

Manual Teknikal Toplevel ERP (Developer)

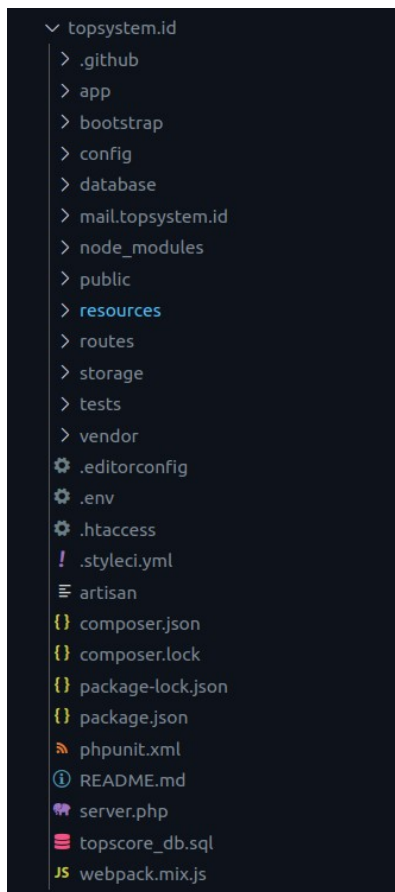
A) Teknologi

Berikut adalah teknologi yang digunakan dalam membangun Toplevel

1. Cloud Server : Sistem Operasi Ubuntu 20.04 LTS dan CyberPanel 2.3.1
2. HTTP Server : OpenLite Speed dan Apache
3. Database : MySQL InnoDB
4. PHP Server : 7.3 (MySQL)
5. Toplevel.id : Laravel 8 PHP 8.0 + jQuery 3.6.0
6. Topscore.id : Laravel 8 PHP 8.0 + jQuery 3.6.0
7. Reseller : Laravel 8 PHP 8.0 + jQuery 3.6.0
8. Investor/Urban : Laravel 8 PHP 8.0 + jQuery 3.6.0
9. API : Laravel 8 PHP 8.0
10. Frontend Template : Metronic 8.1 Demo 1 dan Demo 2

B) Struktur Proyek

Karena seluruh sistem dibuat menggunakan framework Laravel dengan versi yang sama maka akan memiliki struktur yang sama seperti pada gambar dibawah:



4 folder yang paling sering berinteraksi dalam proses development adalah /app , /public , /resources dan /routes.

Folder /app berisi script Controller utama yang menjalankan berbagai function termasuk export, import serta model.

Folder /public berisi asset berupa gambar dan hasil upload oleh user, jika pengelolaan website disimpan pada sub folder /api.

Folder /resources berisi file views untuk mengatur frontend atau tampilan utama termasuk script jQuery.

Folder /routes berisi file untuk mengatur routing request data dari view ke controller.

Serta 1 file .env yang digunakan untuk mengatur data login ke MySQL server dan Email server.

C) Penjelasan Sistem

1) Toplevel.id

Sistem ini membantu operasional perusahaan mulai dari pengelolaan website toplevel.id, keluar masuk stok pada warehouse/inventory, penjualan online dan offline, data API kerjasama pihak external, tracking aktifitas user, manajemen data customer dan user sistem, laporan penjualan dan stok, manajemen integrasi stok marketplace dengan internal stok, rating kepuasan customer, voting system, reseller system, user urban system.

Pada pengelolaan server CyberPanel topscore.id:8090, nama website proyek adalah Topssystem.id, dengan penjelasan sebagai berikut:

* Note :

- Agar lebih mudah dalam membaca komunikasi antara ajax jQuery frontend dengan function pada backend silahkan lihat file pada routes/web.php.
- Semua penjelasan menu dibawah adalah pada sisi administrator, karena semua fitur yang ada pada divisi lain sudah ada disini.
- Mungkin terdapat perbedaan di beberapa file, dimana saya ada yang menggunakan app/Models sebagai CRUD ataupun Eloquent, dan ada juga yang menggunakan DB:: query builder manual untuk performa dan mempercepat proses development.
- Cara saya memanggil file javascript pada views ada pada structure.blade.php, dimana saya membaca segment url yang sedang aktif, jika terdeteksi aktif kemudian akan saya include sesuai file jsnya, jika dirasa cara ini agak menghambat ketika sudah terlalu banyak file, mungkin bisa dimodifikasi untuk langsung dihubungkan pada group folder file.

app/Http/Controllers	resources/views/	database	keterangan
AuthController.php	auth/	ts_users ts_user_groups ts_groups	Login user.
BrandController.php	app/brand	ts_brands	Pengelolaan data brand. * brand
ProductUnitController.php	app/product_unit	ts_product_units	Pengelolaan satuan artikel. * satuan artikel
GenderController.php	app/gender	ts_genders	Pengelolaan gender artikel. * gender
SeasonController.php	app/season	ts_seasons	Pengelolaan season artikel. * season
ProductCategory.php	app/product_category	ts_product_categories	Pengelolaan kategori artikel. * kategori artikel
ProductSubCategory.php	app/product_sub_category	ts_product_sub_categories	Pengelolaan sub kategori artikel. * sub kategori artikel
ProductSubSubCategory.php	app/product_sub_sub_category	ts_product_sub_sub_categories	Pengelolaan sub sub kategori artikel. * sub-sub kategori artikel
MainColorController.php	app/main_color	ts_main_colors	Pengelolaan warna artikel. * warna artikel
SizeController.php	app/size	ts_sizes	Pengelolaan size artikel, masing – masing size

			memiliki sub kategori tersendiri. * size artikel
ProductController.php	app/product	ts_products	Pengelolaan data artikel, master data utama yang digunakan kedalam berbagai bentuk system. Khusus website topscore.id jika artikel tidak memiliki gambar maka tidak akan tampil pada web. * data artikel
StoreTypeController.php StoreTypeDivisionController.php	app/ store_type_division	ts_store_types ts_store_type_divisions	Pengelolaan data divisi dan sub dari divisi tersebut, store types biasanya untuk menentukan tipe invoice online atau offline, yang biasa terpakai saat ini masih stt_id 1 dan 2 atau online dan offline, kemudian store type divisions adalah penjabaran dari invoice pada tabel pos_transactions, apakah ini divisi shopee, tokped, dll sesuai induk utama divisi. Selain itu ts_store_types juga berelasi pada data user, dimana untuk membedakan antara POS Online dan Offline. * data divisi
StoreController.php WarehouseController.php	app/store	ts_stores ts_warehouses	Pengelolaan data store, tabel store berelasi dengan tabel warehouse, sebenarnya tabel warehouse tidak pernah terpakai, dulu konsep awalnya adalah untuk mengetahui store A atau B berasal dari warehouse mana, namun hingga sekarang konsep tak terealisasi, sehingga abaikan saja untuk warehouse. Tabel store ini juga berelasi pada

			ts_pos_transactions untuk membedakan invoice sesuai store masing-masing, juga setup BIN / lokasi penyimpanan gudang pada ts_product_locations, serta data user, bertugas sesuai store masing – masing. * data store
ProductSupplierController.php	app/product_supplier	ts_product_suppliers	Pengelolaan data suppliers, yang berelasi dengan fitur atau tabel PO ts_purchase_orders dan juga Data Artikel atau ts_products. * data supplier
ProductLocationController.php	app/product_location	ts_product_locations	Pengelolaan data lokasi simpan barang / bin, akan dibedakan berdasarkan store alias berelasi dengan ts_stores, pada tabel ini ada 1 kolom Default Penerimaan, untuk menentukan default BIN ketika barang datang dari PO telah di approve, otomatis stok akan tersetup ke BIN default ini. * lokasi simpan
PaymentMethodController.php	app/payment_method	ts_payment_methods	Pengelolaan data metode pembayaran, berelasi dengan tabel pos_transactions pm_id pada POS App. * metode pembayaran
CourierController.php	app/courier	ts_couriers	Pengelolaan data kurir, berelasi dengan tabel ts_pos_transactions, untuk sales Online menentukan kurir dan ongkos kirim pada POS, namun khusus website id ini tidak dipanggil, tetapi digantikan dengan kolom pos_courier

			berupa string karena menggunakan API dan courier bawaan raja ongkir. * kurir pengiriman
TaxController.php	app/tax	ts_taxes	Pengelolaan data pajak, namun tidak pernah dipakai hingga sekarang, dulu konsep awal untuk menentukan pajak pada PO. * data pajak
StockTypeController.php	app/stock_type	ts_stock_types	Pengelolaan data tipe stok, berelasi dengan penerimaan PO, digunakan sebagai id untuk memisahkan tipe asset pada dashboard, Cash Credit / Consignment/ dst. * tipe stok
PurchaseOrderController.php	app/purchase_order	ts_purchase_orders ts_purchase_order_articles ts_purchase_order_article_details ts_products ts_product_stocks	Pengelolaan data PO, dimana kolom po_disc, po_extra_disc dan po_sub_disc pada ts_purchase_orders 'tidak' terpakai, Ada 5 tabel yang terlibat pada fitur ini, untuk menentukan artikel serta detail sizenya, dll. * pembelian
PurchaseOrderReceiveController.php	app/purchase_order_receive	ts_purchase_order_article_detail_statuses	Pengelolaan penerimaan PO, pada fitur ini cukup panggil data PO sebelumnya tanpa bisa edit, namun harus menentukan beberapa kriteria seperti tipe stok, invoice, tanggal, dll. Data penerimaan akan disimpan pada database tersebut. Pengambilan HPP ada pada tabel ini dengan metode average atau rata – rata (karena metode real terhenti sejak opening Bekasi, lupa dimaintenance) * penerimaan

