

## BAB 3

### PELAKSANAAN KERJA MAGANG

#### 3.1 Kedudukan dan Organisasi

PT. Hashmicro Solusi Indonesia memiliki divisi *software implementation consultant intern* yang terdiri atas mahasiswa-mahasiswa yang sedang melakukan magang di perusahaan tersebut dan disupervisi langsung oleh *project coordinator, business analyst project manager*. Divisi ini bertanggung jawab dalam membantu mengatasi permasalahan klien ketika klien mengalami suatu masalah dalam menggunakan sistem dan menemukan isu dalam penggunaan sistem. Selain itu, divisi ini juga bertanggung jawab dalam memperbaiki isu yang dialami oleh klien dan apabila divisi ini tidak dapat mengatasi isu major. Maka, akan dialihkan ke *software programmer*.

Pada saat berada di divisi ini, koordinasi dilakukan melalui Whatsapp yang terdiri atas CEO (*Chief Executive Officer*), *project manager, business analyst* dan *project coordinator*. Untuk projek yang ditangani, divisi ini melakukan diskusi lebih dalam melalui Skype untuk melakukan pembahasan dengan divisi lain. Setiap hari Kamis/Jumat akan dilakukan rapat bersama melalui *Google Meet* untuk membahas persoalan yang terjadi dalam penggunaan sistem dengan klien. Lalu, biasanya setiap ada isu dalam penggunaan sistem akan dilakukan diskusi secara bersama-sama dengan divisi lain untuk menemukan solusi dari permasalahan yang terjadi melalui Skype.

Untuk melakukan komunikasi dengan klien dari perusahaan, divisi ini melakukan komunikasi melalui Whatsapp untuk melakukan *update* mengenai *time-line*, hal yang akan dilakukan kedepannya dan juga proses yang telah dikerjakan oleh divisi ini. Di dalam grup Whatsapp, terdiri atas klien, *project coordinator, project manager*, dan semua anggota divisi ini.

#### 3.2 Tugas yang Dilakukan

Proyek awal dalam magang ini adalah melanjutkan pengembangan sistem *enterprise resource planning* yang telah dibuat oleh *project manager* sebelumnya. Tujuan dilakukan proyek ini adalah untuk mempermudah klien dalam menjalankan aktivitas jual beli pada setiap outlet dan juga memudahkan dalam merekap setiap barang yang masuk dan keluar dari pabrik dengan menggunakan modul *point of sale*

dan *inventory*. Hal-hal yang dibutuhkan dari fitur *point of sale* adalah *dashboard* yang digunakan untuk melihat *point of sale* yang diassign ke setiap user kasir.

Lalu, ada *orders* untuk melihat setiap order yang masuk dari setiap outlet. Selanjutnya, *products* untuk menampilkan produk-produk yang dijual dalam sistem *point of sale*. Kemudian, *programs* yang berfungsi untuk membuat promo yang bisa disesuaikan dengan keinginan klien. Lalu, ada *report orders* dan *POS payment method* untuk melihat data penjualan per harinya dan melihat data penjualan customer dengan metode pembayaran yang berbeda per harinya.

Untuk *inventory*, hal-hal yang dibutuhkan adalah *material request* untuk membuat dan menampilkan produk-produk yang dibutuhkan oleh *store leader* dari setiap outlet untuk didistribusikan dari pabrik. Kemudian, *internal transfer* yang berguna untuk melakukan pengiriman barang dari outlet ke outlet atau dari pabrik ke outlet ataupun dari outlet ke booth/moko. Kemudian, ada *products* yang berfungsi untuk menampilkan produk-produk yang didistribusikan oleh pabrik pembuatan.

Selanjutnya, ada *inventory adjustment plus* yang berfungsi untuk menambahkan jumlah stok produk yang tersedia pada pabrik bandung. Kemudian, ada *reports* yang terdiri atas *daily stock report*, *inventory stock*, dan *stock moves* yang masing-masing memiliki fungsi yang berbeda. Untuk *daily stock report* berfungsi untuk melihat produk-produk yang masuk pada outlet tertentu dan produk yang terjual berapa per harinya.

