambar 5. 70 Implementasi halaman *customer manager*

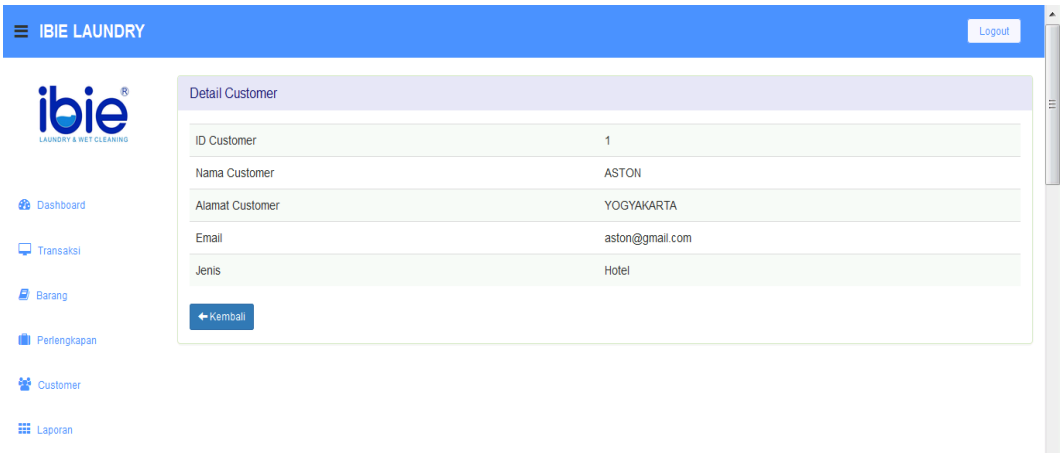


Gambar 5. 71 Implementasi halaman tambah *customer*

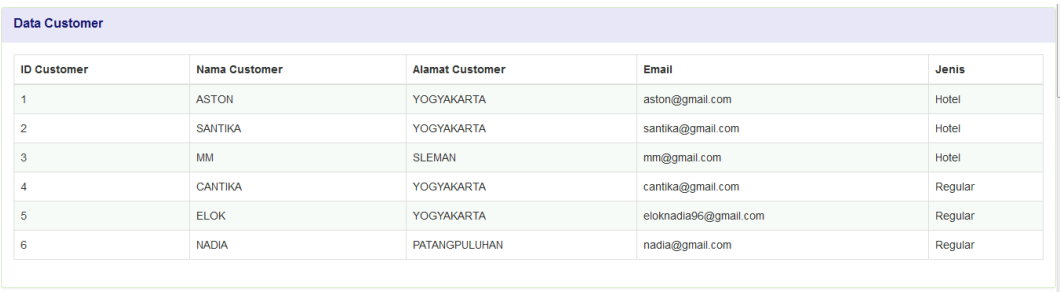


Gambar 5. 72 Implementasi halaman ubah *customer*

Gambar 5. 73 Implementasi halaman *detail customer admin*



Gambar 5. 74 Implementasi halaman *detail customer manager*



Data Customer

ID Customer	Nama Customer	Alamat Customer	Email	Jenis
1	ASTON	YOGYAKARTA	aston@gmail.com	Hotel
2	SANTIKA	YOGYAKARTA	santika@gmail.com	Hotel
3	MM	SLEMAN	mm@gmail.com	Hotel
4	CANTIKA	YOGYAKARTA	cantika@gmail.com	Regular
5	ELOK	YOGYAKARTA	eloknadia96@gmail.com	Regular
6	NADIA	PATANGPULUHAN	nadia@gmail.com	Regular

Gambar 5. 75 Implementasi halaman cetak *customer*

pada Gambar 5.76 dan 5.77 merupakan potongan kode yang melakukan aksi menampilkan halaman *customer admin* dan *manager*.

```
1  public function customer() {
2          $data['active'] = 'customer';
3          if ($this->session->userdata['logged_in']['level'] != "admin") {
4              redirect('admin/c_customer');
5          }
6          $data['title'] = 'Tabel Customer customer';
7          $data['cuser'] = $this->m_customer->get_allcustomer();
8          $this->template->load('sys_template','customer/v_user', $data);
9      }
```

Gambar 5. 76 Potongan kode halaman *customer admin*

Untuk keterangan gambar diatas, baris ke 7 merupakan kode untuk memanggil halaman *customer* menggunakan mode m_*customer*. Sedangkan kode pada baris ke 8 merupakan kode untuk mengarahkan ke halaman tabel *customer*.

```
1  public function customer() {
2          $data['active'] = 'customer';
3          if ($this->session->userdata['logged_in']['level'] != "manager") {
4              redirect('manager/c_customer');
5          }
6          $data['title'] = 'Tabel Customer customer';
7          $data['cuser'] = $this->m_customer->get_allcustomer();
8          $this->template->load('sys_template_manager','customer/v_user_manager', $data);
9      }
```

Gambar 5. 77 Potongan kode halaman *customer manager*

Sama seperti potongan kode halaman sebelumnya, yaitu halaman *customer admin*, yang membedakan pada potongan kode ini adalah pada baris ke 8 menuju ke tampilan/view khusus *manager*.

Pada Gambar 5.78 merupakan potongan kode untuk halaman tambah *customer*. Halaman ini hanya dapat diakses oleh *admin* saja.

```
1   public function form() {
2           $data['active'] = 'customer';
3           $mau_ke = $this->uri->segment(4);
4           $id = $this->uri->segment(5);
5           if ($mau_ke == "add") {
6               $data['title'] = 'Tambah Customer customer';
7               $data['aksi'] = 'aksi_add';
8               $data['id_customer'] = $this->m_customer->getIdcustomer();
9               $this->template->load('sys_template','customer/v_form', $data);
10          }
11          elseif ($mau_ke == "aksi_add") {
12              $data = array(
13                  'id_customer'  => ($this->input->post('id_customer')),
14                  'nama_customer' => ($this->input->post('nama_customer')),
15                  'alamat_customer'  => ($this->input->post('alamat_customer')),
16                  'email'  => ($this->input->post('email')),
17                  'jenis'  => ($this->input->post('jenis'))
18              );
19              $this->m_customer->get_insert($data);
20              $this->session->set_flashdata("Pesan", "<div class=\"alert alert-success\" id=\"alert\">
21                          <i class=\"glyphicon glyphicon-ok\"></i> Data berhasil di Simpan</div>");
22              redirect('admin/c_customer/customer');
23          }
```

Gambar 5. 78 Potongan kode halaman tambah *customer*

Untuk keterangan gambar diatas, pada baris ke 3 dan 4 digunakan untuk mengambil variabel url. kemudian, dimulai dari baris ke 5 digunakan untuk mengarahkan fungsi form sesuai dengan uri segmentnya. Untuk kode yang dimulai dari baris ke 11 digunakan untuk menginputkan data *customer*.

Kemudian untuk potongan kode halaman ubah data *customer* dapat dilihat pada Gambar 5.79

```
1   public function form() {
2           $data['active'] = 'customer';
3           $mau_ke = $this->uri->segment(4);
4           $id = $this->uri->segment(5);
5           if ($mau_ke == "edit") {
6               $data['qdata'] = $this->m_customer->get_customer_byid($id);
7               $data['title'] = 'Edit Customer customer';
8               $data['aksi']  = 'aksi_edit';
9               $data['id_customer'] = $this->m_customer->getIdcustomer();
10              $this->template->load('sys_template','customer/v_form', $data);
11          }
12          elseif ($mau_ke == "aksi_edit") {
13              $data = array(
14                  'id_customer'  => ($this->input->post('id_customer')),
15                  'nama_customer' => ($this->input->post('nama_customer')),
16                  'alamat_customer'  => ($this->input->post('alamat_customer')),
17                  'email'  => ($this->input->post('email')),
18                  'jenis'  => ($this->input->post('jenis'))
19              );
20              $hasil = $this->m_customer->get_update($id, $data);
21              $this->session->set_flashdata("Pesan", "<div class=\"alert alert-success\" id=\"alert\">
22                          <i class=\"glyphicon glyphicon-ok\"></i> Data berhasil di Update</div>");
23              redirect('admin/c_customer/customer');
24          }
25      }
```

Gambar 5. 79 Potongan kode halaman ubah *customer*

Untuk keterangan gambar diatas, data *customer* yang ingin diubah berdasarkan dengan id_*customer* yang dipilih sesuai pada kode baris ke 6. Setelah

id_*customer* sudah dipilih maka sistem akan mengarahkan ke halaman update barang seperti pada baris ke 20.

Kemudian untuk potongan kode halaman *detail* customer *admin* dan *manager* dapat dilihat pada Gambar 5.80 dan 5.81

```
1  public function detail($id) {
2          $data['active'] = 'customer';
3          $data['title'] = 'Detail Customer customer';
4          $data['quser'] = $this->m_customer->get_customer_byid($id);
5          $this->template->load('sys_template','customer/v_detail', $data);
6      }
```

Gambar 5. 80 Potongan kode halaman *detail customer admin*

Untuk keterangan gambar diatas, baris ke 4 merupakan kode untuk memanggil halaman *customer* menggunakan kode m_*customer*. Sedangkan kode pada baris ke 5 merupakan kode untuk mengarahkan ke halaman *detail customer*.

```
1  public function detail($id) {
2          $data['active'] = 'customer';
3          $data['title'] = 'Detail Customer customer';
4          $data['quser'] = $this->m_customer->get_customer_byid($id);
5          $this->template->load('sys_template_manager','customer/v_detail_manager', $data);
6      }
```

Gambar 5. 81 Potongan kode halaman *detail customer manager*

Sama seperti potongan kode halaman sebelumnya, yaitu halaman *detail customer admin*, yang membedakan pada potongan kode ini adalah pada baris ke 5 menuju ke tampilan/view khusus *manager*.

Pada Gambar 5.81 dan 5.82 merupakan potongan kode untuk halaman cetak *customer admin* dan halaman cetak *customer manager*.

```
1  public function customer_print() {
2          $data['active'] = 'customer';
3          if ($this->session->userdata['logged_in']['level'] != "admin") {
4              redirect('admin/c_customer');
5          }
6          $data['title'] = 'Tabel Customer customer';
7          $data['quser'] = $this->m_customer->get_allcustomer();
8          $this->template->load('pure_template','customer/v_user_print', $data);
9      }
```

Gambar 5. 82 Potongan kode halaman cetak *customer admin*

Untuk keterangan gambar diatas, baris ke 7 merupakan kode untuk memanggil halaman *customer* menggunakan kode m_*customer*. Sedangkan kode pada baris ke 8 merupakan kode untuk mengarahkan ke halaman *detail customer*,

berbeda dengan *detail customer*, halaman ini menuju ke tampilan pure template yang siap untuk dicetak.