POReceiveApprovalController.php	app/po_approval	ts_purchase_order_article_detail_statuses	Pengelolaan pada fitur Approval Penerimaan, fitur ini masih menggunakan database terkait PO, hanya saja pembedanya jika sudah approve, u_id_approve nya tidak akan kosong, digroup by poads_invoice. Qty pada penerimaan baru akan masuk ke BIN default jika sudah di approve melalui fitur ini * approval penerimaan
DashboardController.php	app/dashboard	Jika admin : akan berkolaborasi ke banyak database sesuai component yang ditampilkan, untuk mengetahui tabel terkait silahkan perhatikan Model yang diimport ataupun Query Builder DB::table	Jika admin : dashboard akan mengarah ke file tsb. Perhitungan hpp pada dashboard menggunakan metode avg yang diambil dari tabel ts_purchase_order_article_detail_statuses dan atau ts_purchase_order_article_details
Jika Helper : Controller yang aktif ada pada PointOfSalesController.php untuk packingListByInvoice dan orderListByInvoice Dan juga TrackingController.php untuk memasukkan atau mengeluarkan artikel.	app/dashboard/helper	Jika helper : database yang berinteraksi disini adalah ts_pos_transactions, ts_pos_transaction_details, ts_product_location_setup_transactions	Jika helper : dashboard akan mengarah ke subfolder helper, pos_transaction untuk menarik data Invoice, sedangkan pos_transaction_details untuk menandai artikel sudah diambil atau belum pada pos_td_qty_pickup, dan product_location_setup_transactions untuk helper mengelola artikel keluar masuk sesuai statusnya. * helper app * dashboard
InstockApprovalController.php	app/instock_approval	ts_instock_exception_approvals	Pengelolaan data user yang berhak melakukan approve instock dan juga setup memasukkan artikel ke bin exception.

InstockListController.php	app/instock_list	ts_instock_lists	<p>Sedangkan tabel satunya untuk mencatat history approval instock.</p> <p>* instock & exception approval * instock list</p>
MassAdjustmentController.php	app/mass_adjustment	ts_mass_adjustments ts_mass_adjustment_details	<p>Pengelolaan data mass adjustment, digunakan untuk melakukan adjustment menggunakan export dan import template. Qty baru akan berubah ketika sudah diklik Eksekusi.</p> <p>* mass adjustment</p>
ScanAdjustmentController.php	app/scan_adjustment	ts_scan_adjustments ts_scan_adjustment_details ts_scan_adjustment_sub_categories ts_scan_adjustment_brands	<p>Pengelolaan data adjustment menggunakan scanner, tabel brand dan sub category digunakan sebagai rules untuk menentukan kriteria artikel yang akan ditentukan.</p> <p>* scan adjustment</p>
AdjustmentController.php	app/adjustment	ts_bin_adjustments	<p>Pengelolaan data adjustment yang masih manual pick pada bin yang dipilih kemudian historynya disimpan pada tabel tersebut.</p> <p>* manual adjustment</p>
ExceptionLocationController.php	app/exception_location	ts_exception_locations	<p>Pengelolaan data exception location, jadi apapun BIN / Lokasi Simpan yang disetup pada menu ini, maka tidak akan dimunculkan pada data stok system manapun yang siap jual, termasuk pada API. Pengecekan pada tabel ini wajib dipanggil ke setiap apapun yang berhubungan dengan menampilkan data stok</p> <p>* exception location</p>

B1g1Controller.php	app/b1g1_location	ts_buy_one_get_ones	Pengelolaan data untuk data stok b1g1 yang bisa dipick langsung pada POS Offline, jadi pada POS ada opsi WAITING, STORE, WGN dan B1G1, maka artikel bisa langsung dipanggil pada opsi B1G1 tanpa perlu dipick pada fitur Data Stok. * b1g1 location (direct pick)
QtyExceptionController.php	app/qty_exception	ts_qty_exceptions	Pengelolaan data untuk menghidden artikel yang akan ditampilkan pada website, termasuk jumlah qty, misalkan artikel A qty 10, jika artikel A disetup pada fitur ini sebanyak 5, maka pada web hanya akan muncul 5. * web qty exception
ProductLocationSetupController.php	app/product_location_setup	ts_product_location_setups ts_product_stocks	Fitur ini dulunya digunakan sebagai media untuk melakukan setup stok artikel pada BIN, namun sekarang fungsinya hanya untuk melihat stok apa saja yang tersedia pada setiap BIN, ** PENTING ** tabel ts_product_location_setups merupakan kunci dari total stok yang tersedia pada setiap artikel pada bin, kolom PLS_QTY ini yang dibaca sebagai jumlah stok pada semua sistem apapun terkait topsystem. Sedangkan PS_QTY pada ts_product_stocks tidak lagi dipakai, QTY tsb tidak berpengaruh apa – apa. * stok pada bin
ProductLocationSetupV2	app/	ts_product_location_set	Jika pada file

Controller.php	product_location_setup_v2	ups ts_product_mutations	sebelumnya user tidak bisa setup, maka pada fitur ini bisa, tabel product_mutations untuk mencatat history user yang melakukan aktifitas mutasi dengan pls_id sebagai bin awal, dan pl_id sebagai tujuan lokasi setup. pmt_old_qty hanya untuk memberi tahu berapa qty bin awal sebelum mutasi, dan pmt_qty adalah qty yang dimutasi * setup lokasi stok
StockTransferController.php	app/stock_transfer	ts_stock_transfers ts_stock_transfer_details	Pengelolaan data artikel yang akan ditransfer dari store awal menuju store tujuan, pada stf_status 0 berarti sedang dikerjakan, 1 berarti in progress, 2 berarti done, 3 berarti draft. ** INFO ** Jika artikel sudah dipanggil pada menu ini, maka artikel sudah memotong stok pada tabel ts_product_location_setup * transfer stok
StockTransferDataController.php	app/stock_transfer_data	ts_stock_transfer_details_statuses	Pengelolaan data artikel yang sudah sampai atau diterima pada store tujuan, setiap penerimaan akan direkam pada tabel tersebut, baik partial maupun langsung full. stfd_qty adalah berapa yang dikirim sedangkan stfds_qty adalah berapa yang diterima. * penerimaan transfer
StockDataController.php	app/stock_data	Pada fitur ini tabel yang berperan sesuai yang ada dalam file controller bagian paling atas use app\Model	Menu ini digunakan untuk melihat data stok, sekaligus tim offline melakukan pick artikel untuk request kepada

		ataupun raw Query Builder pada DB::table, terutama ts_product_location_set ups terkait qty setiap artikel.	helper, sales online tidak bisa pick, sales offline selain posisi store bertugas juga tidak bisa pick. * data stok
StockTrackingController.php	app/stock_tracking	ts_product_location_set up_transactions beserta relasi – relasinya	Fitur ini digunakan untuk melacak history keluar masuk stok dari penjualan semua divisi, lebih track ke pergerakan stock bukan invoice. Setiap artikel berhasil terjual akan berstatus done, jika masuk lagi berstatus instock. Jika ada status complaint berarti sedang dalam proses refund / exchange yang nyangkut pada POS dan belum diselesaikan. Status refund/exchange dapat menjadi instock setelah helper memasukkan artikel ke gudang. * stock tracking
PointOfSaleController.php	app/offline_pos app/pos	ts_pos_transactions ts_pos_transaction_details ts_store_type_divisions ts_product_location_set up_transactions	Pada fitur ini akan membedakan app pos berdasarkan stt_name pada tabel ts_store_types, jika online akan diarahkan ke /pos dan offline ke /offline_pos. stt_id berelasi pada table ts_users. Untuk penjelasan mendetail pada fitur ini cek pada logical note paling bawah tabel. * pos app
NamesetDataController.php	app/nameset_data	ts_pos_transaction_details	Pengelolaan data artikel yang memiliki nameset order, pengambilan pada database berdasarkan kriteria kolom pos_td_nameset bernilai satu dan harga nameset tidak kosong, jika sudah

			<p>selesai maka tinggal setting pos_td_nameset menjadi 0 dan plst_status pada ts_product_location_setup_transactions menjadi DONE, maka akan hilang dari tabel.</p> <p>* nameset reminder</p>
TargetController.php	app/target	ts_targets ts_sub_targets ts_sub_sub_targets	<p>Pengelolaan data setup target berdasarkan bulan tahun, kemudian detail store mana, dan kemudian detail penjabaran perhari berapa, fitur target ini ditarik ke menu Sales Summary atau pada file PosSummaryController.php * catatan: fitur ini nampaknya perlu dibuatkan kolom untuk menampung hasil perhitungan agar tidak lemot dan berat.</p> <p>* setup target</p>
ProductDiscountController.php	app/product_discount	ts_product_discounts ts_product_discount_details	<p>Pengelolaan data setup diskon dimana tabel utama sebagai setup tipe potongan diskon sesuai store atau divisi, jika st_id null maka diskon akan berlaku untuk semua store namun mengikuti divisi terkait atau std_id.</p> <p>** Setiap pemanggilan artikel untuk ditampilkan, silahkan validasi harga jual artikel pada kedua tabel product_discounts ini, jika masuk discount biasanya tarik harga bandrol atau price_tag, jika tidak cukup sell_price (price_tag dan sell_price ini ada pada table ts_product_stocks dan ts_products).</p> <p>* setup diskon</p>