Untuk *inventory stock* berfungsi untuk menampilkan jumlah *available quantity* produk dari setiap outlet. Terakhir, untuk *stock moves* berguna untuk melihat pergerakan barang yang terjadi pada setiap outlet.

Fitur yang terdapat pada sistem ini memiliki ketentuannya masing-masing, dimana semua halaman fitur diatas dapat dilihat, dibuat, dihapus dan diperbarui oleh administrator. Sedangkan untuk fitur *dashboard* POS hanya bisa diakses oleh kasir dan *store leader* dari masing-masing outlet yang telah diassign termasuk administrator. Untuk fitur *orders*, *session*, *report orders* dan *POS Order Payment Method*, *material request*, *internal transfer*, *inventory stock*, *daily stock report*, dan *stock moves* dapat diperbarui dan diakses oleh akun *store leader*.

Selain itu, untuk fitur *material request*, *internal transfer*, dan *inventory stock* juga dapat diakses oleh akun *product distribution* dan *production packaging*. Untuk fitur *inventory adjustment plus* hanya dapat diakses oleh akun *production packaging* dan juga administrator.

Bahasa pemrograman yang digunakan adalah Odoo, Javascript, PostgreSQL dan juga Python. Odoo digunakan sebagai dasar dari proyek tersebut / yang

menampilkan semua program yang telah dibuat. Javascript digunakan untuk membuat tampilan dari sistem tersebut. PostgreSQL digunakan untuk menyimpan data. Python digunakan sebagai bahasa utama dalam menjalankan setiap aktivitas di sistem.

Tabel 3.1. Realisasi Kerja Magang Sistem ERP

Minggu Ke -	Pekerjaan yang dilakukan
1	Mempelajari <i>flow</i> sistem <i>enterprise resource planning</i> secara keseluruhan, mempelajari bahasa pemrograman yang digunakan dan mempelajari setiap detail sistem yang disepakati dengan klien
2	Mempelajari proyek baru yang dipegang dan mempelajari setiap detail alur sistem secara keseluruhan khususnya <i>point of sale</i> dan <i>inventory</i> .
3	Mempelajari cara melakukan <i>push module</i> pada Gitlab dan cara melakukan update modul ke sistem menggunakan Putty.
4	Melakukan perbaikan issue-issue minor seperti issue <i>coupon</i> yang tidak bisa digunakan pada saat transaksi, <i>material request line</i> , dan <i>daily stock report</i> .
5	Mempelajari untuk melakukan duplikasi dan merestore <i>database</i> serta melakukan analisa untuk mengatasi issue <i>lagging</i> pada <i>report orders</i> .
6	Menambahkan <i>daily stock report</i> dalam bentuk excel, <i>daily orders</i> untuk <i>orders</i> dalam rentang 7 hari, <i>monthly orders</i> untuk rentang 35 hari dan <i>promotion restriction</i> ketika minimal <i>quantity</i> produk tidak sesuai serta saya mempelajari mengenai cara untuk melakukan <i>revert code</i> agar <i>session report</i> bisa dibuka tanpa perlu melakukan <i>set closing balance</i> .
7	Melakukan pembuatan promosi baru untuk diskon per produk Rp1.000/dus dan menambahkan <i>field cashbox history</i> untuk melakukan pencatatan pada transaksi <i>put money in</i> dan <i>take money out</i> .
8	Melakukan penghapusan pada PostgreSQL pada
Lanjut pada halaman berikutnya	

Tabel 3.1 Realiasi Kerja Magang Sistem ERP (Lanjutan)