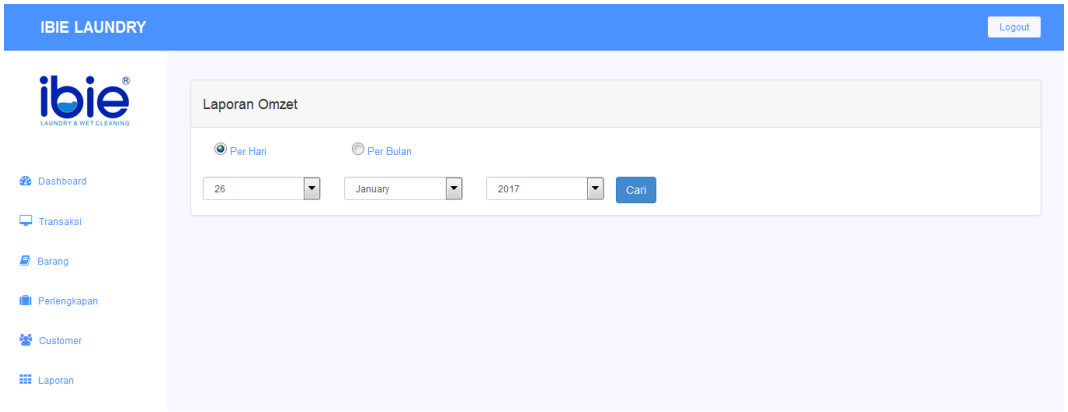
```
1  public function customer_print() {
2          $data['active'] = 'customer';
3          if ($this->session->userdata['logged_in']['level'] != "manager") {
4              redirect('manager/c_customer');
5          }
6          $data['title'] = 'Tabel Customer customer';
7          $data['cuser'] = $this->m_customer->get_allcustomer();
8          $this->template->load('pure_template','customer/v_user_manager_print', $data);
9      }
```

Gambar 5. 83 Potongan kode halaman cetak *customer manager*

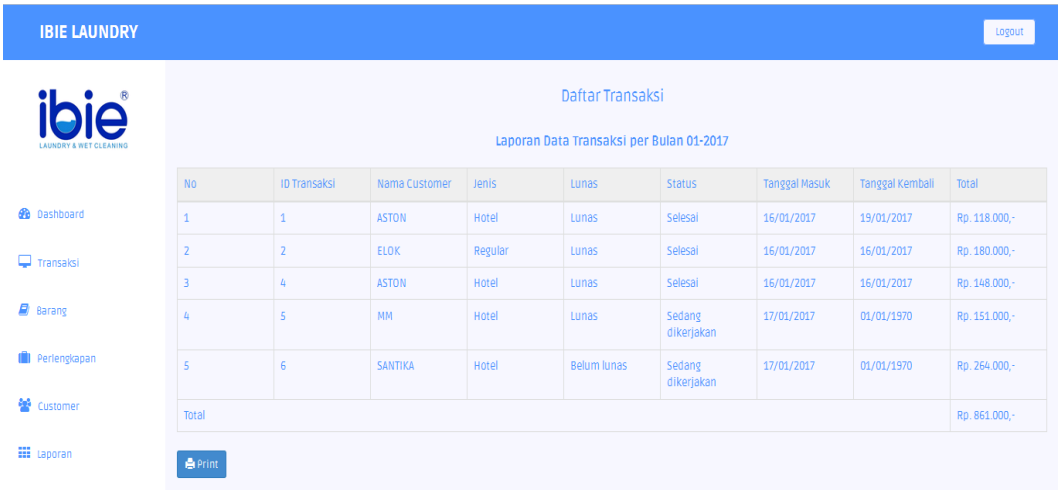
Sama seperti potongan kode halaman sebelumnya, yaitu halaman cetak *customer admin*, yang membedakan pada potongan kode ini adalah pada baris ke 8 menuju ke tampilan/view khusus *manager* dengan tampilan pure template yang siap untuk dicetak.

5.3.9 Halaman laporan

Halaman laporan adalah halaman yang digunakan untuk melihat laporan, yang berhak melihat halaman ini adalah *manager*. Halaman ini meliputi pilih jangka waktu laporan, laporan *omzet*, laporan pengeluaran dan laporan *profit*. Implementasi halaman jangka waktu laporan dapat dilihat pada Gambar 5.84, implementasi halaman laporan *omzet* dapat dilihat pada Gambar 5.85, implementasi halaman cetak laporan *omzet* dapat dilihat pada Gambar 5.86, implementasi halaman laporan pengeluaran dapat dilihat pada Gambar 5.87, implementasi halaman cetak laporan pengeluaran dapat dilihat pada Gambar 5.88, implementasi halaman laporan *profit* dapat dilihat pada Gambar 5.89, dan implementasi halaman cetak laporan *profit* dapat dilihat pada Gambar 5.90.



Gambar 5. 84 Implementasi halaman jangka waktu laporan



No	ID Transaksi	Nama Customer	Jenis	Lunas	Status	Tanggal Masuk	Tanggal Kembali	Total
1	1	ASTON	Hotel	Lunas	Selesai	16/01/2017	19/01/2017	Rp. 118.000,-
2	2	ELOK	Regular	Lunas	Selesai	16/01/2017	16/01/2017	Rp. 180.000,-
3	4	ASTON	Hotel	Lunas	Selesai	16/01/2017	16/01/2017	Rp. 148.000,-
4	5	MM	Hotel	Lunas	Sedang dikerjakan	17/01/2017	01/01/1970	Rp. 151.000,-
5	6	SANTIKA	Hotel	Belum lunas	Sedang dikerjakan	17/01/2017	01/01/1970	Rp. 264.000,-
Total								Rp. 861.000,-

Gambar 5. 85 Implementasi halaman laporan *omzet*

Daftar Transaksi

Laporan Data Transaksi per Bulan 01-2017

No	ID Transaksi	Nama Customer	Jenis	Lunas	Status	Tanggal Masuk	Tanggal Kembali	Total
1	1	ASTON	Hotel	Lunas	Selesai	16/01/2017	19/01/2017	Rp. 118.000,-
2	2	ELOK	Regular	Lunas	Selesai	16/01/2017	16/01/2017	Rp. 180.000,-
3	4	ASTON	Hotel	Lunas	Selesai	16/01/2017	16/01/2017	Rp. 148.000,-
4	5	MM	Hotel	Lunas	Sedang dikerjakan	17/01/2017	01/01/1970	Rp. 151.000,-
5	6	SANTIKA	Hotel	Belum lunas	Sedang dikerjakan	17/01/2017	01/01/1970	Rp. 264.000,-
Total								Rp. 861.000,-

Gambar 5. 86 Implementasi halaman cetak laporan *omzet*

Gambar 5. 87 Implementasi halaman laporan pengeluaran

Daftar Pembelian Perlengkapan

Laporan Pembelian Perlengkapan per Bulan 01-2017

No	ID Perlengkapan	Tanggal	Total
1	1	16/01/2017	Rp. 2.000,-
2	3	16/01/2017	Rp. 4.000,-
3	5	17/01/2017	Rp. 11.000,-
4	6	17/01/2017	Rp. 20.000,-
Total			Rp. 37.000,-

Gambar 5. 88 Implementasi halaman cetak laporan pengeluaran

Gambar 5. 89 Implementasi halaman laporan *profit*

Laporan Data Profit per Bulan 01-2017

Daftar Transaksi

No	ID Transaksi	Nama Customer	Jenis	Lunas	Status	Tanggal Masuk	Tanggal Kembali	Total
1	1	ASTON	Hotel	Lunas	Selesai	16/01/2017	19/01/2017	Rp. 118.000,-
2	2	ELOK	Regular	Lunas	Selesai	16/01/2017	16/01/2017	Rp. 180.000,-
3	4	ASTON	Hotel	Lunas	Selesai	16/01/2017	16/01/2017	Rp. 148.000,-
4	5	MM	Hotel	Lunas	Sedang dikerjakan	17/01/2017	01/01/1970	Rp. 151.000,-
5	6	SANTIKA	Hotel	Belum lunas	Sedang dikerjakan	17/01/2017	01/01/1970	Rp. 264.000,-
Total								Rp. 861.000,-

Daftar Perlengkapan

No	ID Perlengkapan	Tanggal	Total
1	1	16/01/2017	Rp. 2.000,-
2	3	16/01/2017	Rp. 4.000,-
3	5	17/01/2017	Rp. 11.000,-
4	6	17/01/2017	Rp. 20.000,-
Total			Rp. 37.000,-

Profit: Rp 861.000 - Rp 37.000 = Rp 824.000,-

Gambar 5. 90 Implementasi halaman cetak laporan *profit*

pada Gambar 5.91 merupakan potongan kode yang melakukan aksi menampilkan halaman jangka waktu laporan.

```
1  public function laporan_omzet() {
2          $data['active'] = 'laporan_omzet';
3          if ($this->session->userdata['logged_in']['level'] != "manager") {
4              redirect('manager/c_laporan');
5          }
6          $this->template->load('sys_template_manager','laporan/v_lap_omzet', $data);
7      }
```

Gambar 5. 91 Potongan kode halaman jangka waktu laporan

Untuk keterangan gambar diatas, kode pada baris ke 6 merupakan kode untuk mengarahkan ke halaman tabel laporan *omzet*. Sama seperti laporan *omzet*, laporan pengeluaran dan laporan *profit* juga pada baris ke 8 mengarah ke tampilan/view laporan pengeluaran dan laporan *profit*.

Pada Gambar 5.92 merupakan potongan kode untuk halaman laporan *omzet*.

```
1  public function proc_lap_omzet()
2      {
3          $data['active'] = 'laporan_omzet';
4          if ($this->session->userdata['logged_in']['level'] != "manager") {
5              redirect('manager/c_laporan');
6          }
7          $tanggal=$this->input->post('tanggal');
8          $bulan=$this->input->post('bulan');
9          $tahun=$this->input->post('tahun');
10         $pilihan=$this->input->post('pilihan');
11         $data['tanggal'] = $tanggal;
12         $data['bulan'] = $bulan;
13         $data['tahun'] = $tahun;
14         $data['pilihan'] = $pilihan;
15         $data['controller'] = $this->uri->segment(2);
16         $data['action'] = $this->uri->segment(3);
17         if($pilihan=="perhari"){
18             $data["transaksi"]=$this->m_laporan->daily_report($tanggal,$bulan,$tahun);
19             $data["judul"]="Laporan Data Transaksi per Tanggal " .$tanggal."-".$bulan."-".$tahun;
20             $data['pencarian']=$tanggal."-".$bulan."-".$tahun;
21         }
22         else {
23             $data["transaksi"]=$this->m_laporan->monthly_report($bulan,$tahun);
24             $data['judul']="Laporan Data Transaksi per Bulan " .$bulan."-".$tahun;
25             $data['pencarian']="x-".$bulan."-".$tahun;
26         }
27         $this->template->load('sys_template_manager','laporan/v_hasil_lap_omzet', $data);
28     }
```

Gambar 5. 92 Potongan kode halaman laporan *omzet*

Untuk keterangan gambar diatas, laporan *omzet* yang dipanggil berdasarkan pilihan jangka waktu seperti yang dimulai pada baris ke 17. Apabila sudah berhasil maka sistem akan membuka halaman *detail* laporan *omzet* sesuai dengan kode pada baris ke 27.