VoucherController.php	app/voucher	ts_vouchers ts_voucher_transactions	<p>Pengelolaan data setup voucher yang digunakan pada POS, sebenarnya juga pada website, sudah berjalan pada website, namun khusus website perlu dicek atau testing kembali, karena sudah lama tidak dicek penggunaan pada website, karena ada update beberapa rules pada voucher seperti penambahan tipe gift dan repeat used. Tipe Gift dan Cashback belum berjalan.</p> <p>Khusus tabel ts_voucher_transactions, untuk merekam invoice yang menggunakan serta memvalidasi penggunaan berulang, namun data pada tabel ini belum ditarik pada laporan penjualan mengingat terlalu crowded development. Tabel ini sering digunakan pada controller PointOfSaleController.php * setup voucher</p>
PosSummaryController.php	app/pos_summary	Database yang terlibat ada pada use app\ Models paling atas, lebih ke tabel – tabel terkait penjualan	<p>Menu ini untuk melihat summary penjualan seluruh store yang sudah dipotong biaya admin, dapat difilter berdasarkan tanggal dan store, selain Depok dan Bandung, maka ada Cross Order Chart ditengah antara online dan offline. * sales summary</p>
CrossOrderController.php	app/cross_order	ts_pos_transactions ts_pos_transaction_details	Pengelolaan data orderan cross atau lintas cabang, cukup panggil data pada database

			<p>disamping dengan nilai cross_order 1 pada tabel ts_pos_transactions, kemudian ketika dikonfirmasi oleh penerima orderan, maka potong stok sesuai bin yang telah dipilih, kemudian rekam u_id_cross pada ts_pos_transactions sebagai user yang melakukan action konfirmasi, jika cancel, cukup hilangkan semua centang ketika user akan konfirmasi invoice, maka invoice otomatis cancel.</p> <p>* cross order</p>
InvoiceTrackingController.php	app/invoice_tracking	ts_pos_transactions ts_pos_shipping_information	<p>Pengelolaan data invoice transaksi dari seluruh divisi, juga untuk update data nomor resi pengiriman, yang kemudian setiap update resi akan mengirim notifikasi ke wa customer, ts_shipping_information untuk menangkap status terakhir nomor resi dari invoice tersebut melalui API binderbyte.</p> <p>* invoice tracking</p>
ResellerActivityController.php	app/reseller_activity	ts_reseller_activities ts_customers	<p>Untuk melihat aktifitas reseller pada reseller.topscore.id, user menggunakan tabel customer sebagai data login. Cukup rekam dan input setiap kegiatan user beserta id user pada tabel activities.</p> <p>* aktifitas reseller</p>
WebinarController.php	app/webinar	ts_webinar_registrations	<p>Untuk menampung dan menampilkan hasil pendaftaran webinar dari website https://www.topscore.id/webinar</p>

			* pendaftaran webinar
ResellerController.php	app/reseller	ts_customers ts_reseller_deposits ts_reseller_orders ts_reseller_level_accesses ts_reseller_levels ts_reseller_add_discount_accesses ts_reseller_add_discounts	<p>Pengelolaan data reseller beserta setup reseller dengan level dan additional discount. Untuk level akan tersetup otomatis sesuai jumlah belanja yang telah dilakukan, namun jika ada case error saat reseller login atau menggunakan sistem, bisa jadi penyebabnya belum memiliki level (untuk case error ini sudah di solve dan tidak pernah terjadi lagi).</p> <p>Username dan nomor hp pada reseller harus unique. Terdapat kolom sudah pernah belanja ? Jika diisi jumlah tersebut akan terakumulasi dengan jumlah yang pernah reseller transaksi pada sistem.</p> <p>* data reseller</p>
ResellerLevelController.php	app/reseller_level	ts_reseller_levels ts_reseller_level_accesses	<p>Pengelolaan data level untuk reseller, dan relasi antara level dengan reseller, yang paling penting adalah rl_amount sebagai jumlah potongan diskon reseller, rl_min_transaction sebagai jumlah minimal reseller harus belanja untuk mencapai level tersebut (akumulasi history total belanja), serta rl_min_item_value sebagai nilai minimal harga suatu item, misalkan diisi 500000 maka harga sepatu minimal harus 500rb baru discount akan terapply, discount akan memotong masing – masing harga item.</p>

			* reseller level
ResellerBrandLevelController.php	app/ reseller_brand_level	ts_reseller_brand_levels	Pengelolaan detail discount hingga ke brand, jika tidak ada discount yang disetup pada menu ini, maka potongan harga artikel akan mengikuti induk discount pada tabel ts_reseller_levels, namun misal brand A kita setup 10 sedangkan pada induk 15, maka artikel akan mengikuti detail pada tabel ini, yaitu sebesar 10, tabel induk diabaikan. * reseller brand level
ResellerAddDiscountController.php	app/ reseller_add_discount	ts_reseller_add_discounts ts_reseller_add_discount_accesses	Pengelolaan data additional discount pada reseller, perbedaan antara diskon pada level dengan menu ini adalah jika level memotong discount artikel, sedangkan additional memotong akumulasi total keseluruhan harga akhir. Hanya reseller tertentu yang mendapatkan hak additional disc. * reseller additional disc
ResellerConfirmationController.php	app/ reseller_confirmation	ts_reseller_confirmations ts_customers	Pengelolaan data konfirmasi bayar pendaftaran reseller. * reseller konfirmasi
ResellerTransactionController.php	app/ reseller_transaction	ts_reseller_transactions ts_reseller_transaction_details	Pengelolaan data transaksi reseller yang telah checkout. Pada tabel ini ada 2 logic pemotongan stok, jika reseller 'tidak' menggunakan saldo maka invoice tercreate baru dari sisi reseller, belum jadi invoice internal dan stok belum memotong sebelum reseller telah bayar dan

			<p>dikonfirmasi. Namun jika menggunakan saldo maka stok akan langsung terpotong dan terekam pada stock tracking, namun tetap harus dikonfirmasi untuk menjadikannya invoice internal. Admin yang bertugas masih bisa menghapus atau mengurangi stok yang dibeli reseller ketika detail invoice diklik dan klik button minus pada kolom BIN. Penjelasan detail ada di Logical Note</p> <p>* reseller transaksi</p>
ResellerDepositController.php	app/reseller_deposit	ts_reseller_deposits ts_reseller_deposit_details	<p>Pengelolaan saldo deposit pada reseller, tabel utama untuk mencatat siapa user dan saldo yang aktif, kemudian tabel detail digunakan untuk mencatat history transaksi, admin dapat menambah atau mengurangi deposit, kemudian setiap reseller checkout menggunakan saldo otomatis akan tercatat disini dengan user default mas afa atau dengan id 37.</p> <p>* reseller deposit</p>
ClientCredentialController.php	app/client_credential	ts_client_credentials	<p>Pengelolaan data client_id dan client_api_key untuk user yang akan bekerja sama menjual stok wholesale topscore sebagai validasi ketika hit endpoint API. Menggunakan bantuan library Str::random.</p> <p>* reseller api access</p>
VotingController.php	app/voting	ts_vote_categories ts_vote_category_details	<p>Pengelolaan data voting pada voting system vote.topscore.id, tabel</p>

		ts_vote_category_detail_results ts_vote_ip_addresses	utama untuk menentukan kategori apapun yang akan divote, category_details untuk menjabarkan masing – masing item pada kategori utama, dimana vcd_id_reference dapat diambil dari berbagai id terkait, kemudian category_detail_results digunakan untuk menangkap ip user yang melakukan voting, dimana tabel ip user menangkap id customer sebagai user yang melakukan vote. via_blocked digunakan untuk block IP yang melakukan spam. * kelola voting
UserRatingController.php	app/user_rating app/ rating_by_customer	ts_user_ratings	Perhatikan routes pada routes/web.php, jika segment url mengarah ke user_rating maka digunakan untuk melihat hasil rating dari customer kepada kasir, sedangkan rating_by_customer akan diarahkan ke aplikasi pada tab untuk customer mengisi rating kepada kasir. App pada tab baru akan aktif jika kasir memilih customer rating ya pada POS, jika tidak tampilan akan blank. Disini saya masih menggunakan setInterval untuk membaca data yang masuk (saran, sebaiknya kedepan diganti menggunakan pusher agar lebih hemat memory). * rating customer app * user rating

CustomerController.php	app/customer	ts_customers ts_customer_types	Pengelolaan data customer beserta tipe customer, tabel ini juga digunakan sebagai login user pada website topscore.id dan reseller.topscore.id, namun bedanya yang bisa login reseller hanya ct_id 2 dan 4 atau reseller dan dropshipper. Biasanya ada yang st_id dan stt_id kosong atau tidak kosong, yang tidak kosong masuk melalui POS Online ataupun User Rating, juga register Website. * data customer
DebtListController.php	app/debt_list	ts_debt_lists ts_debt_list_payments	Pengelolaan data daftar hutang, tabel utama digunakan untuk membuat data hutang, kemudian list_payments untuk mencatat pembayaran dari hutang tersebut. Data pada tabel ini juga yang ditarik pada dashboard v1 maupun v2. * daftar hutang
SalesReportController.php controller terkait: InvoiceReportController.php ArticleReportController.php	app/report/ sales_report	Tabel yang terlibat sesuai dengan yang ada pada controller, lebih ke tabel – tabel seputar transaksi.	Pengelolaan data laporan penjualan dimana terdapat 4 tabel pada fitur ini, untuk cek Harga Beli Harga Jual, kemudian tabel By Invoice (ts_pos_transactions), By Article (ts_pos_transaction_details), serta tabel untuk melihat detail Cross Order ketika diklik pada Summary Penjualan (tombol cross order ini hanya muncul jika store selain Depok, Bandung, Urban dan Wholesale). * laporan penjualan
StockCardController.php	app/stock_card	ts_stock_exports	Menu ini digunakan