Minggu Ke -	Pekerjaan yang dilakukan
9	<p><i>field point earned</i> dan <i>point used</i> yang menyebabkan server <i>lagging</i> karena memakan banyak <i>core</i> pada server.</p> <p>Melakukan perbaikan promosi yang dibuat sesuai dengan keinginan klien misalnya produk A sejumlah 20 dan produk B sejumlah 30 dalam kategori produk yang sama, melakukan perubahan UI (<i>User Interface</i>) pada tampilan POS order payment method report karena mengambil data yang sangat banyak pada server.</p>
10	<p>Memperbaiki <i>access right</i> milik kasir, mengubah PPN, memperbaiki <i>rounding</i> agar bisa melakukan validasi pembayaran dan mempersiapkan untuk <i>testing</i> 5 outlet.</p>
11	<p>Melakukan pembuatan diskon <i>Free</i> UHT dan diskon 10K dengan minimum pembelian sebesar 150 Ribu Rupiah, memperbaiki <i>rounding</i> agar <i>order</i> bisa <i>terpaid</i> secara otomatis, memperbaiki <i>daily stock report</i> agar sesuai dengan <i>stock card</i>.</p>
12	<p>Melakukan perbaikan diskon <i>Free</i> UHT yang sekarang bisa jalan ketika 3 <i>random</i> produk dengan ketentuan 2 produk A dan 1 Produk B secara akumulasi ditotal menjadi tiga produk.</p>
13	<p>Memperbaiki <i>daily stock report</i> pada file PDF dengan menghapus <i>field in</i> dan <i>field out</i>.</p>
14	<p>Libur lebaran.</p>
15	<p>Memperbaiki <i>access right store cashier</i> dan <i>store leader</i>.</p>
16	<p>Memperbaiki <i>access right</i> pada production packaging dan juga <i>product distribution</i>.</p>
17	<p>Melakukan pembersihan <i>database</i> untuk melakukan <i>internal testing</i> oleh klien, menambahkan <i>field additional destination location</i> agar outlet bisa melakukan <i>internal transfer</i> ke outlet lain, dan memperbaiki <i>issue picking error</i> yang muncul akibat transaksi dengan jumlah 0.</p>
18	<p>Memperbaiki <i>access right</i> kasir dengan mengubah "pos.session" menjadi <i>read-only</i>, memperbaiki <i>promotion service</i> menjadi <i>exclude</i> PPN di <i>order</i>, dan memperbaiki selisih barang di <i>stock card report</i> akibat <i>picking error</i>.</p>

Lanjut pada halaman berikutnya

Tabel 3.1 Realiasi Kerja Magang Sistem ERP (Lanjutan)

Minggu Ke -	Pekerjaan yang dilakukan
19	Memperbaiki <i>issue</i> promo <i>Free</i> UHT yang bisa jalan walaupun stok UHT sudah kosong.
20	Memperbaiki <i>flow validate</i> pada halaman <i>internal transfer</i> dan promosi yang bisa digunakan oleh pelanggan spesifik serta promo yang bisa distak banyak dalam satu transaksi.
21	Memperbaiki <i>issue compliment</i> yang dianggap sebagai <i>payment method rounding</i> dan menghilangkan <i>payment method rounding</i> dan <i>compliment</i> pada sistem <i>point of sale</i> .

Pada Tabel 3.1 menunjukkan bahwa pada minggu pertama, mempelajari alur proses keseluruhan dari sistem ERP (*Enterprise Resource Planning*) secara meluas mulai dari *purchasing, inventory, sales, accounting, point of sale, manufacture, construction*, dan *human resource management*. Lalu, diassign ke suatu proyek dan mempelajari secara detail proses *inventory* dan juga *point of sale* dari kebutuhan klien. Selain itu, mempelajari penggunaan bahasa pemrograman yang dipakai dalam Odoo untuk mengetahui secara detail bahasa pemrograman yang digunakan dalam sistem Odoo.

Pada minggu kedua, mempelajari secara mendetail *flow* atau alur sistem dari klien secara keseluruhan yang dijelaskan oleh *project manager* yang memegang proyek ini. Lalu, mempelajari dan menganalisa setiap masalah yang ada pada sistem klien untuk diperbaiki. Selain itu, pada minggu ini, diperkenalkan dengan klien yang melakukan pemesanan sistem agar lebih mudah dalam berkomunikasi.

Pada minggu ketiga, mempelajari cara melakukan *push* modul dari Github dan mengupdate ke server menggunakan aplikasi Putty. Disana, saya diajarkan untuk memasukkan *ip address* dan *port* dari server klien. Kemudian, memasukkan *username* dan *password* serta memasukkan *command* untuk melakukan *push* pada server klien.

Pada minggu keempat, melakukan perbaikan *issue* minor seperti kupon yang tidak bisa digunakan dalam melakukan transaksi. Lalu, masalah *material request line* yang mengalami kendala dimana pada *internal transfer quantity* terdapat masalah yaitu ketika ditambahkan *query* baru, maka akan menimpa *query*

yang lama. Terakhir, *issue* pada *daily stock report* dimana ketika dilakukan *inventory adjustment plus* dan dilakukan transaksi maka tidak tercatat dalam *daily stock report*.