Pada Gambar 5.93 merupakan potongan kode untuk halaman cetak laporan *omzet*.

```
1  public function print_result_omzet()
2      {
3          $data['active'] = 'laporan_omzet';
4          if ($this->session->userdata['logged_in']['level'] != "manager") {
5              redirect('manager/c_laporan');
6          }
7          $tanggal=$this->input->post('tanggal');
8          $bulan=$this->input->post('bulan');
9          $tahun=$this->input->post('tahun');
10         $pilihan=$this->input->post('pilihan');
11         $data['tanggal'] = $tanggal;
12         $data['bulan'] = $bulan;
13         $data['tahun'] = $tahun;
14         $data['pilihan'] = $pilihan;
15         $data['controller'] = $this->uri->segment(2);
16         $data['action'] = $this->uri->segment(3);
17         if($pilihan=="perhari"){
18             $data["transaksi"]=$this->m_laporan->daily_report($tanggal,$bulan,$tahun);
19             $data["judul"]="Laporan Data Transaksi per Tanggal " .$tanggal."-".$bulan."-".$tahun;
20             $data['pencarian']=$tanggal."-".$bulan."-".$tahun;
21         }
22         else {
23             $data["transaksi"]=$this->m_laporan->monthly_report($bulan,$tahun);
24             $data['judul']="Laporan Data Transaksi per Bulan " .$bulan."-".$tahun;
25             $data['pencarian']="x-".$bulan."-".$tahun;
26         }
27         $this->template->load('pure_template','laporan/v_print_hasil_lap_omzet', $data);
28     }
```

Gambar 5. 93 Potongan kode halaman cetak laporan *omzet*

Untuk keterangan gambar diatas, laporan *omzet* yang dipanggil berdasarkan pilihan jangka waktu seperti yang dimulai pada baris ke 17. Apabila sudah berhasil maka sistem akan membuka halaman *detail* laporan *omzet* sesuai dengan kode pada baris ke 27. Berbeda dengan laporan *omzet*, halaman ini menuju ke tampilan pure template yang siap untuk dicetak.

Pada Gambar 5.92 merupakan potongan kode untuk halaman laporan pengeluaran.

```
1  public function proc_lap_perlengkapan()
2      {
3          $data['active'] = 'laporan_perlengkapan';
4          if ($this->session->userdata['logged_in']['level'] != "manager") {
5              redirect('manager/c_laporan');
6          }
7          $tanggal=$this->input->post('tanggal');
8          $bulan=$this->input->post('bulan');
9          $tahun=$this->input->post('tahun');
10         $pilihan=$this->input->post('pilihan');
11         $data['tanggal'] = $tanggal;
12         $data['bulan'] = $bulan;
13         $data['tahun'] = $tahun;
14         $data['pilihan'] = $pilihan;
15         $data['controller'] = $this->uri->segment(2);
16         $data['action'] = $this->uri->segment(3);
17         if($pilihan=="perhari"){
18             $data["perlengkapan"]=$this->m_laporan->daily_report_perlengkapan($tanggal,$bulan,$tahun);
19             $data["judul"]="Laporan Data Perlengkapan per Tanggal " .$tanggal."-".$bulan."-".$tahun;
20             $data['pencarian']=$tanggal."-".$bulan."-".$tahun;
21         }
22         else {
23             $data["perlengkapan"]=$this->m_laporan->monthly_report_perlengkapan($bulan,$tahun);
24             $data['judul']="Laporan Data Perlengkapan per Bulan " .$bulan."-".$tahun;
25             $data['pencarian']="x-".$bulan."-".$tahun;
26         }
27         $this->template->load('sys_template_manager','laporan/v_hasil_lap_perlengkapan', $data);
28     }
```

Gambar 5. 94 Potongan kode halaman laporan pengeluaran

Untuk keterangan gambar diatas, laporan pengeluaran yang dipanggil berdasarkan pilihan jangka waktu seperti yang dimulai pada baris ke 17. Apabila sudah berhasil maka sistem akan membuka halaman *detail* laporan pengeluaran sesuai dengan kode pada baris ke 27.

Pada Gambar 5.95 merupakan potongan kode untuk halaman cetak laporan pengeluaran.

```
1   public function print_result_perlengkapan()
2       {
3           $data['active'] = 'laporan_perlengkapan';
4           if ($this->session->userdata['logged_in']['level'] != "manager") {
5               redirect('manager/c_laporan');
6           }
7           $tanggal=$this->input->post('tanggal');
8           $bulan=$this->input->post('bulan');
9           $tahun=$this->input->post('tahun');
10          $pilihan=$this->input->post('pilihan');
11          $data['tanggal'] = $tanggal;
12          $data['bulan'] = $bulan;
13          $data['tahun'] = $tahun;
14          $data['pilihan'] = $pilihan;
15          $data['controller'] = $this->uri->segment(2);
16          $data['action'] = $this->uri->segment(3);
17          if($pilihan=="perhari"){
18              $data["perlengkapan"]=$this->m_laporan->daily_report_perlengkapan($tanggal,$bulan,$tahun);
19              $data["judul"]="Laporan Data Perlengkapan per Tanggal " .$tanggal."-".$bulan."-".$tahun;
20              $data['pencarian']=$tanggal."-".$bulan."-".$tahun;
21          }
22          else {
23              $data["perlengkapan"]=$this->m_laporan->monthly_report_perlengkapan($bulan,$tahun);
24              $data['judul']="Laporan Data Perlengkapan per Bulan " .$bulan."-".$tahun;
25              $data['pencarian']="x-".$bulan."-".$tahun;
26          }
27          $this->template->load('pure_template','laporan/v_print_hasil_lap_perlengkapan', $data);
28      }
```

Gambar 5. 95 Potongan kode halaman cetak laporan pengeluaran

Untuk keterangan gambar diatas, laporan pengeluaran yang dipanggil berdasarkan pilihan jangka waktu seperti yang dimulai pada baris ke 17. Apabila sudah berhasil maka sistem akan membuka halaman *detail* laporan pengeluaran sesuai dengan kode pada baris ke 27. Berbeda dengan laporan pengeluaran, halaman ini menuju ke tampilan pure template yang siap untuk dicetak.

Pada Gambar 5.96 merupakan potongan kode untuk halaman laporan *profit*.

```
1   public function proc_lap_profit(){
2           $data['active'] = 'laporan_profit';
3           if ($this->session->userdata['logged_in']['level'] != "manager") {
4               redirect('manager/c_laporan');
5           }
6           $tanggal=$this->input->post('tanggal');
7           $bulan=$this->input->post('bulan');
8           $tahun=$this->input->post('tahun');
9           $pilihan=$this->input->post('pilihan');
10          $data['tanggal'] = $tanggal;
11          $data['bulan'] = $bulan;
12          $data['tahun'] = $tahun;
13          $data['pilihan'] = $pilihan;
14          $data['controller'] = $this->uri->segment(2);
15          $data['action'] = $this->uri->segment(3);
16          if($pilihan=="perhari"){
17              $data["transaksi"]=$this->m_laporan->daily_report($tanggal,$bulan,$tahun);
18              $data["perlengkapan"]=$this->m_laporan->daily_report_perlengkapan($tanggal,$bulan,$tahun);
19              $data["judul"]="Laporan Data Profit per Tanggal " .$tanggal."-".$bulan."-".$tahun;
20              $data['pencarian']=$tanggal."-".$bulan."-".$tahun;
21          } else {
22              $data["transaksi"]=$this->m_laporan->monthly_report($bulan,$tahun);
23              $data["perlengkapan"]=$this->m_laporan->monthly_report_perlengkapan($bulan,$tahun);
24              $data['judul']="Laporan Data Profit per Bulan " .$bulan."-".$tahun;
25              $data['pencarian']="x-".$bulan."-".$tahun;
26          }
27          $this->template->load('sys_template_manager','laporan/v_hasil_lap_profit', $data);
28      }
```

Gambar 5. 96 Potongan kode halaman laporan *profit*

Untuk keterangan gambar diatas, laporan *profit* yang dipanggil berdasarkan pilihan jangka waktu seperti yang dimulai pada baris ke 17. Apabila sudah berhasil maka sistem akan membuka halaman *detail* laporan *profit* sesuai dengan kode pada baris ke 27.

Pada Gambar 5.95 merupakan potongan kode untuk halaman cetak laporan *profit*.

```
1   public function print_result_profit(){
2           $data['active'] = 'laporan_profit';
3           if ($this->session->userdata['logged_in']['level'] != "manager") {
4               redirect('manager/c_laporan');
5           }
6           $tanggal=$this->input->post('tanggal');
7           $bulan=$this->input->post('bulan');
8           $tahun=$this->input->post('tahun');
9           $pilihan=$this->input->post('pilihan');
10          $data['tanggal'] = $tanggal;
11          $data['bulan'] = $bulan;
12          $data['tahun'] = $tahun;
13          $data['pilihan'] = $pilihan;
14          $data['controller'] = $this->uri->segment(2);
15          $data['action'] = $this->uri->segment(3);
16          if($pilihan=="perhari"){
17              $data["transaksi"]=$this->m_laporan->daily_report($tanggal,$bulan,$tahun);
18              $data["perlengkapan"]=$this->m_laporan->daily_report_perlengkapan($tanggal,$bulan,$tahun);
19              $data["judul"]="Laporan Data Profit per Tanggal " .$tanggal."-".$bulan."-".$tahun;
20              $data['pencarian']=$tanggal."-".$bulan."-".$tahun;
21          } else {
22              $data["transaksi"]=$this->m_laporan->monthly_report($bulan,$tahun);
23              $data["perlengkapan"]=$this->m_laporan->monthly_report_perlengkapan($bulan,$tahun);
24              $data['judul']="Laporan Data Profit per Bulan " .$bulan."-".$tahun;
25              $data['pencarian']="x-".$bulan."-".$tahun;
26          }
27          $this->template->load('pure_template','laporan/v_print_hasil_lap_profit', $data);
28      }
```

Gambar 5. 97 Potongan kode halaman cetak laporan *profit*

Untuk keterangan gambar diatas, laporan *profit* yang dipanggil berdasarkan pilihan jangka waktu seperti yang dimulai pada baris ke 17. Apabila sudah berhasil maka sistem akan membuka halaman *detail* laporan *profit* sesuai dengan kode pada baris ke 27. Berbeda dengan laporan *profit*, halaman ini menuju ke tampilan pure template yang siap untuk dicetak.