			<p>untuk mengelola laporan stok yang terjadi pada system, mulai dari beginning stok hingga ke ending, dimana setiap komponen diambil dari tabelnya masing – masing, seperti purchase dari ts_purchase_order_article_detail_statuses, adjustment dari ts_bin_adjustments, scan_adjustments, dll. Perlu optimasi pada fitur ini, karena query salah, terlalu banyak query yang dilooping dan tidak efektif, sehingga lemot.</p> <p>* laporan stok</p>
DashboardV2Controller.php	app/dashboard_v2	ts_dashboard_informations ts_brand_information	<p>Menu ini digunakan untuk melihat history ringkasan dashboard yang diclose setiap hari pukul 23.57 pada cron jobs. Awalnya sebagai tampilan baru dashboard, namun tidak jadi dan diputuskan untuk jadi laporan summary saja.</p> <p>* laporan summary</p>
PoReceiveReportController.php	app/report/po_receive	Tabel yang terlibat lebih ke purchase_orders dan turunannya, sesuai relasi pada controller.	<p>Pengelolaan data untuk melihat informasi datang barang berdasarkan PO, berapa qty yang di PO dan berapa yang sudah diterima, data dapat dilihat dan diexport sesuai kriteria filter yang ditentukan, sudah full, belum full dan range tanggal.</p> <p>* laporan datang barang</p>
UserController.php	app/user	ts_users ts_groups ts_user_groups ts_stores ts_store_types	<p>Pengelolaan data user yang menggunakan topsystem beserta setup level tiap user. Tertinggi adalah level administrator yang dapat mengakses semua menu,</p>

			<p>selain itu hanya bisa mengakses menu tertentu sesuai pada file yang mengandung nama sidebar pada views/app/_partials. Namun kekurangannya hanya sekedar menghidden user dari akses ke url menu, tetapi jika user paham dan langsung akses url maka tetap akan dapat masuk meski ada rules yang sudah dipisahkan sesuai level. (semoga dapat diperbaiki kedepannya, karena kemarin tidak dibenahi dikejar deadline fitur lainnya).</p> <p>* data user</p>
InvestorController.php	app/investor	ts_investors	<p>Pengelolaan data login user pada urban.topscore.id, user akan mengakses summary dashboard sesuai store yang ditentukan pada data masing – masing.</p> <p>* data investor</p>
MarketplaceManagerController.php	app/marketplace_manager	ts_marketplace_managers	<p>Pengelolaan data untuk synchronize kode pada template marketplace dengan artikel pada internal sistem, disini user harus menyesuaikan urutan posisi kode pada template agar import membaca posisi tersebut. Namun ada kekurangan, user masih harus input kode satu per satu, tidak cukup waktu untuk pengembangan.</p> <p>* marketplace stock</p>
ShopeeController.php	app/shopee_data	ts_shopee_data ts_template_groups ts_product_stocks	<p>Menu ini dihentikan dalam development, awalnya digunakan untuk import template marketplace shopee dan auto synchroniz antara</p>

			penamaan artikel pada template dengan internal system, sehingga dapat menghasilkan data stok terbaru. Sudah berjalan namun belum terlalu akurat, mungkin kedepannya ingin dijadikan referensi dalam pengembangan. * shopee (trial)
WebTransactionController.php	app/web_transaction	ts_pos_transactions ts_pos_transaction_details	Pengelolaan data transaksi pada website topscore.id, masih seputar tabel pos_transactions, namun dikhususkan dengan kolom is_website bernilai 1. ketika user checkout pada website sudah memotong stok, namun diberi batas waktu maksimal 6 jam untuk menyelesaikan pembayaran, ada cronjob untuk melakukan check terhadap pembayaran midtrans, serta mengirim chat whatsapp reminder pembayaran kepada customer. * transaksi
WebConfirmationController.php	app/web_confirmation	ts_confirmations	Pengelolaan data konfirmasi pembayaran pada transaksi website, ada 2 metode pembayaran pada website, yaitu midtrans dan manual, jika manual maka bukti transfer dari customer akan ditampung pada tabel tersebut, dan dikelola pada controller ini. Jika sukses maka status akan PAID, jika Cancel maka stok akan dikembalikan lagi. * konfirmasi
WebArticleController.php	app/web_article	ts_products beserta relasi terkait.	Pengelolaan data artikel yang akan ditampilkan

			<p>pada website, menu ini hanya untuk melengkapi data seperti gambar, slug, berat artikel (wajib untuk menghitung ongkir).</p> <p>* artikel</p>
TopdealControllerr.php	app/topdeal	ts_topdeals ts_top_deal_details	<p>Pengelolaan data banner Topdeals pada website topscore.id beserta detail artikel yang akan ditampilkan, topdeals tidak akan ditampilkan jika tanggal expired atau kurang dari hari ini ataupun juga jika status nonaktif.</p> <p>* setup topdeals</p>
WebBanner	app/web_banner	ts_banners ts_banner_brands ts_banner_brand_details	<p>Pengelolaan data banner pada website topscore.id , dimana tabel utama untuk menampung gambar banner, slug, urutan banner serta posisi banner, jika is child 1 maka banner akan berada dibawah banner best seller dan terbaru. Kemudian banner_brands untuk detail brand yang akan muncul jika banner tersebut klik, kemudian banner_brand_details untuk setup detail sub sub kategori yang akan ditampilkan (jika tidak disetup, maka default akan menampilkan artikel apapun asal 1 brand tanpa melihat sub sub kategori).</p> <p>* setup banner</p>
BankController.php	app/bank	ts_banks	<p>Digunakan untuk mengelola nomor rekening bank transfer ketika customer berhasil checkout, akan diarahkan sesuai bank atau metode yang dipilih customer.</p>

			Kekurangannya jika menambah bank baru maka tidak otomatis tampil, masih harus ditambah secara manual script pada project topscore.id (dahulu terburu – buru ingin launching web, sehingga terlewat bagian ini, sebenarnya tinggal dipanggil dan dilooping) * akun bank
WebBrandController.php	app/brand	ts_brands	Sebenarnya sama saja ketika akan mengelola data brand pada master data, hanya saja lebih difokuskan untuk mengupdate gambar, banner dan slug. Gambar utama akan ditampilkan pada menu Brand dan Sliding brand di landing page topscore.id. * slug brand
WebCategoryController.php	app/web_category	ts_category_slugs	Digunakan untuk mengelola 6 gambar icon dibawah sliding brand pada landing page, beserta detail banner dan slug ketika diklik. Slug dapat diarahkan ke mana saja, baik kategori slug, brand slug, promo, dll. Sesuai logika slug yang mampu dibaca oleh system. * slug kategori
WebSubCategoryController.php	app/web_sub_category	ts_product_sub_categories	Pengelolaan data slug pada sub_categories, jika url website topscore.id memiliki segment yang mengandung slug terkait, maka artikel yang ditampilkan akan mengacu pada id sub categories, atau pada ts_products dengan psc_id sesuai slug. * slug sub kategori

FreeShippingController.php	app/free_shipping	ts_free_shipping_costs	Pengelolaan data daerah yang akan mendapatkan gratis ongkir jika ID daerah tersebut terdeteksi pada tabel ini, daerah diambil dari API raja ongkir. Fitur ini ketika customer akan melakukan checkout pada topscore.id. * setup free ongkir
WebReminderController.php	app/web_reminder	ts_user_reminders	Pengelolaan data user yang mengirimkan request pesan pengingat jika produk kosong, ada pada detail produk website topscore.id dibawah size artikel. * cust request/reminder
MetaTagController.php	app/meta_tag	ts_meta_tags ts_meta_tag_details	Pengelolaan data meta tags yang ditampilkan pada bagian footer website topscore.id, dapat disisipkan sub kategori dan brand, untuk keperluan SEO. * meta tags
BlogController.php	app/blog_controller	ts_blog_categories ts_blog_contents	Pengelolaan data content blog pada website topscore.id, kekurangannya insert images belum dapat direct upload tanpa menyisipkan url. * blog
WhatsappController.php	app/whatsapp	ts_whatsapps	Digunakan untuk mengirimkan blast whatsapp ke nomor customer, tabel tersebut hanya untuk menyimpan history pengiriman pesan beserta status, kekurangan masih suka terputus, tidak semua terblast. * wa blast
UpdatedDashboardController.php	app/updated_dashboard	Tabel ada pada semua Model / Query Builder didalam controller ini.	Digunakan untuk mengelola menu Dashboard V2, semua komponen pada halaman

			ini diambil dan dihitung berdasarkan tabel masing – masing. Baca logical note untuk gambaran script. * dashboard v2
UserActivityController.php	app/dashboard app/ updated_dashboard	ts_user_activities	Digunakan untuk melihat aktifitas user pada sistem, setiap controller akan memiliki function yang merekam aktifitas mereka, namun belum merata kesemua fitur, jika ada controller yang belum memiliki function UserActivity berarti belum ditrack aktifitas pada fitur tersebut. * aktifitas user
AssetDetailController.php	app/asset_detail	Tabel terkait ada pada controller, sesuai komponen yang dibutuhkan pada fitur.	Fitur ini ada pada dashboard v2 tombol Lihat Detail Asset/Penjualan, hampir mirip dengan Laporan Stok namun memiliki value dalam nilai rupiah, dan juga yang diambil hanya transaksi stok dari purchase, transfer dan penjualan, tidak sekompleks Laporan Stok. * lihat detail asset/penjualan
StoreAgingController.php	app/store_aging	ts_store_agings ts_store_aging_details	Digunakan untuk membatasi online cross order melakukan pickup barang ke store yang dituju, juga sebagai setup untuk menarik artikel dengan aging tertentu ke wholesale (development belum selesai karena waktu tidak cukup untuk penarikan ke wholesale). * setup store aging
InvoiceEditorController.php	app/invoice_editor	ts_invoice_editor_permissions	Fitur ini digunakan untuk Head Online dan

		ts_invoice_editors	Head Offline melakukan edit pada invoice, jika pada invoice mengalami kendala salah divisi, atau salah harga, atau mau membatalkan item, user yang melakukan edit akan direkam pada history ts_invoice_editors. * invoice editor
--	--	--------------------	--