Pada minggu kelima, mempelajari cara untuk melakukan duplikasi *database* milik klien dan melakukan *restore database* untuk melakukan *backup* semisalnya *database* ingin disimpan di dalam komputer. Kemudian, melakukan analisa untuk mengatasi masalah *lagging* ketika membuka *report orders* yang dapat mengganggu kasir dalam melakukan transaksi di semua outlet. Sehingga, ketika menu *report orders* ditekan, maka semua kasir akan mengalami *loading* juga.

Pada minggu keenam, menambahkan *daily stock report* yang berbentuk file excel agar memudahkan dalam melakukan *pengeditan* oleh klien. Lalu, membuat menu baru yaitu *daily orders* yang diset dalam rentang waktu 7 hari untuk mengatasi *issue lagging* pada *report orders*. Selain itu, menambahkan juga menu baru dengan rentang waktu 35 hari dengan nama *daily orders monthly* agar bisa melihat data penjualan dalam sebulan.

Kemudian, memperbaiki *issue* pada promosi dengan menambahkan *warning alert* ketika *product quantity* tidak sesuai jumlahnya. Terakhir, mempelajari bagaimana cara untuk melakukan *revert code* yang telah dibuat pada *session report* agar klien bisa membuka *session report* sebelum melakukan validasi dan *set closing balance*.

Pada minggu ketujuh, melakukan pembuatan diskon *reseller* dengan ketentuan pembelian 50 dus produk maka akan mendapatkan diskon sebesar 1k per dus. Kemudian, menambahkan field *cashbox history* pada bagian *session* untuk melakukan perekapan keuangan yang dilakukan kasir khususnya pada *put money in* dan *take money out*.

Pada minggu kedelapan, setelah melakukan analisa pada server yang *lagging* ketika membuka *report orders*. Ditemukan bahwa penyebab utama adalah pada *field point earned* dan juga *point used*. Maka dari itu, dilakukan penghapusan kedua field tersebut pada PostgreSQL untuk mengatasi *lagging* pada *report orders* yang memakan banyak *core* pada server.

Pada minggu kesembilan, melakukan perbaikan pada diskon *reseller* yang telah dibuat sebelumnya dengan mengset promosi agar bisa jalan ketika produk *random* diakumulasikan menjadi 50 produk. Sebelumnya, diskon hanya bisa jalan ketika satu produk berjumlah 50, bukan secara akumulasi seperti 20 Produk A dan 30 Produk B. Lalu, memperbaiki tampilan *user interface* pada *report POS order payment method* yang mengambil data yang sangat banyak pada server.

Pada minggu kesepuluh, melakukan perbaikan *access right* pada kasir yang sebelumnya *diset* menjadi *manager*. Lalu, mengubah PPN dari 10 persen menjadi 11 persen dikarenakan perubahan pajak dari pemerintah per April 2022. Kemudian, memperbaiki permasalahan *rounding* yang menyebabkan banyak *order* tidak *terpaid* secara otomatis setelah dilakukan perubahan PPN. Terakhir, mempersiapkan segala kebutuhan untuk *internal testing* klien.

Pada minggu kesebelas, melakukan pembuatan diskon baru yaitu *Free UHT* dan diskon 10k dengan minimum pembelian sebesar 150k. Kemudian, memperbaiki permasalahan *rounding* yang belum sempurna pada minggu lalu yang menyebabkan kasir tidak bisa melakukan validasi *closing balance*. Lalu, memperbaiki *daily stock report* agar sesuai dengan jumlah *stock card* dikarenakan ada perbedaan jumlah produk yang signifikan pada *daily stock report*.

Pada minggu keduabelas, melakukan perbaikan pada diskon *Free UHT* dengan memperbaiki kategori produk sehingga hanya produk tertentu yang masuk ke dalam diskon ini. Kemudian, melakukan *set* promosi agar bisa diakumulasi seperti diskon *reseller* misalnya 2 produk A dan 1 produk B bukan 3 produk A saja.