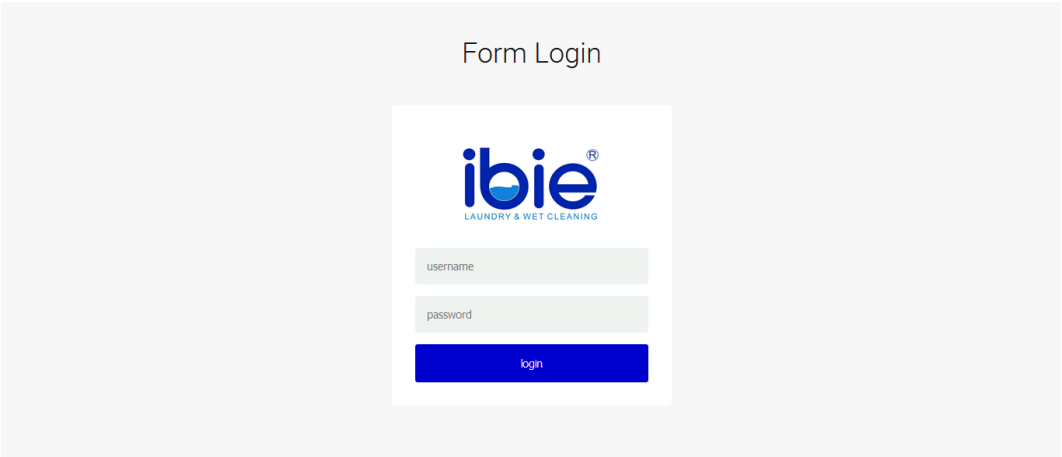
BAB VI
PENGUJIAN DAN PEMBAHASAN

6.1 Uji Coba Kemampuan Sistem

Pada bab ini akan membahas hasil dari penelitian dan pengujian sistem berdasarkan kemampuan sistem dalam memenuhi kebutuhan fungsional dan kebutuhan non-fungsional. Berikut merupakan hasil pengujian sistem yang telah dilakukan.

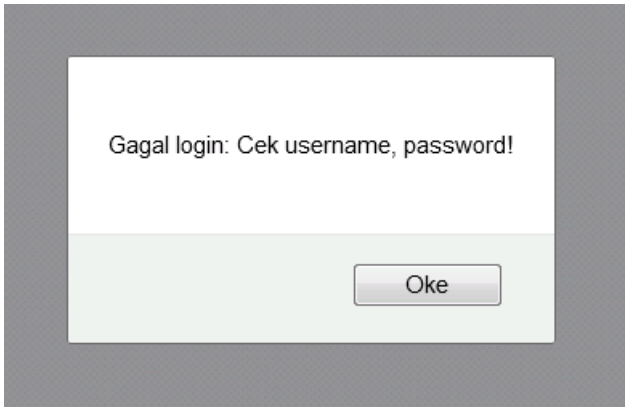
6.1.1 Halaman Login

Semua pengguna untuk dapat mengakses sistem memerluakan proses login. Dengan adanya proses login akan membuat orang-orang yang tidak punya hak mengakses sistem tidak bisa masuk ke sistem. Apabila pengguna belum login ataupun mencoba mengakses URL secara manual pada addres bar akan selalu berada di halaman login. Tampilan halaman login dapat dilihat pada Gambar 6.1.



Gambar 6. 1 Tampilan halaman login

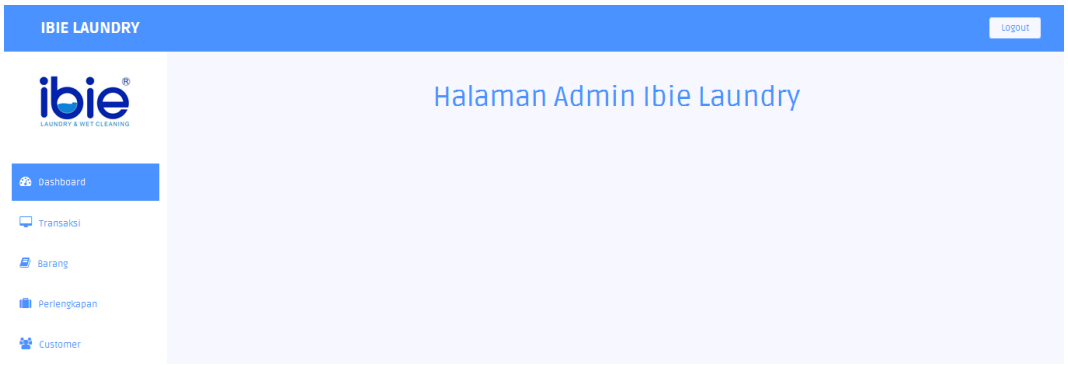
Pada saat akan *login*, pengguna diharuskan memasukkan *username* dan *password*. Jika dalam memasukkan kedua data tersebut terjadi kesalahan maka sistem akan menampilkan pesan peringatan seperti pada Gambar 6.2.



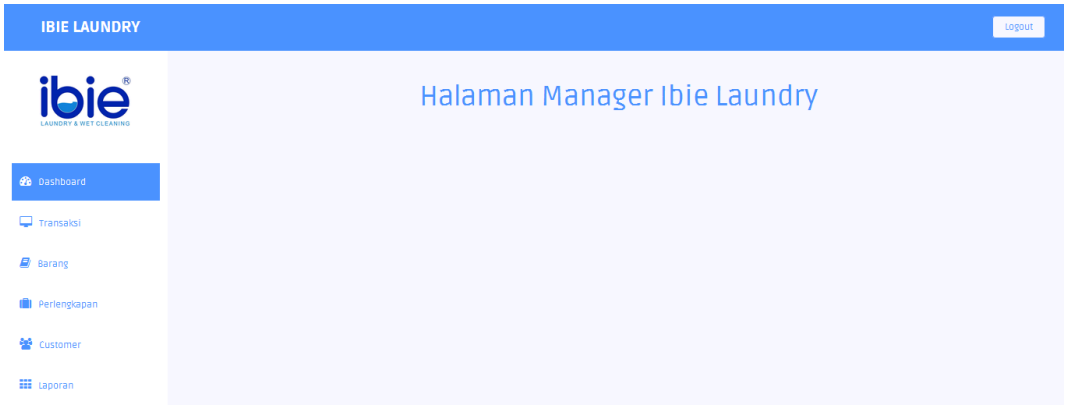
Gambar 6. 2 Tampilan halaman gagal login

6.1.2 Halaman Dashboard

Halaman *dashboard* merupakan halaman yang akan ditampilkan pada saat pengguna berhasil masuk kedalam sistem. Tampilan halaman *dashboard* pada *admin* dan *manager* hanya akan berbeda pada daftar menu yang ditampilkan di sebelah kiri dari halaman *dashboard*. Tampilan halaman *dashboard admin* dapat dilihat pada Gambar 6.3 dan tampilan halaman *dashboard manager* dapat dilihat pada Gambar 6.4.



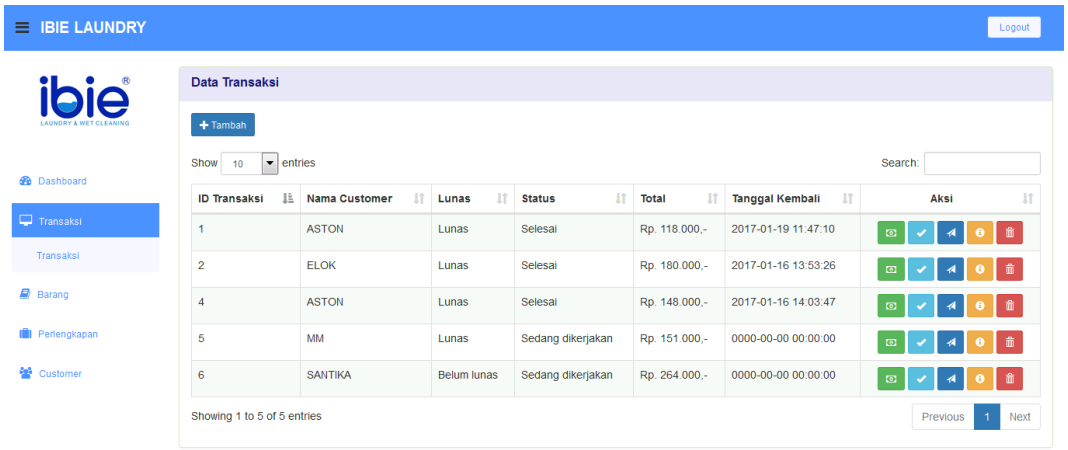
Gambar 6. 3 Tampilan halaman dashboard *admin*



Gambar 6. 4 Tampilan halaman dashboard *manager*

6.1.3 Halaman Transaksi

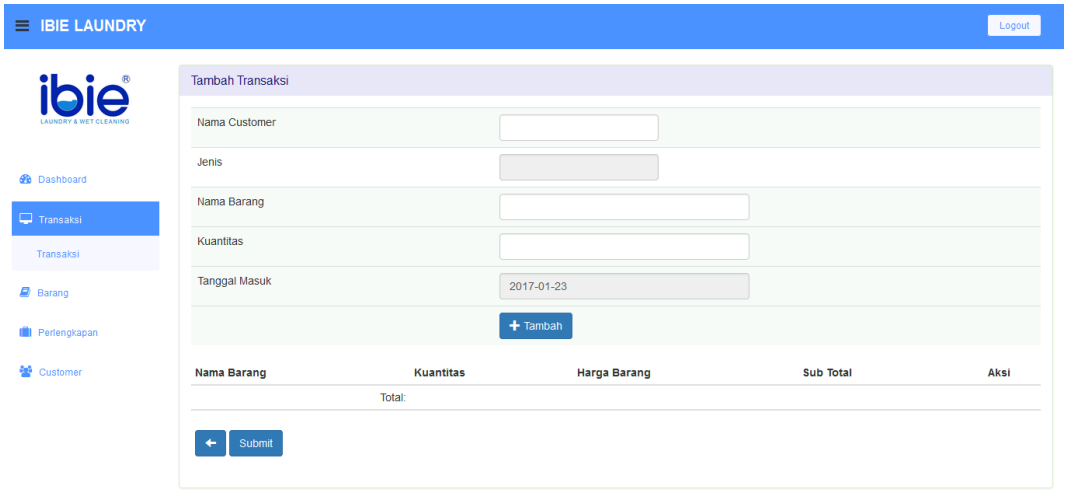
Halaman transaksi merupakan halaman yang digunakan untuk mengelola transaksi yang dilakukan. Halaman ini hanya dapat dikelola oleh *admin* saja. Pada halaman transaksi, sistem akan menampilkan data transaksi yang telah dikakukan. Tampilan data transaksi dapat dilihat pada Gambar 6.5.



Gambar 6. 5 Tampilan halaman transaksi

Pada halaman ini terdapat tombol-tombol aktif yang memiliki perannya masing-masing, diantaranya adalah tombol tambah, tombol lunas, tombol status, tombol kembalikan, tombol *detail* dan tombol hapus.