2) Topscore.id

Merupakan website Ecommerce topscore.id, proyek ini memiliki 2 tampilan berbeda antara mobile dan website, sehingga proyek memiliki 2 struktur folder views untuk web dan mobile, stok yang ditampilkan pada website hanya dari Store Depok dan Urban atau dengan id 2 dan 4.

app/Http/Controllers	resources/views/ web/ mobile/	database	keterangan
AuthController.php	_partials/_header _partials/_js	ts_customers	Untuk login user, popup login dan js ada pada lokasi file terlampir.
BlogController.php	_blog	ts_blog_categories ts_blog_contents	Untuk menampilkan data blog
CartController.php	_cart	ts_carts	Untuk mengelola data keranjang user, pada database jika user klik beli langsung otomatis is_direct bernilai 1, jika hanya menambahkan ke keranjang akan bernilai 0. 1 dan 0 ini untuk menarik item ketika user akan checkout.
ConfirmationController.php	_confirmation	ts_confirmations ts_pos_transactions	Untuk mengelola menu konfirmasi pada website, dengan relasi ke tabel pos_transactions untuk membaca bahwa invoice yang akan dikonfirmasi benar – benar valid.
DashboardController.php	_dashboard	Tabel sesuai apa yang ada pada controller.	Landing page utama ketika pertama kali

			website topscore.id diakses, item dilooping berdasarkan database terkait konten.
ProductController.php	_product	Tabel terkait ada pada controller	Digunakan untuk pengelolaan data artikel pada product listing, ketika landing page diklik Lihat Semua, ataupun pencarian search ataupun brand diklik, akan masuk ke controller ini. memiliki multiple filter sesuai kriteria seperti sub kategori, size, brand, dan harga.
ReminderController.php	_product_detail	ts_user_reminders	Digunakan untuk menyimpan data ketika user klik Tidak menemukan ukuran yang Anda cari? Pada detail artikel.
SearchController.php	_product		Digunakan untuk mengatur pencarian dan result dari pencarian.
SitemapController.php	../sitemap.blade.php	Tabel diambil sesuai pada controller yang memiliki slug untuk diakses	Digunakan untuk membuat sitemap, agar bisa ditracking pada Google Search Console dan SEO.
ShippingController.php	_cashier		Digunakan untuk menarik data ongkos kirim dari API RajaOngkir.
SlugController.php			Merupakan controller paling penting untuk mengarahkan setiap halaman slug yang diakses oleh user, biasanya untuk membaca segment setelah / pada topscore.id , contoh topscore.id/sepatu-futsal-blabla , sepatu-futsal-blabla ini akan dicontrol oleh

			SlugController agar diarahkan ke page dan konten yang sesuai.
TransactionController.php	_cashier dan _order		Digunakan untuk merekam hasil checkout customer serta memotong stok sesuai BIN, dan menyimpan data ke snap midtrans (jika method midtrans).
UserController.php		ts_customers	Digunakan untuk menyimpan hasil edit data dari user
VoucherController.php		ts_vouchers	Digunakan untuk validasi voucher yang diinput user pada saat di keranjang atau kasir website, namun perlu diperhatikan lagi validasinya.
WebinarController.php	_webinar	ts_webinar_registrations	Digunakan untuk merekam data user yang melakukan pendaftaran pada url topscore.id/webinar

3) Vote.Topscore.id

Merupakan voting sistem untuk pengambilan suara terhadap opsi yang telah diseting dan tentukan pada menu Kelola Voting pada Topsystem.

app/Http/Controllers	resources/views/	database	keterangan
DoVotingController.php	Voting	ts_vote_categories ts_vote_category_details ts_vote_category_details_results ts_vote_ip_addresses	Penjelasannya sama seperti pada menu voting topsystem, hanya saja project pada website ini lebih untuk mengelola hasil tindakan user yang melakukan vote, untuk menampilkan dan menyimpan data terkait user.

4) Reseller.Topscore.id

Merupakan sistem yang digunakan oleh para reseller untuk melihat dan membeli data stok Topscore Wholesale, dimana pengelolaan data user serta transaksi dapat dilihat pada menu Reseller di Topsystem

app/Http/Controllers	resources/views/	database	keterangan
AuthController.php		ts_customers	Untuk mengelola login dan register customer serta auto setting basic level dari reseller, tipe customer (ct_id) harus '4' alias reseller yang bisa login.
CartController.php	app/_partials/header	ts_reseller_carts beserta tabel yang ada didalam controller	Untuk mengatur keranjang belanja reseller, serta penarikan setup diskon pada setiap level terhadap artikel pada keranjang. Serta ketika user checkout keranjang tersebut untuk diinput kedalam invoice reseller.
CheckExistController.php			Untuk validasi data customers terkait username, telepon dan email yang sudah ada.
ConfirmationController.php	app/dashboard	ts_reseller_confirmations	Digunakan untuk merekam konfirmasi uang pembayaran reseller ketika berhasil register pada sistem.
DashboardController.php	app/dashboard	Semua tabel yang ada pada controller ini, terutama terkait data artikel dan stok. ts_products ts_product_location_sets ts_exception_locations dll.	Digunakan untuk menampilkan data stok pada halaman utama ketika user berhasil login. Data stok dipanggil secara frontend side, tidak server side agar mudah custom artikel yang dipanggil berdasarkan aging collaboration.
LandingController.php	app/landing		Untuk menampilkan halaman landing page ketika web pertama kali diakses reseller.topscore.id
TransactionController.php	app/transaction	ts_reseller_transactions ts_reseller_transaction_details ts_reseller_trx_confirmations	Untuk menampilkan data transaksi yang telah dilakukan oleh reseller, beserta menampilkan detail

			dan konfirmasi bukti pembayaran dari checkout artikel.
--	--	--	--

5) Urban.Topscore.id

Merupakan sistem yang digunakan oleh user untuk melihat data aset serta penjualan pada Tops Urban Depok, pengelolaan data user yang dapat menggunakan sistem ini ada pada menu Data Investor Topssystem, namun sebenarnya tidak hanya untuk melihat store Tops Urban, tetapi bisa untuk store lainnya sesuai apa yang disetting.

app/Http/Controllers	resources/views/	database	keterangan
AuthController.php	auth/	ts_investors	Login user untuk urban.topscore.id
DashboardController.php	app/dashboard	Tabel sesuai dengan komponen yang ditampilkan terkait penjualan dan aset	Untuk menampilkan konten saat user berhasil login.
TableController.php		Tabel sesuai dengan apa yang diklik oleh user	Untuk menampilkan grafik dan tabel sesuai dengan apa yang diklik oleh user.

6) Api.Topscore.id

Merupakan API yang digunakan untuk pihak external synchronize data stok Topscore Wholesale dan Urban.

app/Http/Controllers/API	Database	keterangan
StockController.php	ts_products ts_product_location_setups ts_client_credentials serta tabel lain sesuai pada controller	Digunakan untuk menampilkan request dari user terhadap data stok, sebelum diarahkan pada database, user harus validasi dahulu api key dan client id yang mereka sisipkan pada header.
TransactionController.php	ts_client_credential_transactions ts_client_credential_transaction_details	Controller ini awalnya digunakan untuk user melakukan checkout stock, namun tidak terpakai hingga saat ini, bisa diabaikan saja atau dikembangkan.

Logical Note

Perihal	Keterangan
Data Stok	Perlu diperhatikan, algorithm yang berjalan saat ini untuk qty stock diambil dari PLS_QTY pada tabel ts_product_location_setups, namun jangan lupa dimanapun ada pemanggilan stok, selalu lampirkan pengecekan bin yang

didaftarkan pada `ts_exception_locations`, agar tidak ditampilkan pada data stok.

Untuk pembacaan aging diambil dari 2 tabel, yang diutamakan `ts_purchase_order_transaction_detail_statuses` dan yang kedua adalah transfer dari `ts_stock_transfer_detail_statuses`, keduanya dari field `created_at`.

Penjelasan script jquery datatables frontend :

```
var stock_data_table = $('#StockDatatb').DataTable({
  destroy: true,
  processing: false,
  serverSide: true,
  responsive: false,
  dom: 'rt<"text-right"i>',
  buttons: [
    { "extend": 'excelHtml5', "text": 'Excel', "className": 'btn btn-primary btn-xs' }
  ],
  ajax: {
    url: "{ url('stock_data_datatables') }",
    data: function (d) {
      d.search = $('#stock_data_search').val();
      d.br_id = $('#br_id').val();
      d.pc_id = $('#pc_id').val();
      d.psc_id = $('#psc_id').val();
      d.pssc_id = $('#pssc_id').val();
      d.sz_id = $('#sz_id').val();
      d.display_status = $('#display_status').val();
      d.st_id = $('#st_id_filter').val();
    }
  },
  columns: [
    { data: 'article_name', name: 'article_name', orderable: false },
    { data: 'article_stock', name: 'article_stock', orderable: false },
  ],
  columnDefs: [
    {
      "targets": 0,
      "className": "text-left",
      "width": "0%"
    }
  ],
  rowCallback: function( row, data, index) {
    if (data.article_stock.indexOf("<table></table>") >= 0) {
      $(row).hide();
    }
  },
  lengthMenu: [[10, 25, 50, 100, -1], [10, 25, 50, 100, "Semua"]],
  language: {
    "lengthMenu": "_MENU_",
  },
  order: [[0, 'desc']],
});
```

Setiap inisialisasi datatables terhadap id tabel disimpan kedalam sebuah variabel, agar variabel tersebut dapat ditrigger agar berubah/redraw ketika ada suatu event seperti filter atau pencarian. Untuk properties datatables seperti `serverSide`, `responsive` bisa baca didokumentasi resmi jquery datatables.