Pada minggu ketigabelas, melakukan penghapusan kolom *field in* dan *field out*. Sekarang, hanya menyisakan stok *balance* akhir saja. Dikarenakan pada *stock card* sudah tertera untuk *field in* dan *field out*.

Pada minggu keempatbelas, tidak ada kegiatan apa-apa dikarenakan libur lebaran dalam rentang waktu seminggu.

Pada minggu kelimabelas, memperbaiki *access right* kasir dan juga *store leader*. Kasir sekarang tidak bisa mengakses halaman menu yang tidak boleh diakses. Jadi, kasir hanya memiliki *dashboard kasir saja*. Untuk *store leader* bisa mengecek *order* yang masuk dan juga *session* dari kasir outlet masing-masing. Kemudian, *store leader* juga bisa melakukan *material request* ke pabrik Bandung untuk meminta stok barang.

Pada minggu keenambelas, memperbaiki *access right* pada *production packaging* dan *product distribution*. *Access right* diperbaiki dengan cara menghilangkan menu-menu yang tidak dapat diakses oleh kedua akun tersebut.

Pada minggu ketujuhbelas, melakukan pembersihan *database* untuk persiapan *internal testing* yang akan dilakukan oleh tim klien. Data-data yang dibersihkan meliputi *order*, *session*, *internal transfer*, dan juga *material request*. Kemudian, menambahkan *field additional destination location* pada setiap *user store leader* agar bisa melakukan *internal transfer* ke outlet lain tanpa mempengaruhi *source location*.

Terakhir, memperbaiki *issue picking error* yang muncul akibat adanya transaksi dengan jumlah 0. Sehingga, menyebabkan *session* muncul *picking error* disebabkan sistem tidak bisa mengolah produk dengan jumlah 0. Hal ini diselesaikan dengan cara ketika *order* dilakukan *payment* dalam jumlah barang 0. Maka, di menu orders tidak akan muncul transaksi dengan jumlah 0.

Pada minggu kedelapanbelas, memperbaiki *access right* kasir dengan membuat object *pos.session* dan *pos.config* menjadi *read-only* dikarenakan kasir bisa melakukan pembuatan promosi sendiri dan mengubah *pricelist* yang telah diset secara global. Kemudian, memperbaiki *promotion service* menjadi *exclude PPN*. Sehingga, pada struk pembelian tercantum diskon sebelum PPN ditambah dengan PPN dari pembelian pelanggan.

Jadi, di halaman *report orders* tertera promosi yang *exclude PPN* sesuai dengan permintaan klien. Terakhir, memperbaiki selisih stok barang pada *stock card* yang muncul akibat dari *picking error*. Hal ini disebabkan karena ketika ditekan tombol *process picking error* maka barang akan diproses ketika ditekan tombol tersebut. Seharusnya, ketika ditekan tombol *process picking error*, maka akan memotong stok barang di hari terjadi *picking error*.

Pada minggu kesembilanbelas, memperbaiki *issue* pada promo *free UHT* dimana promo bisa jalan ketika produk UHT kosong. Seharusnya, promo hanya berjalan ketika stok dari UHT ada. Untuk itu, *issue* ini diselesaikan dengan cara menambahkan *warning* ketika produk UHT benar-benar habis dalam POS.

Pada minggu kedua puluh, memperbaiki flow *validate* pada halaman *internal transfer* sehingga sekarang *store leader* yang membuat *material request* tidak bisa melakukan validasi barang sendiri dari awal hingga akhir. Kemudian, memperbaiki *issue* dimana ketika *store leader source location* melakukan *internal transfer* ke *store leader destination location*, maka *store leader destination location* tidak bisa melakukan validasi barang yang dilakukan oleh *store leader source location*.