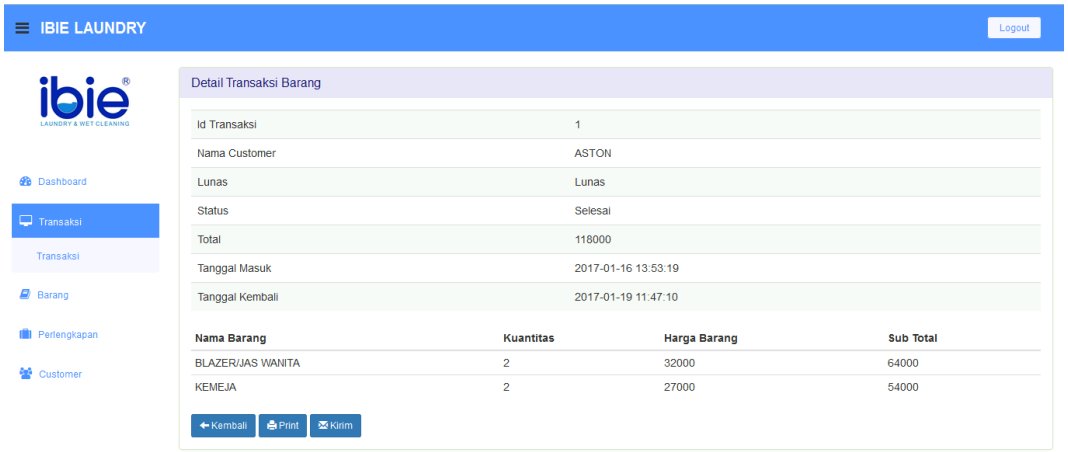
Tombol tambah digunakan untuk menambahkan transaksi yang akan dilakukan. Halaman tambah transaksi dapat dilihat pada Gambar 6.6.



Gambar 6. 6 Tampilan halaman tambah transaksi

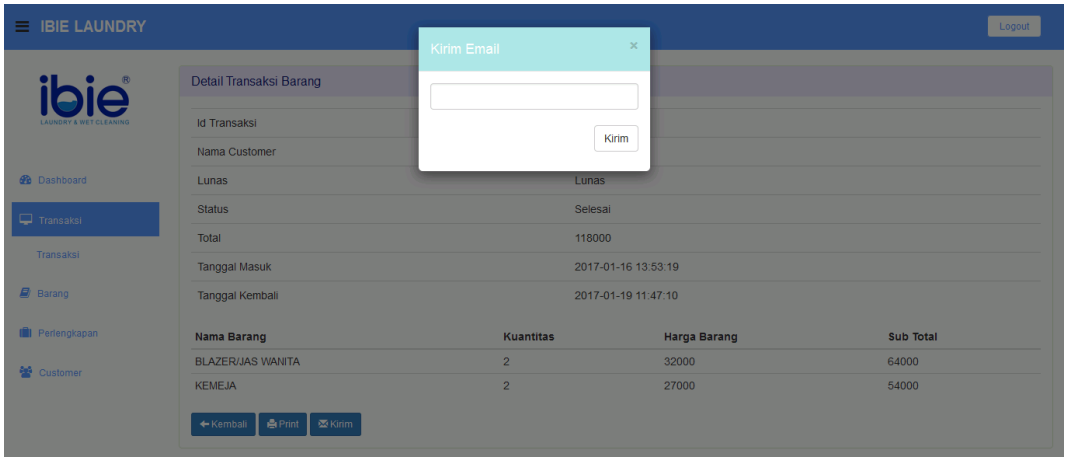
Setelah melakukan transaksi, sistem akan menuju ke halaman transaksi, kemudian akan diberikan penjelasan tombol-tombol berikutnya. Pada saat menekan tombol lunas, maka sistem akan mengupdate kolom lunas yang sebelumnya terdapat isian "belum lunas" menjadi "lunas". Sama seperti tombol lunas, pada saat menekan tombol status, maka sistem akan mengupdate kolom status yang sebelumnya terdapat isian "sedang dikerjakan" menjadi "selesai". Kemudian, tidak jauh berbeda dengan tombol sebelumnya, pada saat menekan tombol kembalikan, maka sistem akan mengupdate kolom tanggal kembali dengan tanggal dan waktu ketika menekan tombol kembalikan.

Tombol selanjutnya adalah tombol hapus, tombol ini digunakan untuk menghapus data transaksi yang telah dilakukan. Kemudian, untuk tombol *detail* transaksi digunakan untuk melihat halaman *detail* transaksi yang telah dilakukan. Halaman ini juga dapat diakses oleh *manager*. Halaman *detail* transaksi dapat dilihat pada Gambar 6.7.



Gambar 6. 7 Tampilan halaman *detail* transaksi

Didalam halaman *detail* transaksi terdapat tombol print dan kirim. Pada saat menekan tombol print, sistem akan melakukan proses cetak transaksi. *Manager* juga dapat mencetak *detail* transaksi. Kemudian untuk tombol kirim, sistem akan menampilkan form untuk mengisi *email customer* yang langsung bisa dikirim oleh *admin* kepada *customer*. Tampilan form kirim *email* dapat dilihat pada Gambar 6.8.

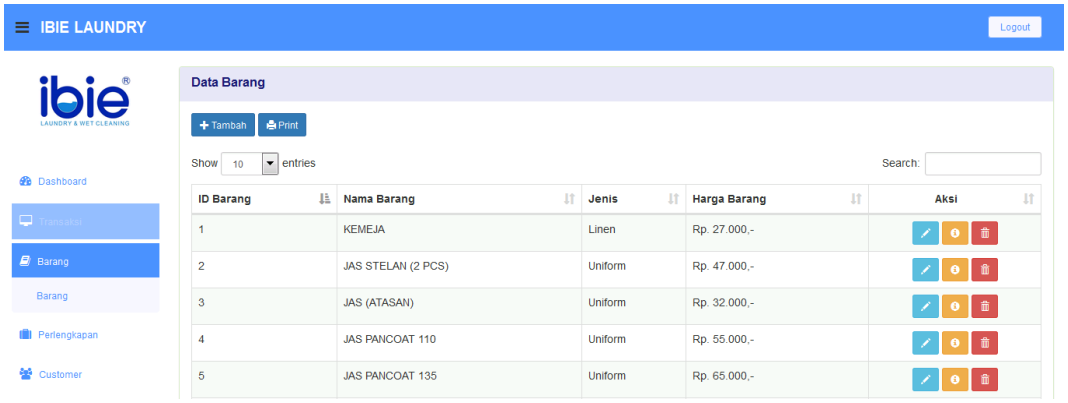


Gambar 6. 8 Tampilan halaman form kirim *email*

6.1.4 Halaman Barang

Halaman barang merupakan halaman yang digunakan untuk mengelola barang. Halaman ini hanya dapat dikelola oleh *admin* saja. Pada halaman barang,

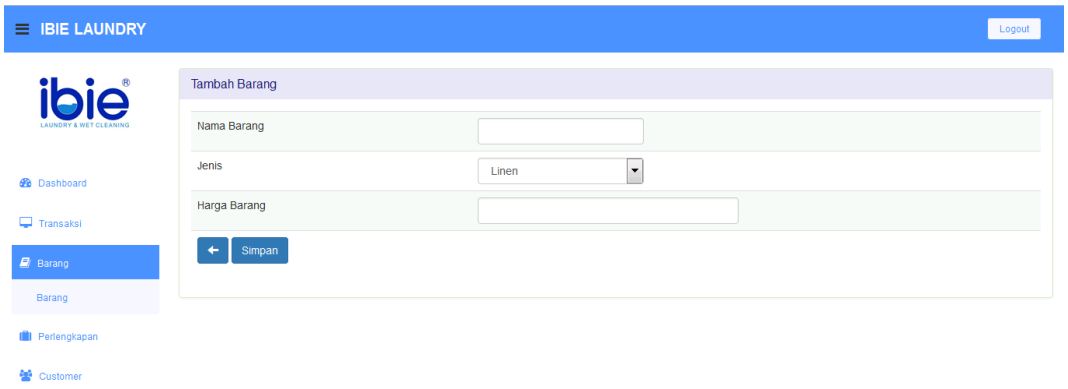
sistem akan menampilkan data barang. Tampilan data barang dapat dilihat pada Gambar 6.9.



Gambar 6. 9 Tampilan halaman barang

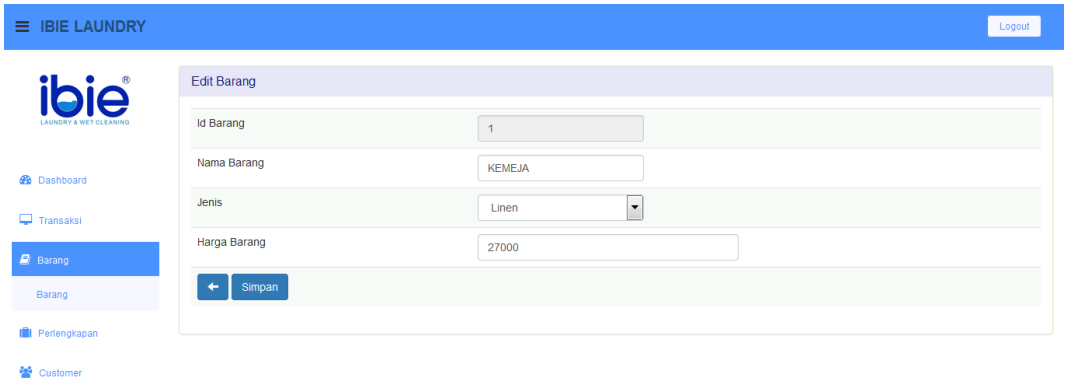
Pada halaman ini terdapat tombol-tombol aktif yang memiliki perannya masing-masing, diantaranya adalah tombol tambah, tombol ubah, tombol *detail*, tombol hapus dan tombol cetak.

Tombol tambah digunakan untuk menambahkan barang. Halaman tambah barang dapat dilihat pada Gambar 6.10.



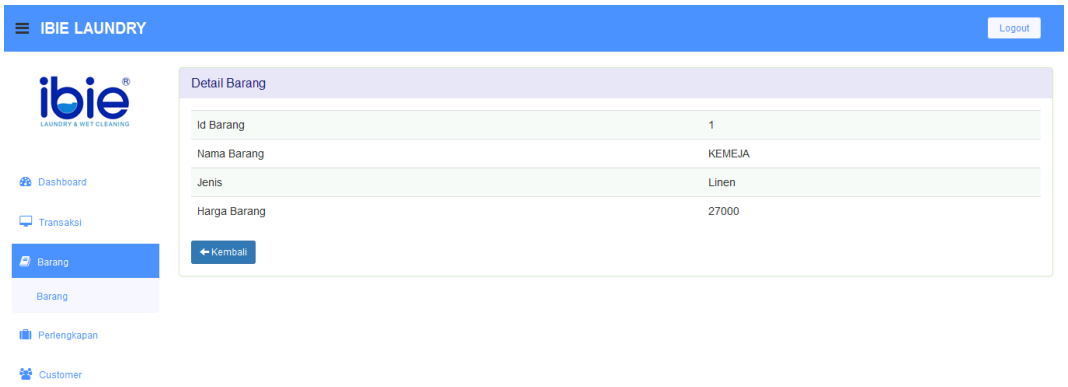
Gambar 6. 10 Tampilan halaman tambah barang

Pada saat menekan tombol ubah, tombol ini digunakan untuk mengubah data barang. Halaman ubah barang dapat dilihat pada Gambar 6.11.



Gambar 6. 11 Tampilan halaman ubah barang

Tombol *detail* digunakan untuk melihat *detail* barang. *Manager* dapat mengakses halaman ini. Halaman *detail* barang dapat dilihat pada Gambar 6.12.