* `dom` : berfungsi untuk memanggil fitur bawaan plugin, seperti button export, filter, paging, pencarian. Kemudian `buttons`: untuk customize tampilan button.

* `ajax` : terdapat 2 subproperty yaitu `url` dan `ajax`, `url` adalah arah routes kemana plugin akan berkomunikasi dengan backend php, dan [data: function](#) dapat berisi variabel apapun yang ingin kita tangkap ke backend sebagai filter, dll.

* `columns` : berisi nama field yang telah kita setup pada backend, `data`: dapat

berisi nama field sesuai database ataupun hasil custom dari → editColumn, kemudian name: harus berisi sesuai dengan yang ada pada database.

* columnDefs : berisi custom penampilan baris data sesuai row yang kita inginkan.

* lengthMenu : untuk membuat custom berapa data yang ingin kita tampilkan pada fitur length datatables.

* order : untuk mengurutkan data, namun order ini tidak akan berfungsi jika pada backend kita sudah menentukan → orderBy

penjelasan script backend, bisa dilihat pada controller StockDataController baris ke 54 getDatatables :

```
public function getDatatables(Request $request)
{
    $exception = ExceptionLocation::select('pl_code')
    ->leftJoin('product_locations', 'product_locations.id', '=', 'exception_locations.pl_id')->get()->toArray();

    $big1_setup = BuyOneGetOne::select('pl_code')
    ->leftJoin('product_locations', 'product_locations.id', '=', 'buy_one_get_ones.pl_id')->get()->toArray();
    if (empty($request->st_id)) {
        $st_id = $request->st_id;
    } else {
        $st_id = Auth::user()->st_id;
    }
}
```

Dapat dilihat pada bagian awal function saya memanggil bin yang sudah disetup pada ExceptionLocation agar bisa saya batasi untuk tidak dapat dipanggil. Kemudian pada baris berikutnya saya panggil bin yang ada pada buyonegetone agar bisa saya beri tanda warna khusus pada tampilan data stok.

```
if(request()->ajax()) {
    return datatables()->of(Product::selectRaw("ts_products.id as pid, CONCAT(p_name,' ('.br_name,')') as p_name_brand, p_name, br_name, ps_qty, p_price."
    ->leftJoin('brands', 'brands.id', '=', 'products.br_id')
    ->leftJoin('product_stocks', 'product_stocks.p_id', '=', 'products.id')
    ->leftJoin('sizes', 'sizes.id', '=', 'product_stocks.sz_id')
    ->leftJoin('product_location_setups', 'product_location_setups.pst_id', '=', 'product_stocks.id')
    ->leftJoin('product_locations', 'product_locations.id', '=', 'product_location_setups.pl_id')
    ->where('pls_qty', '>', 0)
    ->whereNotIn('pl_code', $exception)
    ->where('product_locations.st_id', '=', $st_id)
    ->groupBy('p_name_brand')
    ->orderByDesc('products.updated_at'))
    ->editColumn('article_name', function($data){
        $price_tag = 0;
        $sell_price = 0;
        if (empty($data->p_price_tag)) {
            $price_tag = $data->p_price_tag;
        } else {
            $price_tag = $data->ps_price_tag;
        }
        if (empty($data->p_sell_price)) {
            $sell_price = $data->p_sell_price;
        } else {
            $sell_price = $data->ps_sell_price;
        }
        $hb = '<span style="white-space: nowrap; font-weight:bold; background: rgb(0,0,0);
        background: linear-gradient(90deg, rgba(0,0,0,1) 0%, rgba(116,0,0,1) 35%);" class="badge badge-sm badge-primary">B: '.number_format($price_tag).'
        return '<span style="white-space: nowrap; font-weight:bold;" class="btn btn-sm btn-primary">['.$data->br_name.'] '.$data->p_name.' '.$hb.'</span>';
    });
}
```

Kemudian baris berikutnya saya cek apakah ada request dari ajax, jika ada maka akan return beberapa pemanggilan pada database dengan beberapa custom untuk dilempar ke frontend.

Dapat dilihat saya langsung memanggil Product:: yang berarti Eloquent dari Model, sebenarnya bisa juga via manual Query Builder dengan bentuk [DB::table\('ts_products'\)](#), kemudian disitu saya menggunakan selectRaw() tidak select(), karena ada beberapa yang ingin saya custom pada query yaitu penggunaan CONCAT() agar bisa memanggil nama+brand digabungkan sekaligus. Jika menggunakan RAW biasanya field akan diapit oleh selectRaw("field1, field2, dst") , jika select biasa dapat dipisah select('field1', 'field2', 'field3').

Dibaris tersebut terdapat whereNotIn yang berarti saya tidak ingin stok dari lokasi exception ditampilkan.


```

->editColumn('article_name', function($data){
    $price_tag = 0;
    $sell_price = 0;
    if (!empty($data->p_price_tag)) {
        $price_tag = $data->p_price_tag;
    } else {
        $price_tag = $data->ps_price_tag;
    }
    if (!empty($data->p_sell_price)) {
        $sell_price = $data->p_sell_price;
    } else {
        $sell_price = $data->ps_sell_price;
    }
    $hb = '<span style="white-space: nowrap; font-weight:bold; background: rgb(0,0,0);
background: linear-gradient(90deg, rgba(0,0,0,1) 0%, rgba(116,0,0,1) 35%);" class="badge badge-sm badge-primary">B: '.number_format($price
return '<span style="white-space: nowrap; font-weight:bold;" class="btn btn-sm btn-primary">['.$data->br_name.'] '.$data->p_name.' '.$hb.'<
});
->editColumn('article_age', function($data){
});
->editColumn('article_stock', function($data) use ($request, $exception, $tblg_setup, $st_id) {
    $sz_id = $request->get('sz_id');
    $item = Product::selectRaw("ts_products.id as pid, pst.id, CONCAT(p_name, ' ('.br_name.')') as p_name_brand, p_name, p_color, br_name, ps_qt
->leftJoin('brands', 'brands.id', '=', 'products.br_id')
->leftJoin('product_stocks', 'product_stocks.p_id', '=', 'products.id')
->leftJoin('product_location_setups', 'product_location_setups.pst_id', '=', 'product_stocks.id')
->leftJoin('product_locations', 'product_locations.id', '=', 'product_location_setups.pl_id')
->where('pls_qty', '>', 0)
->where('product_locations.st_id', '=', $st_id)
->whereNotIn('pl_code', $exception)
->where('p_name', $data->p_name)
->where('br_name', $data->br_name)
->where(function($w) use ($sz_id) {
    if (!empty($sz_id)) {
        if (count($sz_id) > 0) {

```

Kemudian setelah penutupan query, saya ingin mengcustom beberapa field, sehingga saya gunakan → editColumn, didalam chain function ini kita dapat menambahkan algoritma atau function apapun untuk direturn, jangan lupa jika kita ingin memanggil external variabel gunakan use (\$var1, \$var2) agar bisa dipakai dalam editColumn. Nama hasil edit column ini bisa dipanggil pada frontend pada → columns: { data: disini seperti penjelasan pada script frontend.

Jika dirasa editColumn memiliki element HTML dan settingan CSS, dapat kita daftarkan column tersebut pada → rawColumns agar tampil sesuai, jika tidak dia hanya akan menjadi text script.

```

->rawColumns(['article_name', 'article_age', 'article_stock'])

```

Kemudian setelah rawColumns terdapat → filter, disini kita dapat mengelola variabel yang kita tangkap dari frontend, sebenarnya tidak mesti menangkap disini, bisa juga pada query → where(function()) dan juga pada editColumn tergantung contoh kasusnya, namun disini sebagai contoh ada pada filter

```

->filter(function ($instance) use ($request) {
    if (!empty($request->get('br_id'))) {
        $instance->where(function($w) use($request){
            $br_id = $request->get('br_id');
            $count = (Integer)count($br_id);
            $where = array();
            if ($count > 0) {
                for ($i = 0; $i < $count; $i++) {
                    $where[] = $br_id[$i];
                }
                $w->orWhereIn('br_id', $where);
            } else {
                $w->orWhere('br_id', '=', $br_id[0]);
            }
        });
    }
});

```

Jangan lupa use (\$request) agar variabel dari request http bisa kita tangkap, disini pada frontend saya memiliki filter br_id atau brand, agar data stok bisa difilter multiple brand. Karena biasanya default get method dari datatables akan mengubah array menjadi data yang terpisah dengan koma, maka pada backend saya looping dahulu dengan mengubah variabel menjadi array dari hasil explode.

