

LAMPIRAN D
SEGMENT PROGRAM

Segmen Program C.1 Membuat Transaksi Pembelian Step 2

```
01:         $barangs = json_decode($request->input('barang')) ??
02:         "[]";
03:
04:         $ppn_include = $request->input('ppn-include');
05:         $invoice = Session::get('invoice_cart');
06:
07:         $total = 0;
08:         $cleanTotal = 0; //total harga bersih, untuk
09: perhitungan harga non ppn
10:         $list_barang = []; //simpan data barang lengkap
11:         $total_list_barang = 0;
12:
13:         foreach ($barangs as $key => $barang) {
14:             $obj = Barang::find($barang->part);
15:             $obj->qty = $barang->qty;
16:             $price = Util::parseNumericValue($barang->harga);
17:             $obj->clean_price = $price;
18:
19:             if ($ppn_include != null) {
20:                 //kalau harga termasuk PPN
21:                 $ppnItem = $price - ($price / 100 * $invoice-
22: >PPN);
23:                 $obj->harga = $ppnItem;
24:                 $cleanTotal += $price * $barang->qty;
25:             }else{
26:                 $obj->harga = $price;
27:             }
28:
29:             $subtotal = $obj->harga * $barang->qty;
30:             $obj->subtotal = $subtotal;
31:
32:             $list_barang[] = $obj;
33:             $total_list_barang += $subtotal;
34:             $total += $subtotal;
35:         }
36:
37:         if (count($barangs) <= 0) {
38:             return redirect()->back()->withErrors([
39:                 'msg' => 'Minimal membeli 1 barang / paket !'
40:             ]);
41:         }
42:
43:         $invoice->list_barang = $list_barang;
44:         $invoice->total_list_barang = $total_list_barang;
45:         $invoice->total = $total;
46:
47:         if ($ppn_include != null){
48:             //kalau harga termasuk ppn
49:             $invoice->PPN_value = ($cleanTotal / 100) *
50: $invoice->PPN;
51:             $invoice->PPN_included = true;
52:         }else{
53:             $invoice->PPN_value = ($total / 100) * $invoice-
54: >PPN;
55:             $invoice->PPN_included = false;
```

Segmen Program C.1 (Lanjutan)

```
56:         }
57:         $invoice->grandTotal = $total + $invoice->PPN_value;
58:
59:         Session::put('invoice_cart', $invoice);
60:         return redirect('/invoice/confirmation');
```

Segmen Program C.2 Membuat Transaksi Pembelian Step 3

```
01:         $komisiJumlah = 0;
02:         $komisiPenerima = "-";
03:         $karyawan = Session::get('user');
04:         $jatuhTempo = $request->input('jatuhTempo');
05:         $invoice = Session::get('invoice_cart');
06:
07:         //cek apakah transaksi lama
08:         $timePembayaran = $request-
09: >input('timeValuePembayaran');
10:         $timeCreation = $request->input('timePembuatan') ??
11: Carbon::now();
12:
13:         if ($timePembayaran == null) {
14:             if (Carbon::parse($jatuhTempo)->isPast()) {
15:                 toast('Tanggal Jatuh Tempo Minimal Besok',
16:
17: 'error');
18:                 return back()->withErrors([
19:                     'msg' => 'Tanggal Jatuh Tempo Minimal
20: Besok !'
21:                 ]);
22:             }
23:         }
24:
25:         $komisiStatus = $request->input('komisi');
26:         if ($komisiStatus) {
27:             $komisiJumlah = Util::parseNumericValue($request-
28: >input('komisiJumlah'));
29:             $komisiPenerima = $request-
30: >input('komisiPenerima');
31:         }
32:
33:         DB::beginTransaction();
34:         try {
35:             $kode = Util::generateInvoiceCode();
36:             $suratJalan =
37: Util::generateSuratJalanCodeFromInvoiceCode($kode);
38:
39:             $currentDateTime = Carbon::now();
40:             //insert header
41:             $lastId = DB::table('hinvoice')->insertGetId([
42:                 'customer_id' => $invoice->customer->id,
43:                 'karyawan_id' => $karyawan->id,
44:                 'kode' => $kode,
45:                 'surat_jalan' => $suratJalan,
46:                 'total' => $invoice->total,
47:                 'contact_person' => $komisiPenerima,
48:                 'komisi' => $komisiJumlah,
49:                 'ppn' => $invoice->PPN,
50:                 'ppn_value' => $invoice->PPN_value,
51:                 'grand_total' => $invoice->grandTotal,
52:                 'po' => $request->input('po'),
53:                 'jatuh_tempo' => $jatuhTempo,
54:                 'created_at' => $timeCreation,
55:                 'paid_at' => $timePembayaran,
```

Segmen Program C.2 (Lanjutan)

```
56:         'paid_by' => $timePembayaran != null ?
57:     Session::get('user')->id : null,
58:         'confirmed_at' => $timeCreation,
59:         'confirmed_by' => $karyawan->id,
60:         'status'=> 1,
61:     ];
62:
63:     foreach ($invoice->list_barang as $key => $value)
64:     {
65:         DB::table('dinvoice')->insert([
66:             'hinvoice_id' => $lastId,
67:             'part' => $value->part,
68:             'nama' => $value->nama,
69:             'harga' => $value->harga,
70:             'qty' => $value->qty,
71:             'subtotal' => $value->subtotal,
72:             'type' => 'barang',
73:             'created_at' => $currentDateTime
74:         ]);
75:     }
76:
77:     foreach ($invoice->list_paket as $key => $value) {
78:         DB::table('dinvoice')->insert([
79:             'hinvoice_id' => $lastId,
80:             'part' => $value->id,
81:             'nama' => $value->nama,
82:             'harga' => $value->harga,
83:             'qty' => $value->qty,
84:             'subtotal' => $value->subtotal,
85:             'type' => 'paket',
86:             'created_at' => $currentDateTime
87:         ]);
88:     }
89:
90:     Session::remove('invoice_cart');
91:     toast("Transaksi Customer: ".$invoice->customer-
92: >nama.", Berhasil dibuat", 'success');
93:     DB::commit();
94:     return redirect('/invoice');
95: } catch (\Exception $ex) {
96:     DB::rollBack();
97:     return back()->withErrors([
98:         'msg' => $ex->getMessage()
99:     ]);
100: }
```