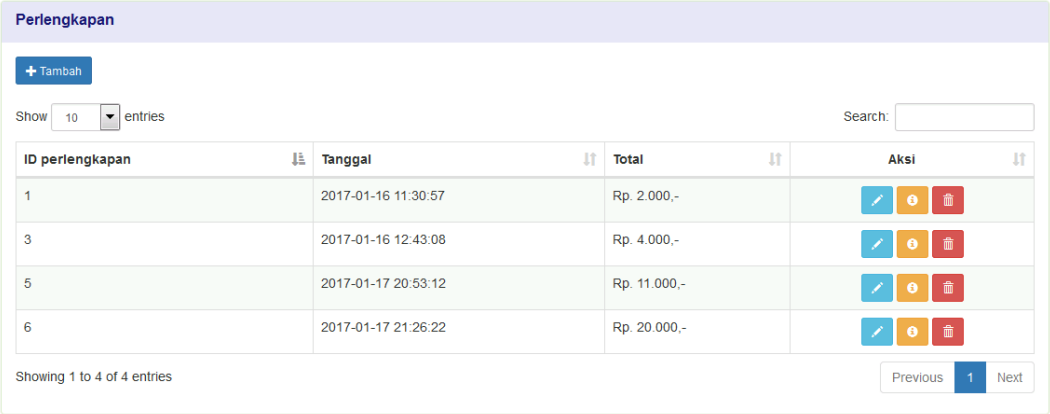


Gambar 6. 12 Tampilan halaman *detail* barang

Kemudian pada saat menekan tombol hapus, sistem akan menghapus data barang yang dipilih, dan kemudian ketika menekan tombol cetak, *admin* dan *manager* dapat mencetak data barang dan kemudian sistem akan melakukan proses cetak data barang.

6.1.5 Halaman Perlengkapan

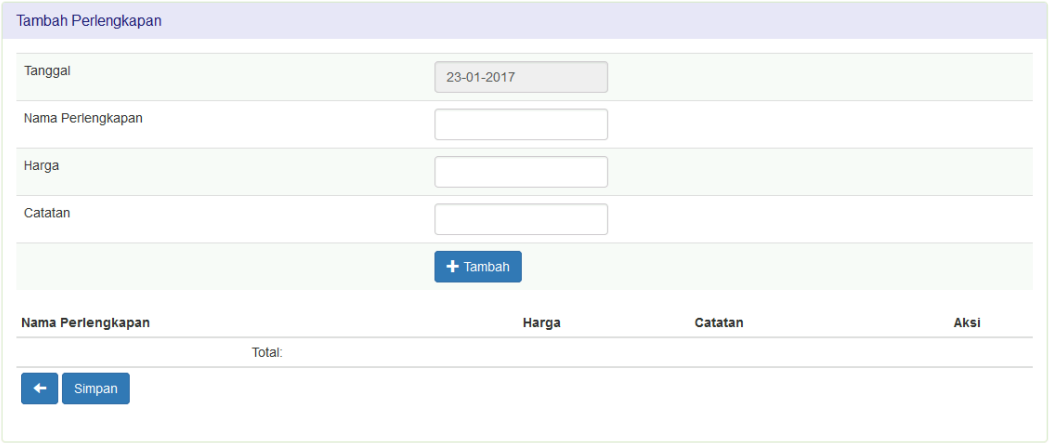
Halaman perlengkapan merupakan halaman yang digunakan untuk mengelola pembelian perlengkapan. Halaman ini dapat dikelola oleh *admin* dan *manager*. Pada halaman perlengkapan, sistem akan menampilkan data pembelian perlengkapan. Tampilan perlengkapan dapat dilihat pada Gambar 6.13.



Gambar 6. 13 Tampilan halaman perlengkapan

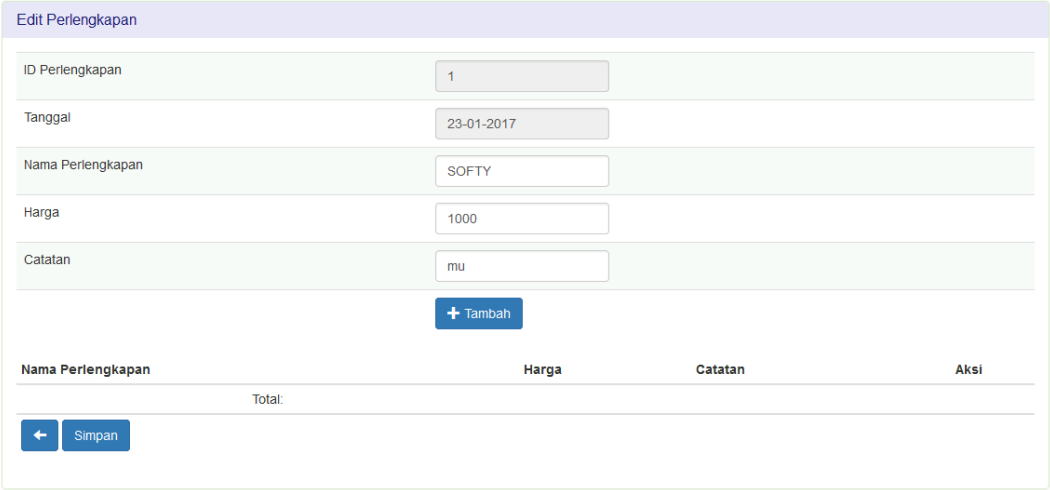
Pada halaman ini terdapat tombol-tombol aktif yang memiliki perannya masing-masing, diantaranya adalah tombol tambah, tombol ubah, tombol *detail*, tombol hapus dan tombol cetak.

Tombol tambah digunakan untuk menambahkan data pembelian perlengkapan. Halaman tambah perlengkapan dapat dilihat pada Gambar 6.14.



Gambar 6. 14 Tampilan halaman tambah perlengkapan

Pada saat menekan tombol ubah, tombol ini digunakan untuk mengubah data pembelian perlengkapan. Halaman ubah perlengkapan dapat dilihat pada Gambar 6.15.



Gambar 6. 15 Tampilan halaman ubah perlengkapan

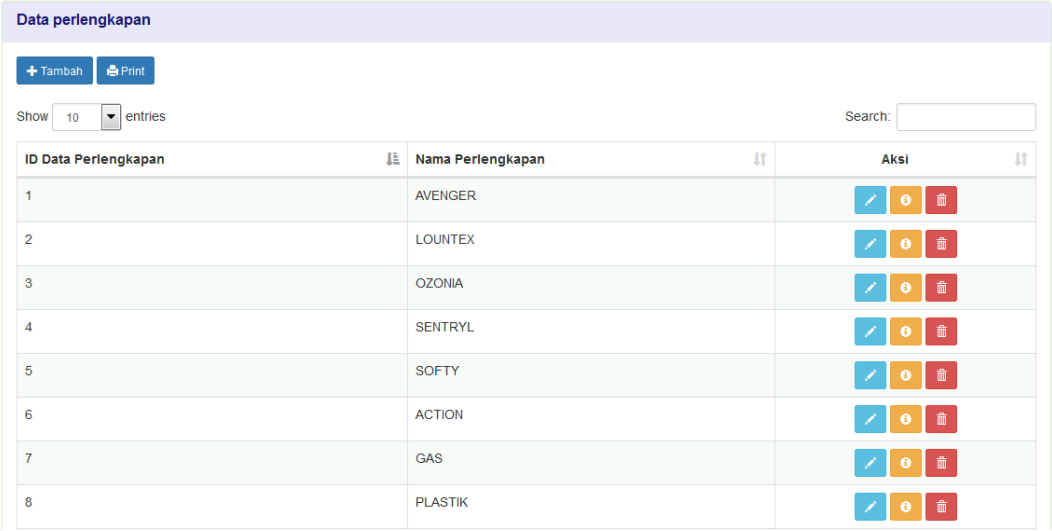
Tombol *detail* digunakan untuk melihat *detail* pembelian perlengkapan. Halaman *detail* perlengkapan dapat dilihat pada Gambar 6.16.

Gambar 6. 16 Tampilan halaman *detail* perlengkapan

Kemudian pada saat menekan tombol hapus, sistem akan menghapus data pembelian perlengkapan yang dipilih, dan kemudian ketika menekan tombol cetak, sistem akan melakukan proses cetak data pembelian perlengkapan.

6.1.6 Halaman Data Perlengkapan

Halaman data perlengkapan merupakan halaman yang digunakan untuk mengelola data perlengkapan. Halaman ini dapat dikelola oleh *admin* dan *manager*. Pada halaman data perlengkapan, sistem akan menampilkan data perlengkapan. Tampilan data perlengkapan dapat dilihat pada Gambar 6.17.



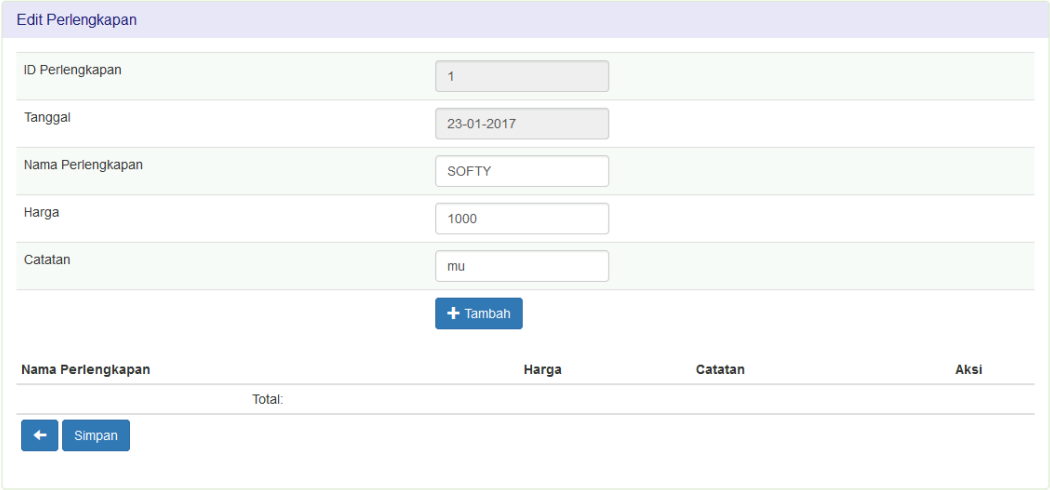
Gambar 6. 17 Tampilan halaman data perlengkapan

Pada halaman ini terdapat tombol-tombol aktif yang memiliki perannya masing-masing, diantaranya adalah tombol tambah, tombol ubah, tombol *detail*, tombol hapus dan tombol cetak.

Tombol tambah digunakan untuk menambahkan data perlengkapan. Halaman tambah data perlengkapan dapat dilihat pada Gambar 6.18.

Gambar 6. 18 Tampilan halaman tambah data perlengkapan

Pada saat menekan tombol ubah, tombol ini digunakan untuk mengubah data perlengkapan. Halaman ubah data perlengkapan dapat dilihat pada Gambar 6.19.