POS

POS memiliki 2 tipe :
1. Online POS

The screenshot shows the TOPSYSTEM POS interface. At the top, there's a header with 'TOPSYSTEM POINT OF SALE', a clock showing '11:10 : 51', the date 'Friday 30 December 2022', and a user profile 'Hallo, ADMIN'. Below the header, there's a 'Divisi' dropdown set to 'SHOPEE'. To the right, there's a 'Customer' section with a 'Customer' dropdown showing 'GHALY TESTING 0813335533 RESELLER' and a 'Check' button. Below it, there's a 'Refund / Penarikan' section with a 'Ketik minimal 6 huruf invoice' field and a 'Store' dropdown set to 'TOPSCORE DEPOK'. To the right of the 'Store' dropdown, there's a 'Store' section with a 'Store' dropdown set to 'TOPSCORE DEPOK'. Below the 'Store' section, there's a 'Produk' section with a 'Ketik minimal 3 huruf pertama nama artikel' field and an 'Invoice (ketik 5 angka invoice)' field. Below the 'Produk' section, there's a table with columns: 'Produk', 'BIN', 'Disc(%)', 'Jml Beli', 'Nameset', 'Marketplace', 'Harga', and 'Subtotal'. The table has one row with the product '[MILLS] NECROM 99001 NAVY 42', BIN '[ML083]', Disc(%) '0', Jml Beli '1', Nameset '0', Marketplace '0', Harga '149,000', and Subtotal '149,000'. To the right of the table, there's a 'Total Item' section with 'Total Item' '1', 'Subtotal' '149,000', 'Nameset' '0', 'Ongkir' '0', and 'Voc' 'Enter stt input'. At the bottom right, there's a 'TOTAL' section with 'TOTAL' '149,000'.

logika online adalah mereka bisa langsung pick artikel pada pos dimanapun posisi artikel tersebut berada pada suatu store, jika divisi marketplace wajib mengisi field marketplace disebelah nameset, kemudian harga tersebut akan terekam pada database pos_transaction_details field pos_td_marketplace_price, namun jika tidak diisi akan masuk ke field pos_td_discount_price, field marketplace_price lebih diutamakan, jika kosong baru baca ke pos_td_discount_price.

Setiap artikel yang dipanggil akan berasal dari store yang mereka pilih, jika mereka memilih beda store dari tempat mereka bertugas, maka akan menjadi cross order dimana field cross_order pada ts_pos_transactions akan menjadi 1 dan memiliki st_id_ref sesuai store yang mereka tuju.

```
function fetch(Request $request)
{
    if($request->get('query'))
    {
        $exception = ExceptionLocation::select('pl_code')
        ->leftJoin('product_locations', 'product_locations.id', '=', 'exception_locations.pl_id')->get()->toArray();

        $query = $request->get('query');
        $type = $request->get('type');
        $std_id = $request->get('std_id');
        $st_id = $request->get('st_id');
        $bgl_id = null;
        $bgl_price = null;
        if (!empty($std_id)) {
            $st_id = $std_id;
        } else {
            $st_id = Auth::user()->st_id;
        }
        if ($st_id != Auth::user()->st_id) {
            $cross = 'true';
        } else {
            $cross = 'false';
        }
        $data = ProductStock::select('p_name', 'p_color', 'p_sell_price', 'p_price_tag', 'products.psc_id', 'ps_price_tag', 'p
        ->join('products', 'products.id', '=', 'product_stocks.p_id')
        ->join('sizes', 'sizes.id', '=', 'product_stocks.sz_id')
        ->join('brands', 'brands.id', '=', 'products.br_id')
        ->join('product_location_setups', 'product_location_setups.pst_id', '=', 'product_stocks.id')
        ->join('product_locations', 'product_locations.id', '=', 'product_location_setups.pl_id')
        ->where('product_locations.st_id', '=', $st_id)
        ->where('pls_qty', '>', '0')
        ->whereNotIn('pl_code', $exception)
        ->whereRaw('CONCAT(br_name, " ", p_name, " ", p_color, " ", sz_name) LIKE ?', "%$query%")
        ->groupBy('product_stocks.id')
        ->limit(13)
        ->get();
    }
}
```

Diatas merupakan contoh script pemanggilan artikel ketika user mengetik pencarian artikel, dimana seperti penjelasan data stok, saya memanggil exceptionlocation agar tidak ditampilkan. Kemudian ada beberapa id yang saya panggil seperti std_id untuk mengecek diskon artikel tersebut pada product_discount_details, st_id untuk posisi store bin. Jika artikel yang dipanggil terdapat setup diskon, maka gunakan harga bandrol atau price_tag untuk memotong value diskon, jika tidak ada gunakan sell_price, harga yang diutamakan ada pada tabel ts_product_stocks ps_price_tag dan ps_sell_price, jika kosong baca ke tabel utama ts_products untuk

p_price_tag dan p_sell_price.

Kemudian setelah user CO stok akan langsung memotong dan helper baru akan mengambilkan item pada invoice.

Function terkait co ada pada PointOfSaleController.php

untuk online saveTransaction() dan saveTransactionDetail()

```
public function saveTransaction(Request $request)
{
    $pm_id = $request->_pm_id;
    $cp_id = $request->_cp_id;
    $std_id = $request->_std_id;
    $cust_id = $request->_cust_id;
    $cross = $request->_cross;
    if ($cust_id == null) {
        $cust_id = 60;
    }
    $st_id = $request->_st_id;
    $pt_id_complaint = $request->_pt_id_complaint;
    $exchange = $request->_exchange;
    if (!empty($pt_id_complaint)) {
        if ($exchange == 'true') {
            DB::table('pos_transactions')->where('id', $pt_id_complaint)->update([
                'pos_refund' => '1',
                'pos_status' => 'EXCHANGE'
            ]);
        } else {
            DB::table('pos_transactions')->where('id', $pt_id_complaint)->update([
                'pos_refund' => '1',
                'pos_status' => 'REFUND'
            ]);
        }
    }
}
```

pada save transaction saya menangkap semua item yang telah diinput oleh user, termasuk jika ada item cross order maupun terdapat invoice yang direfund untuk digantikan dengan invoice tercheckout saat ini. jika terdeteksi ada invoice yang direfund dengan invoice ini maka otomatis invoice lama akan diupdate dengan status exchange atau refund sesuai kasus dilapangan, exchange berarti tukar item ada pemanggilan item pada invoice lama dan item baru pengganti, sedangkan refund hanya pengembalian barang dan value pasti minus. Jika user menggunakan voucher akan direkam pada tabel voucher_transactions setelah invoice berhasil tercreate.

Pada function pertama ini baru untuk inisialisasi sebuah invoice dan direturn kembali ke frontend id invoice untuk merekam detail artikel pada invoice tersebut.

```

$pos_td_description = null;
if (session()->get('voc_item') != Auth::user()->id.'-'. $pst_id) {
    if ($pst_id == $voc_pst_id) {
        session()->put('voc_item', Auth::user()->id.'-'. $pst_id);
        if ($item_qty > 1) {
            $pos_td_discount_price = (($item_qty-1) * $price) + $voc_value;
            $pos_td_description = ($item_qty-1)." x ".$price." | 1 x ".$voc_value."";
        } else {
            $pos_td_discount_price = $item_qty*$voc_value;
            $price = $voc_value;
            $pos_td_description = "1 x ".$voc_value."";
        }
    } else {
        $pos_td_discount_price = $item_qty*$price;
    }
} else {
    $pos_td_discount_price = $item_qty*$price;
}

$create = PosTransactionDetail::create([
    'pt_id' => $pt_id,
    'pst_id' => $pst_id,
    'pl_id' => $pl_id,
    'pos_td_qty' => $item_qty,
    'pos_td_sell_price' => $price,
    'pos_td_discount' => $discount,
    'pos_td_discount_price' => $pos_td_discount_price,
    'pos_td_marketplace_price' => $marketplace_price,
    'pos_td_nameset_price' => $nameset_price,
    'pos_td_nameset' => $nameset,
    'pos_td_description' => $pos_td_description,
    'pos_td_total_price' => $pos_td_discount_price+$nameset_price,
    'created_at' => date('Y-m-d H:i:s'),
]);
if (empty($create)) {

```

Pada saveTransactionDetail saya merekam detail artikel pada tabel pos_transaction_details dan membaca jika ada voucher digunakan maka akan ada deskripsi diselipkan terkait detail tersebut. Setelah item berhasil disimpan saya masih melakukan beberapa aktifitas diantaranya menambahkan qty sold terjual pada ts_products untuk melakukan sorting bestseller pada website topscore.id. Kemudian saya check lagi terkait apakah transaksi tersebut murni normal atau transaksi refund / exchange, maka saya cek pada invoice tracking (ts_product_location_setup_transactions) terkait status complaint pada invoice lama, jika ada maka saya akan menambahkan tracking baru terkait invoice lama dengan status refund/exchange, dan status complaint saya kembalikan menjadi done, complaint bersifat sementara untuk penanda artikel apa yang mau direfund/exchange, sehingga jika sudah selesai akan dikembalikan. Tetapi jika normal tidak ada complaint tinggal insert saja ke stock tracking dengan status waiting online untuk lanjut ke tahap pengambilan oleh helper.

```

if (empty($create)) {
    $sold = DB::table('products')->select('products.id as p_id', 'sold')
        ->leftJoin('product_stocks', 'product_stocks.p_id', '=', 'products.id')
        ->where('product_stocks.id', '=', $pst_id)
        ->get()->first();
    if (empty($sold)) {
        $update_sold = DB::table('products')->where('id', '=', $sold->p_id)
            ->update([
                'sold' => ($sold->sold + $item_qty)
            ]);
    }
    $pls_id = ProductLocationSetup::select('id')->where('pst_id', $pst_id)->where('pl_id', $pl_id)->get();
    $check_plst = ProductLocationSetupTransaction::where([
        'id' => $pls_id,
        'plst_status' => 'COMPLAINT'
    ])->exists();
    if ($check_plst) {
        $plst_data = ProductLocationSetupTransaction::select('pls_id', 'u_id_helper', 'u_id_packer', 'pt_id')
            ->where('id', => $pls_id)
            ->get()->first();
        if ($final_price < 0 AND $exchange != 'true') {
            $insert = ProductLocationSetupTransaction::insert([
                'pt_id' => $plst_data->pt_id,
                'pls_id' => $plst_data->pls_id,
                'u_id_packer' => $plst_data->u_id_packer,
                'u_id_helper' => $plst_data->u_id_helper,
                'u_id' => Auth::user()->id,
                'u_id_refund' => Auth::user()->id,
                'plst_qty' => abs($item_qty),
                'plst_type' => 'OUT',
                'plst_status' => 'REFUND'
            ]);
        }
    }
}

```


2. Offline POS

The screenshot displays the TOPSYSTEM POINT OF SALE interface in offline mode. The header shows the system name, time (14:05:55), date (Friday 30 December 2022), and user (Halo, Ghaly Fadhilah). The sidebar on the right shows the store name 'TOPSCORE BEKASI' and a summary of the current transaction: 2 items, subtotal 398,000, and a total of 398,000. The main area shows a 'LOKASI' dropdown set to 'WAITING', a table of products with columns for product name, stock, quantity, name, price, and subtotal, and a 'Bayar' button at the bottom.