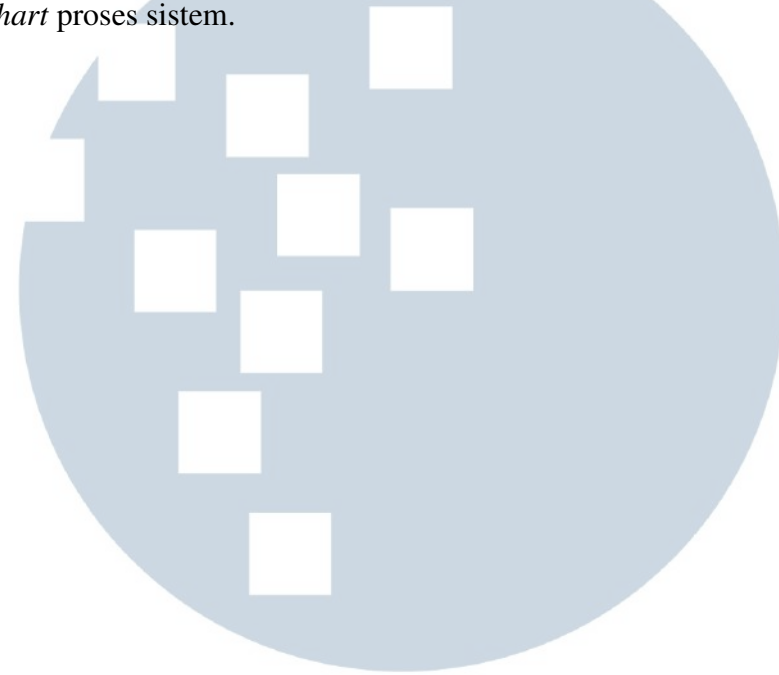
Terakhir, memperbaiki promo yang bisa digunakan berulang-ulang kali dalam satu transaksi dengan menambahkan *field checkbox stackable in promotion*. Ketika *field stackable in promotion* dicentang, maka promo bisa digabungkan dengan promo lain. Kemudian, membuat promo agar bisa diset untuk *customer* tertentu dengan menambahkan *field dedicated customer*.

Pada minggu terakhir, memperbaiki *issue compliment* yang dianggap sebagai *rounding* pada *orders* dan *report orders*. Kemudian, menghilangkan *payment method rounding* dan *compliment* pada sistem POS. Ini sebagai permintaan klien agar tim klien tidak kesusahan dalam merekap uang penjualan.



### 3.3 Uraian Pelaksanaan Magang

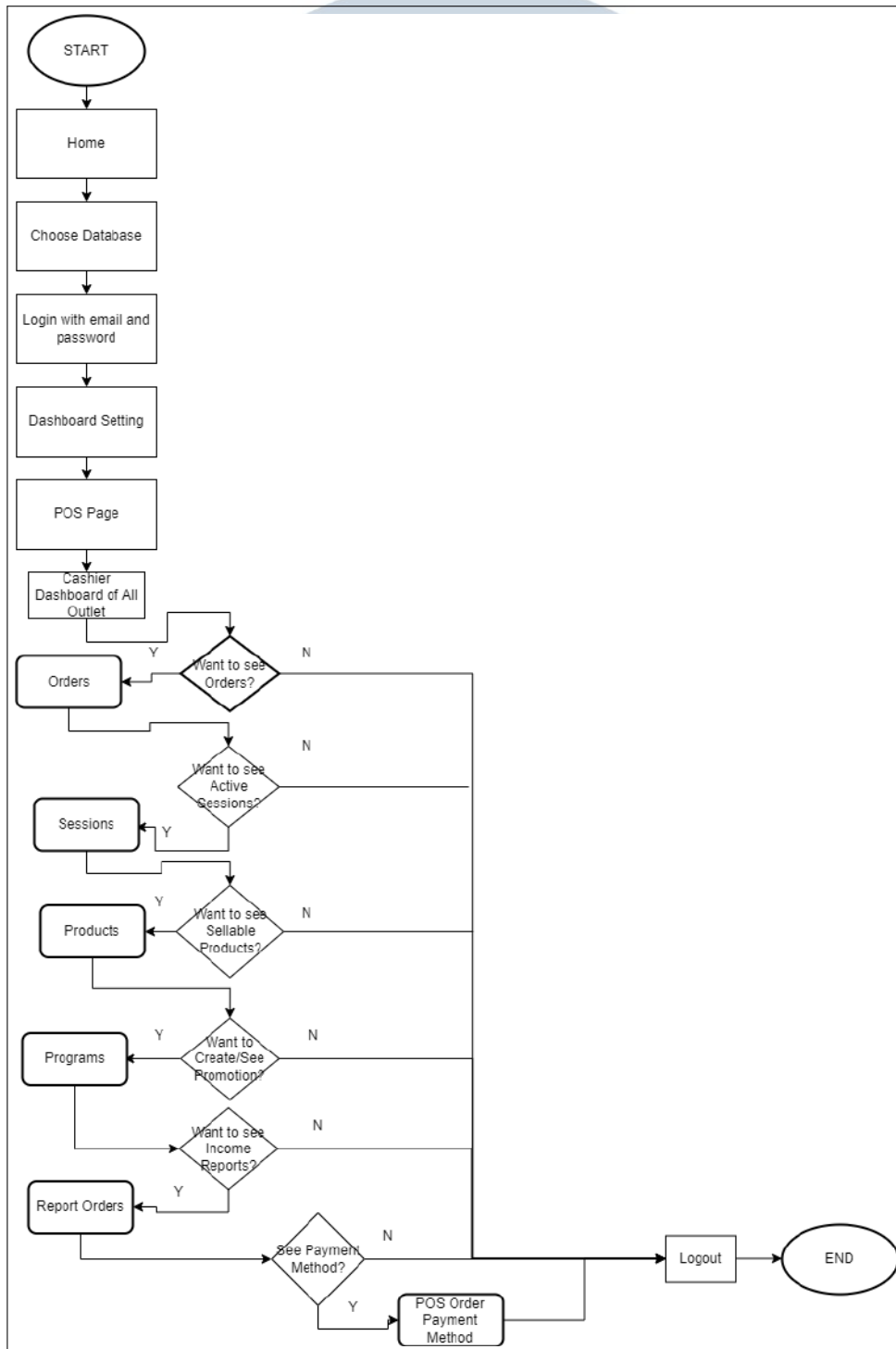
Proses kerja magang selama berada di PT Hashmicro Solusi Indonesia telah dilakukan selama 5 bulan. Uraian pengerjaan terdiri atas flowchart setiap akun dan juga *flowchart* proses sistem.