Gambar 6. 19 Tampilan halaman ubah data perlengkapan

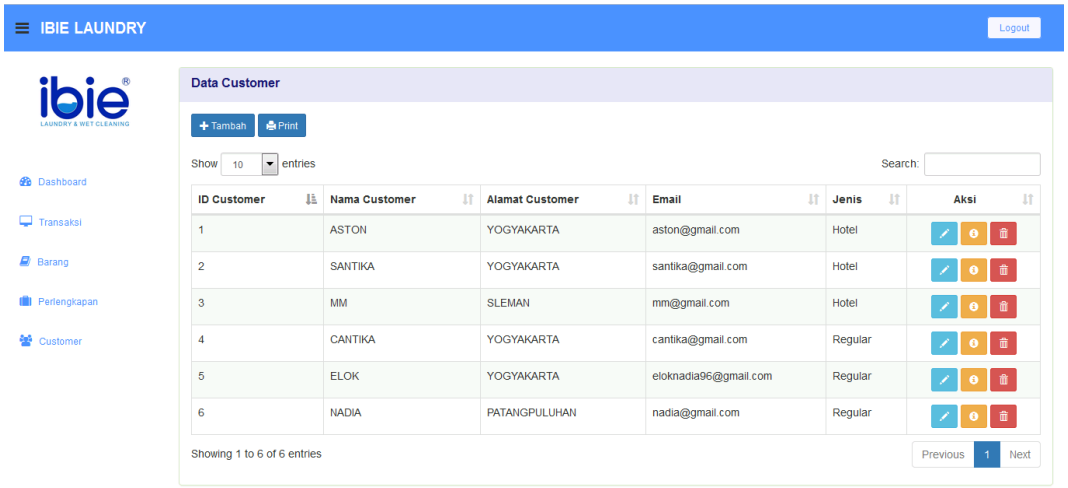
Tombol *detail* digunakan untuk melihat *detail* data perlengkapan. Halaman *detail* data perlengkapan dapat dilihat pada Gambar 6.20.

Gambar 6. 20 Tampilan halaman *detail* data perlengkapan

Kemudian pada saat menekan tombol hapus, sistem akan menghapus data perlengkapan yang dipilih, dan kemudian ketika menekan tombol cetak, sistem akan melakukan proses cetak data perlengkapan.

6.1.7 Halaman *Customer*

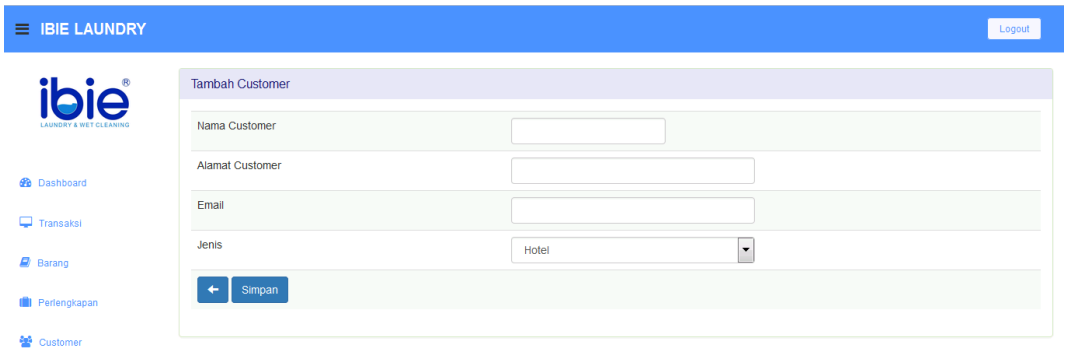
Halaman *customer* merupakan halaman yang digunakan untuk mengelola data *customer*. Halaman ini hanya dapat dikelola oleh *admin* saja. Pada halaman *customer*, sistem akan menampilkan data *customer*. Tampilan data *customer* dapat dilihat pada Gambar 6.21.



Gambar 6. 21 Tampilan halaman *customer*

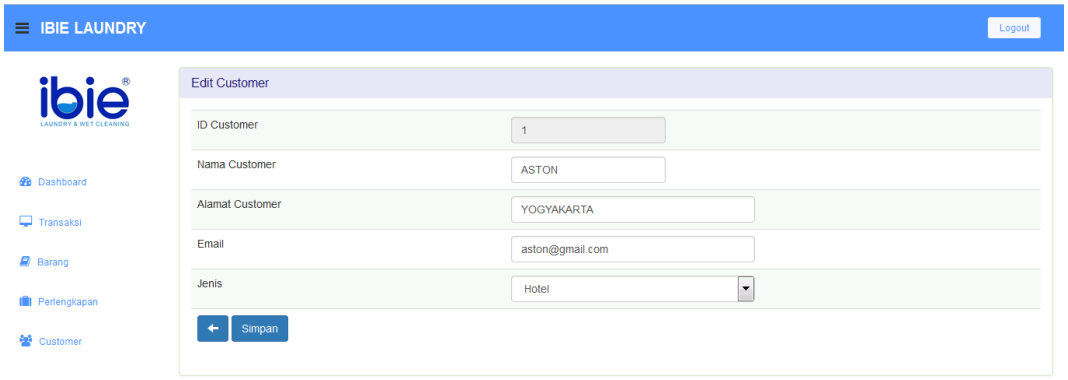
Pada halaman ini terdapat tombol-tombol aktif yang memiliki perannya masing-masing, diantaranya adalah tombol tambah, tombol ubah, tombol *detail*, tombol hapus dan tombol cetak.

Tombol tambah digunakan untuk menambahkan data *customer*. Halaman tambah *customer* dapat dilihat pada Gambar 6.22.



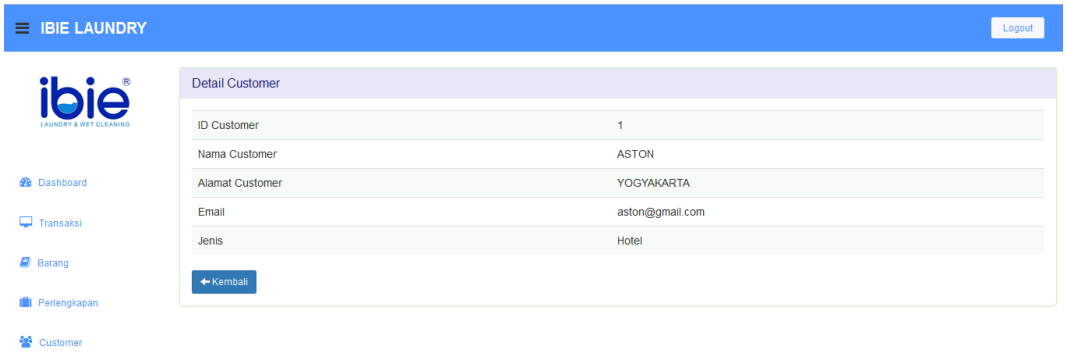
Gambar 6. 22 Tampilan halaman tambah *customer*

Pada saat menekan tombol ubah, tombol ini digunakan untuk mengubah data *customer*. Halaman ubah *customer* dapat dilihat pada Gambar 6.23.



Gambar 6. 23 Tampilan halaman ubah *customer*

Tombol *detail* digunakan untuk melihat *detail* data *customer*. *Manager* dapat mengakses halaman ini. Halaman *detail customer* dapat dilihat pada Gambar 6.24.

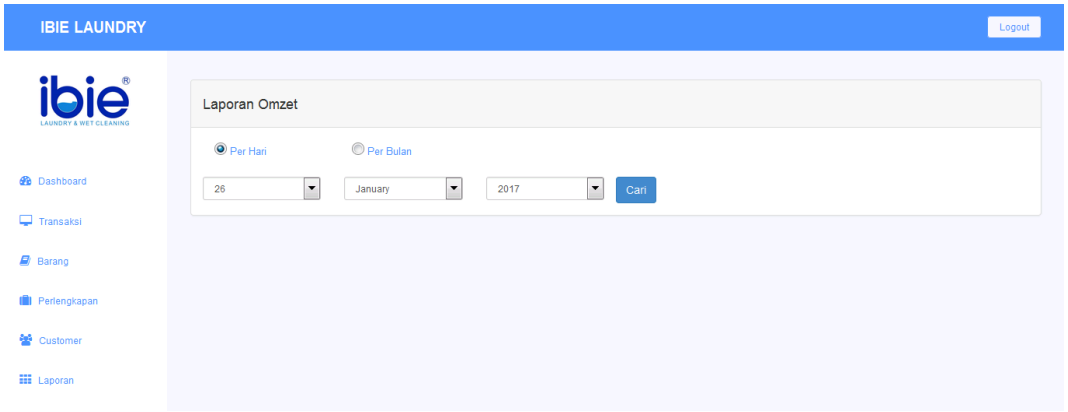


Gambar 6. 24 Tampilan halaman *detail customer*

Kemudian pada saat menekan tombol hapus, sistem akan menghapus data *customer* yang dipilih, dan kemudian ketika menekan tombol cetak, *admin* dan *manager* dapat mencetak data *customer* dan kemudian sistem akan melakukan proses cetak data *customer*.

6.1.8 Halaman Laporan *Omzet*

Halaman laporan *omzet* merupakan halaman yang menampilkan rekap data transaksi dalam perjangka waktu. Halaman ini hanya dapat diakses oleh *manager* saja. *Manager* dapat melihat riwayat transaksi yang sudah dilakukan. Jangka waktu yang dapat dipilih adalah perhari atau perbulan yang ditunjukkan dengan *radio button*. Tampilan halaman laporan *omzet* dapat dilihat pada Gambar 6.25.

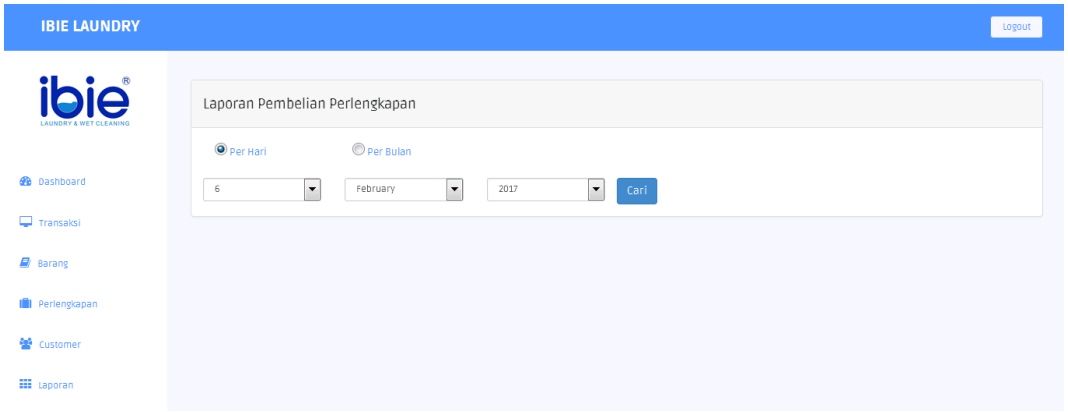


Gambar 6. 25 Tampilan halaman jangka waktu laporan *omzet*

Jika *manager* sudah menentukan jangka waktu yang akan dipilih, setelah itu tekan tombol cari, sehingga hasil laporan akan muncul dan akan sesuai dengan jangka waktu yang telah ditentukan. Tampilan hasil rekap transaksi dapat dilihat pada Gambar 6.26.