Perbedaan jika offline, setiap item yang dipanggil melalui opsi Lokasi Waiting wajib dipick up dahulu pada menu Data Stok dan sudah diambilkan helper dengan status waiting offline, kecuali 3 opsi lainnya dari store, wagon dan b1g1 dapat dipick langsung pada POS dan otomatis status waiting for checkout, jika yang bukan dari waiting dihapus dari POS status akan instock.

Ketika pemanggilan item ke POS Online ada pada function fetch(), jika offline ada pada fetchWaiting(), ada sedikit perbedaan yaitu pada offline melakukan check terhadap artikel b1g1, jika terdeteksi maka akan disisipkan pada property data-b1g1_id dan data-b1g1_price untuk diolah pada frontend.

```

if (!empty($set_discount)) {
    if (date('Y-m-d') <= $set_discount->pd_date) {
        if (!empty($row->ps_price_tag)) {
            $price_tag = $row->ps_price_tag;
        } else {
            $price_tag = $row->p_price_tag;
        }
        if ($set_discount->pd_type == 'percent') {
            $sell_price = $price_tag - ($price_tag/100 * $set_discount->pd_value);
        } else if ($set_discount->pd_type == 'amount') {
            $sell_price = $price_tag - $set_discount->pd_value;
        } else {
            $sell_price = $price_tag;
            $b1g1_id = $row->pst_id;
            $b1g1_price = $sell_price;
        }
    }
}

if ($item_type == 'waiting') {
    $status = '<span class="btn-lg btn-warning">'. $row->plst_status. '</span>';
    $bin = '<span class="btn-lg btn-info">'. strtoupper($row->pl_code). '</span>';
} else {
    $status = '';
    $bin = '<span class="btn-lg btn-info">'. $row->pls_qty. '</span>';
}

$output .= '
<li><a class="btn btn-sm btn-inventory col-12" data-b1g1_id="'. $b1g1_id. '" data-b1g1_price='

```

Pada jquery frontend dapat dilihat ketika id add_to_item_list diklik ataupun via barcode maka ada logika untuk melakukan check terhadap apakah artikel b1g1 atau bukan. Selain itu juga ajax untuk mengubah status di stock tracking dari waiting offline menjadi waiting for checkout.

	<pre> if (bgl_id != '' && bgl_price != '') { bgl_temp.push(bgl_price); highlight = 'background:#ffc107; color:#000; font-weight:bold; border-radius:20px;'; bgl_mode = 'bgl_mode'; bgl_temp = bgl_temp.sort((a, b) => b-a); var bgl_total_row = bgl_temp.length; var bgl_qty_total = 0; if (bgl_total_row > 0) { jQuery('#orderTable tr').each(function(index, row) { var bgl_qty = parseFloat(jQuery(row).find('.item_qty').val()); if (typeof bgl_qty === 'undefined' bgl_qty == '' isNaN(bgl_qty)) { bgl_qty = 0; } if (jQuery(row).hasClass('bgl_mode')) { bgl_qty_total += bgl_qty; if (parseFloat(sell_price) >= parseFloat(bgl_temp[0])) { sell_price = sell_price; jQuery(row).find('.sell_price_item').text('0'); } else { sell_price = 0; } jQuery(row).find('.item_qty').trigger('change'); } }); if (parseFloat(bgl_qty_total) >= 2) { swal('1 Invoice 1 BGl', 'Silahkan checkout item diinvoice yang baru apabila leb return false; } } } </pre>
Reseller Transaksi	<p>Untuk pengelolaan datatable sama saja seperti stock data mungkin hanya tabel customnya yang berbeda. Namun yang perlu diperhatikan pada function doCheckout jika user telah menggunakan saldo maka otomatis stok sudah terpotong saat reseller checkout pada sistem mereka dan validasi disini hanya untuk create internal invoice, namun jika user menggunakan bank transfer stok belum terpotong, melainkan dipotong pada function ini. cara pemotongan qty secara random adalah dengan melooping tabel product_location_setups yang sudah difilter \$exception, kemudian potong dan update qty sesuai artikel yang dibeli oleh reseller.</p>
HPP	<p>HPP diambil dari 2 tabel yaitu ts_purchase_order_details dan ts_purchase_order_detail_statuses dengan metode average, namun beberapa fitur juga ada yang hanya saya ambil berdasarkan hpp terbaru namun lupa dimana, ada juga yang dari poads_total_price dibagi jml qty po jika pembacaan average dari statuses terlalu lelet. Tetapi intinya pengambilan hpp hanya dari 2 table ini.</p>
Export Data	<p>Untuk membuat export data silahkan create file pada app/Exports atau dengan menggunakan perintah php artisan make:export NamaExport, kemudian silahkan tambahkan model untuk eloquent / query builder untuk memanggil data yang akan diexport, jika ingin export dengan judul gunakan WithHeadings, nanti bisa dilihat contoh file export didalam folder app/Exports.</p> <p>Setelah itu tinggal panggil plugin Excel Maatwebsite beserta file Export, contoh shopee</p>

	<pre> use App\Imports\ShopeeImport; use App\Exports\ShopeeExport; use Maatwebsite\Excel\Facades\Excel; class ShopeeController extends Controller { </pre> <p>kemudian buat sebuah function untuk menangkap request (jika ada filter tertentu) dan return Excel::download untuk download results dari class Export yang sudah dibuat dengan nama apapun.</p> <pre> public function exportData(Request \$request) { \$tg_id = \$request->post('tg_id'); return Excel::download(new ShopeeExport(\$tg_id), 'shopee_new_stock.xlsx'); } </pre>
Import Data	<p>Untuk membuat import data sebenarnya hampir sama dengan Export hanya saja jika diexport kita bisa menggunakan WithHeading untuk menambahkan baris judul, diimport kita bisa menggunakan WithStartRow untuk memulai pembacaan pada template pada row keberapa. Kemudian didalam collection kita bisa looping row template yang telah diimport, dari hasil data tersebut terserah mau kita apakan terhadap database. Semua contoh ada pada folder app/Imports.</p> <p>Kemudian pemanggilan pada controller juga sama namun plugin yang digunakan Excel::import</p> <pre> public function importData(Request \$request) { if (request()->hasFile('shopee_template')) { \$tg_id = \$request->post('tg_id'); \$storage = \$request->post('storage'); \$import = new ShopeeImport(\$tg_id, \$storage); Excel::import(\$import, request()->file('shopee_template')); if (\$import->getRowCount() >= 0) { \$r['status'] = '200'; } else { \$r['status'] = '400'; } } else { \$r['status'] = '500'; } return json_encode(\$r); } </pre> <p>untuk getRowCount() itu sebenarnya hanya salah satu cara validasi yang saya gunakan untuk mengetahui import berhasil atau tidak, namun ada banyak cara lainnya yang bisa digunakan sesuai kreatifitas.</p>
Diskon Artikel	<p>Untuk diskon artikel akan memiliki keterkaitan terhadap setup harga artikel. Untuk pembacaan harga diutamakan dahulu dari table ts_product_stocks jika kosong baru ke ts_products. Termasuk juga harga baca dahulu apakah ada ps_price_tag dan ps_sell_price , jika tidak ada baru beralih ke p_price_tag dan p_sell_price , ps ke p.</p> <p>jika artikel terlibat diskon maka potong dari price_tag, namun jika tidak gunakan sell_price.</p> <p>Untuk logika pengecekan pada script saat ini mengutamakan yang ada b1g1 dahulu dengan tanggal yang belum expired, jika ada utamakan b1g1, jika</p>

	tidak ada baru arahkan ke diskon berikutnya sesuai std_id, jika sesuai std_id juga tidak ditemukan, maka arahkan ke diskon yang bersifat all atau hanya disetup st_id tanpa ada std_id.
Harga Jual	Seperti yang dijelaskan sebelumnya, utamakan harga dari ts_product_stocks untuk price_tag dan sell_price, jika kosong baru ke ts_products.
Cross Order	Untuk cross order biasanya jika user online melakukan pick beda store pada POS, jika terdeteksi maka otomatis akan ditandai untuk mengubah pos_transactions field cross order menjadi 1 dan st_id_ref sesuai store yang dituju, kemudian pada menu cross order hanya tampilkan invoice yang memiliki value cross_order 1. namun khusus ke Urban anak Depok dapat langsung CO dan dianggap sebagai invoice biasa tanpa cross order.
Upcloud Server	Biasanya ada kasus CPU server sangat tinggi secara tiba – tiba tanpa ada pemicu sehingga menyebabkan sistem lemot bahkan tidak selesai loadingnya, bisa dipengaruhi oleh cloud server, restart untuk membenahi atau tambah kapasitas CPU (nambah kapasitas biasanya ada biaya)
CronJob	Jika kedepannya ada membuat cronjob dan tidak berjalan, bisa coba direstart dahulu server upcloudnya.

* jika menemukan ada script controller yang terminify (agar performance meningkat) silahkan format pada web ini <https://codebeautify.org/php-beautifier> agar dapat dibaca.