**LAMPIRAN D**

**SEGMEN PROGRAM**

**Segmen Program C.1 Membuat Transaksi Pembelian Step 2**

1. $barangs = json\_decode($request->input('barang') ??
2. "[]");
3. $ppn\_include = $request->input('ppn-include');
4. $invoice = Session::get('invoice\_cart');
5. $total = 0;
6. $cleanTotal = 0; //total harga bersih, untuk
7. perhitungan harga non ppn
8. $list\_barang = []; //simpan data barang lengkap
9. $total\_list\_barang = 0;
10. foreach ($barangs as $key => $barang) {
11. $obj = Barang::find($barang->part);
12. $obj->qty = $barang->qty;
13. $price = Util::parseNumericValue($barang->harga);
14. $obj->clean\_price = $price;
15. if ($ppn\_include != null) {
16. //kalau harga termasuk PPN
17. $ppnItem = $price - ($price / 100 \* $invoice-
18. >PPN);
19. $obj->harga = $ppnItem;
20. $cleanTotal += $price \* $barang->qty;
21. }else{
22. $obj->harga = $price;
23. }
24. $subtotal = $obj->harga \* $barang->qty;
25. $obj->subtotal = $subtotal;
26. $list\_barang[] = $obj;
27. $total\_list\_barang += $subtotal;
28. $total += $subtotal;
29. }
30. if (count($barangs) <= 0) {
31. return redirect()->back()->withErrors([
32. 'msg' => 'Minimal membeli 1 barang / paket !'
33. ]);
34. }
35. $invoice->list\_barang = $list\_barang;
36. $invoice->total\_list\_barang = $total\_list\_barang;
37. $invoice->total = $total;
38. if ($ppn\_include != null){
39. //kalau harga termasuk ppn
40. $invoice->PPN\_value = ($cleanTotal / 100) \*
41. $invoice->PPN;
42. $invoice->PPN\_included = true;
43. }else{
44. $invoice->PPN\_value = ($total / 100) \* $invoice-
45. >PPN;
46. $invoice->PPN\_included = false;

**Segmen Program C.1 (Lanjutan)**

1. }
2. $invoice->grandTotal = $total + $invoice->PPN\_value;
3. Session::put('invoice\_cart', $invoice);
4. return redirect('/invoice/confirmation');

**Segmen Program C.2 Membuat Transaksi Pembelian Step 3**

1. $komisiJumlah = 0;
2. $komisiPenerima = "-";
3. $karyawan = Session::get('user');
4. $jatuhTempo = $request->input('jatuhTempo');
5. $invoice = Session::get('invoice\_cart');
6. //cek apakah transaksi lama
7. $timePembayaran = $request-
8. >input('timeValuePembayaran');
9. $timeCreation = $request->input('timePembuatan') ??
10. Carbon::now();
11. if ($timePembayaran == null) {
12. if (Carbon::parse($jatuhTempo)->isPast()) {
13. toast('Tanggal Jatuh Tempo Minimal Besok',
14. 'error');
15. return back()->withErrors([
16. 'msg' => 'Tanggal Jatuh Tempo Minimal
17. Besok !'
18. ]);
19. }
20. }
21. $komisiStatus = $request->input('komisi');
22. if ($komisiStatus) {
23. $komisiJumlah = Util::parseNumericValue($request-
24. >input('komisiJumlah'));
25. $komisiPenerima = $request-
26. >input('komisiPenerima');
27. }
28. DB::beginTransaction();
29. try {
30. $kode = Util::generateInvoiceCode();
31. $suratJalan =
32. Util::generateSuratJalanCodeFromInvoiceCode($kode);
33. $currentDateTime = Carbon::now();
34. //insert header
35. $lastId = DB::table('hinvoice')->insertGetId([
36. 'customer\_id' => $invoice->customer->id,
37. 'karyawan\_id' => $karyawan->id,
38. 'kode' => $kode,
39. 'surat\_jalan' => $suratJalan,
40. 'total' => $invoice->total,
41. 'contact\_person' => $komisiPenerima,
42. 'komisi' => $komisiJumlah,
43. 'ppn' => $invoice->PPN,
44. 'ppn\_value' => $invoice->PPN\_value,
45. 'grand\_total' => $invoice->grandTotal,
46. 'po' => $request->input('po'),
47. 'jatuh\_tempo' => $jatuhTempo,
48. 'created\_at' => $timeCreation,
49. 'paid\_at' => $timePembayaran,

**Segmen Program C.2 (Lanjutan)**

1. 'paid\_by' => $timePembayaran != null ?
2. Session::get('user')->id : null,
3. 'confirmed\_at' => $timeCreation,
4. 'confirmed\_by' => $karyawan->id,
5. 'status'=> 1,
6. ]);
7. foreach ($invoice->list\_barang as $key => $value)
8. {
9. DB::table('dinvoice')->insert([
10. 'hinvoice\_id' => $lastId,
11. 'part' => $value->part,
12. 'nama' => $value->nama,
13. 'harga' => $value->harga,
14. 'qty' => $value->qty,
15. 'subtotal' => $value->subtotal,
16. 'type' => 'barang',
17. 'created\_at' => $currentDateTime
18. ]);
19. }
20. foreach ($invoice->list\_paket as $key => $value) {
21. DB::table('dinvoice')->insert([
22. 'hinvoice\_id' => $lastId,
23. 'part' => $value->id,
24. 'nama' => $value->nama,
25. 'harga' => $value->harga,
26. 'qty' => $value->qty,
27. 'subtotal' => $value->subtotal,
28. 'type' => 'paket',
29. 'created\_at' => $currentDateTime
30. ]);
31. }
32. Session::remove('invoice\_cart');
33. toast("Transaksi Customer: ".$invoice->customer-
34. >nama.", Berhasil dibuat", 'success');
35. DB::commit();
36. return redirect('/invoice');
37. } catch (\Exception $ex) {
38. DB::rollBack();
39. return back()->withErrors([
40. 'msg' => $ex->getMessage()
41. ]);
42. }