Gambar 6. 26 Halaman Laporan *Omzet*

Halaman laporan pengeluaran merupakan halaman yang menampilkan rekap data pembelian perlengkapan dalam perjangka waktu. Halaman ini hanya dapat diakses oleh *manager* saja. *Manager* dapat melihat riwayat pembelian perlengkapan yang sudah dilakukan. Jangka waktu yang dapat dipilih adalah perhari atau perbulan yang ditunjukkan dengan *radio button*. Tampilan halaman laporan pengeluaran dapat dilihat pada Gambar 6.27.



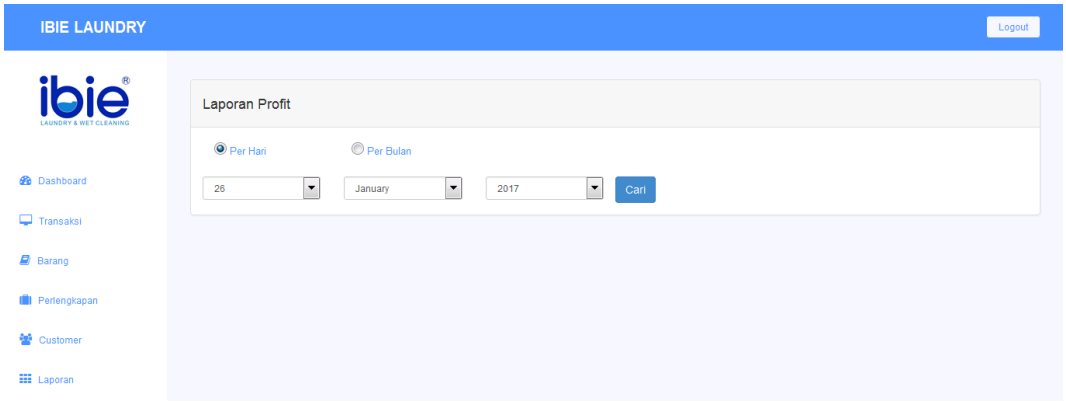
Gambar 6. 27 Tampilan halaman jangka waktu laporan pengeluaran

Jika *manager* sudah menentukan jangka waktu yang akan dipilih, setelah itu tekan tombol cari, sehingga hasil laporan akan muncul dan akan sesuai dengan jangka waktu yang telah ditentukan. Tampilan hasil rekap pembelian perlengkapan dapat dilihat pada Gambar 6.28.

Gambar 6. 28 Tampilan halaman laporan pengeluaran

6.1.9 Halaman Laporan *Profit*

Halaman laporan *profit* merupakan halaman yang menampilkan rekap *profit* dalam perjangka waktu. Halaman ini hanya dapat diakses oleh *manager* saja. *Manager* dapat melihat riwayat transaksi dan pembelian perlengkapan yang sudah dilakukan. Jangka waktu yang dapat dipilih adalah perhari atau perbulan yang ditunjukkan dengan *radio button*. Tampilan halaman laporan *profit* dapat dilihat pada Gambar 6.29.



Gambar 6. 29 Tampilan halaman jangka waktu laporan *profit*

Jika *manager* sudah menentukan jangka waktu yang akan dipilih, setelah itu tekan tombol cari, sehingga hasil laporan akan muncul dan akan sesuai dengan jangka waktu yang telah ditentukan. Tampilan hasil rekap *profit* dapat dilihat pada Gambar 6.30.

Gambar 6. 30 Tampilan halaman laporan *profit*

6.2 Pengujian Oleh Calon Pengguna Sistem

Pengujian sistem ini ditujukan untuk menemukan kendala yang mungkin terjadi ketika sistem ini mulai beroperasi. Dalam pengujian oleh calon pengguna sistem ini, diharapkan penulis mendapatkan saran dari pengguna yang bisa membuat sistem ini lebih optimal dan lebih sesuai dengan keinginan pengguna. Uji coba sistem ini dilakukan dengan dua jenis koresponden, yaitu yang pertama

adalah dari direktur dan yang kedua adalah *admin*. Hal ini dilakukan untuk meminta pendapat dan saran dari pihak yang berinteraksi langsung dengan sistem yang telah dibangun. Serta untuk menemukan kelemahan-kelamahan yang masih terdapat dalam sistem. Walaupun mungkin nantinya saran dan masukan dari responden berada diluar batasan masalah, hasil pengujian sistem ini akan tetap ditampung sebagai bahan pertimbangan ketika sistem ini akan dikembangkan lebih lanjut di masa mendatang.

6.3 Hasil Pengujian Sistem

Hasil rata-rata oleh calon pengguna sistem:

NO	Pertanyaan	Nilai
1	Apakah sistem yang dibangun mudah untuk digunakan?	5
2	Apakah sistem yang dibangun sudah sesuai requirement?	4.5
3	Apakah sistem yang dibangun sudah dapat membantu dan mempermudah pengelolaan data?	5
4	Apakah fitur kirim bukti transaksi membantu dan mempermudah dalam proses pembayaran?	4

Hasil dari korespondensi baik direktur dan *admin* memberikan respon yang positif terhadap sistem yang dibuat. Korespondensi mengklaim bahwa sistem informasi pengolahan transaksi laundry sudah mencukupi keinginan direktur dan *admin*. Sistem ini juga dapat mempermudah *admin* dalam proses transaksi sehingga pembuatan nota dapat lebih efektif dan menghemat waktu.

BAB VII
PENUTUP

7.1 Kesimpulan

Berdasarkan hasil penelitian dan hasil pengujian Sistem Informasi Pengolahan Transaksi Laundry pada bab sebelumnya, maka dapat ditarik kesimpulan sebagai berikut:

1. Sistem informasi pengolahan transaksi laundry berhasil dibangun untuk memudahkan proses yang ada dalam *Ibie Laundry & Wet Cleaning*.
2. Sistem ini memudahkan *admin* dalam proses pembuatan nota dan pengiriman nota langsung kepada *email customer*.
3. Sistem informasi pengolahan transaksi laundry ini memudahkan *manager* untuk merekap *omzet* dan *profit* dari *Ibie Laundry & Wet Cleaning*.
4. Secara keseluruan sistem telah mampu melaksanakan proses-proses yang telah dirancang pada tahap analisis dan perancangan serta memenuhi kebutuhan sistem.

Saran

Berikut beberapa saran yang penulis berikan untuk penelitian serupa dimasa mendatang agar memberikan hasil penelitian yang lebih baik:

1. Sistem dapat mengelompokkan *detail* transaksi *customer* menjadi per-bulan.
2. Sistem dapat mencetak laporan *profit* untuk beberapa hari.

DAFTAR PUSTAKA

Al-Fatah, Hanif., 2007, *Analisis dan Perancangan Sistem Informasi untuk Keungguan Bersaing Perusahaan dan Organisasi Modern*, Andi Offset, Yogyakarta.

Gordon, B.Davis., 2002, *Kerangka Dasar Sistem Informasi Manajemen*, PPM, Jakarta.

Jogiyanto, HM., 2005, *Analisis dan Desain Sistem Informasi*, ANDI, Yogyakarta.

Kadir, A., 2009, *Membuat Aplikasi Web dengan PHP dan Database MySql*,ANDI, Yogyakarta

Korth, H.F., Silberschatz, A., dan Sudarshan S., 1997, Database System Concepts, McGraw-Hill.

Nugroho, Aan., 2011, *Aplikasi Sistem Informasi pada Jasa Usaha Laundry*, Skripsi, Jurusan Teknik Informatika, Universitas Narotama. Surabaya.

Nurwono, Y. 1994. *Manajemen Informasi Pendekatan Globa*l. Jakarta: PT Elex Media Komputindo.

Pressman, R.S., 2010, *Software Engineering*: A Practitioner's Approach, edisi 7, McGrawHill Higher Education, New York.

Resig, John., (2006), *Pro JavaScript Techniques*, Apress, New York.

Rukmanasari, 2013, *Analisis dan Perancangan Sistem Informasi Jasa Laundry pada D&A Laundry dengan Menggunakan Visual Basic 6.0*, Skripsi, Jurusan Teknik Informatika, STIMIK AMIKOM, Yogyakarta.

Shneiderman, B., 1998, *Designing The User Interface: Strategic For Effective HCI*, 3rd Edition, Addison Wesley, Boston.

Supriyanto, A., 2005, *Pengantar Teknologi Informasi*, Salemba Infotek, Jakarta.

Sutabri, Tata., 2004., *Analisa Sistem Informasi*, Andi Offset, Yogyakarta.

Velnita, 2012, *Perancangan Sistem Informasi Jasa Laundry dengan Menggunakan Java Swing dan Persistenc API di Laundry Nareswari*, Skripsi, Jurusan Teknik Informatika, STIMIK AMIKOM, Yogyakarta.

Whitten, Jeffery, L., Bentley, Lonnie, D., Dittman, and Kevin, C., 2004, *Metode Desain dan Analisis Sistem*. ANDI, Yogyakarta.

http://www.getbootstrap.com, diakses pada tanggal 9 Januari 2017.

http://www.jquery.com, diakses pada tanggal 16 Januari 2017.

http://www.stackoverflow.com, diakses pada tanggal 2 Januari 2017.

http://www.w3schools.com, di akses pada tanggal 23 Januari 2017.

LAMPIRAN

Nama : Aphit Dedy N
Jabatan: Direktur

NO	Pertanyaan	Nilai
1	Apakah sistem yang dibangun mudah untuk digunakan?	5
2	Apakah sistem yang dibangun sudah sesuai requirement?	5
3	Apakah sistem yang dibangun sudah dapat membantu dan mempermudah pengelolaan data?	5
4	Apakah fitur kirim bukti transaksi membantu dan mempermudah dalam proses pembayaran?	4
5	Saran: Sudah bagus. Adanya laporan untuk *omzet* dan *profit* sangat membantu meringankan pekerjaan.	

Nilai
Sangat Setuju = 5
Setuju = 4
Cukup = 3
Kurang Setuju = 2
Tidak Setuju = 1

Nama : Geni Nurfitasari
Jabatan: *Admin*

NO	Pertanyaan	Nilai
1	Apakah sistem yang dibangun mudah untuk digunakan?	5
2	Apakah sistem yang dibangun sudah sesuai requirement?	4
3	Apakah sistem yang dibangun sudah dapat membantu dan mempermudah pengelolaan data?	5
4	Apakah fitur kirim bukti transaksi membantu dan mempermudah dalam proses pembayaran?	4
5	Saran: Proses transaksi sangat mudah dan praktis untuk digunakan sehingga tidak butuh waktu yang lama untuk pemuatan nota.	

Nilai
Sangat Setuju = 5
Setuju = 4
Cukup = 3
Kurang Setuju = 2
Tidak Setuju = 1