UMMN

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA

## A. Flowchart Akun



Gambar 3.1. Flowchart sistem sebagai administrator pada halaman point of sale

Berdasarkan Gambar 3.1, pengguna akan memasuki *home* dan memilih *database* yang ingin digunakan oleh pengguna. Kemudian, pengguna melakukan *login* menggunakan email dan password yang telah *diregister* pada akun admin. Lalu, ketika sudah melakukan *login*, pengguna akan masuk ke dalam *dashboard setting*.

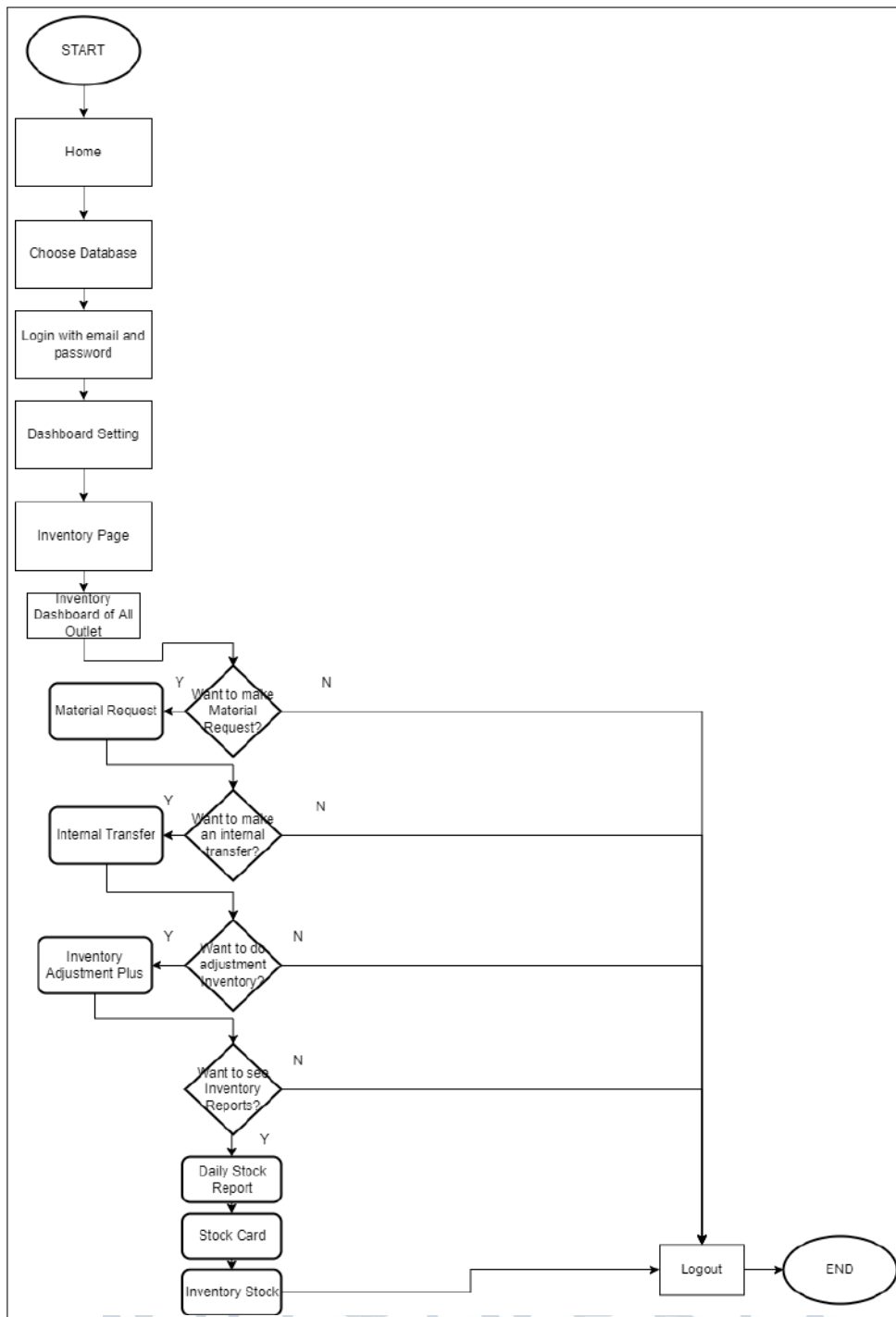
Pada *dashboard setting*, pengguna dapat menambahkan pengguna baru atau mengubah *email* dan *password* pengguna lain pada menu *users*. Kemudian, pengguna dapat memilih modul mana yang ingin dibuka yang meliputi POS dan *inventory*. Jika pengguna ingin membuka modul POS terlebih dahulu, maka pengguna akan ditampilkan dengan *dashboard POS* yang meliputi semua outlet yang ada.

Kemudian, jika pengguna ingin melihat semua transaksi yang masuk ke setiap outlet, pengguna dapat melihat pada menu *orders*. Lalu, jika pengguna sebagai admin ingin melihat data *session* secara keseluruhan untuk seluruh outlet, maka pengguna dapat melihat ke dalam menu *sessions*. Jika pengguna merasa tidak perlu melihatnya, maka pengguna dapat langsung melakukan *logout*.

Kemudian, jika pengguna masih ingin meneruskan dalam penggunaan sistem, maka pengguna dapat melanjutkan untuk melihat produk apa saja yang dijual dalam sistem POS. Pengguna dapat melihat produk yang dijual pada menu *products*. Disana terdapat beberapa pilihan produk yang tersedia dengan disertai harga dari masing-masing produk.

Lalu, jika pengguna ingin menambahkan promosi baru atau ingin melihat promosi yang berlaku di semua outlet, maka pengguna dapat membuka menu *programs* pada *tab promotions*. Lalu, jika pengguna ingin meneruskan untuk melihat *report* penjualan yang masuk ke semua outlet, maka pengguna dapat menekan tombol *orders* pada *tab reports* jika ingin melihat produk apa saja yang terjual dari cabang yang berbeda dengan total harga yang terjual.

Jika pengguna ingin melihat besaran harga berdasarkan metode pembayaran yang digunakan oleh pelanggan, maka pengguna dapat menekan tombol menu *POS order payment method*. Kemudian, jika pengguna tidak ingin melanjutkan ke modul *inventory*, maka pengguna dapat melakukan *logout* untuk keluar dari akun tersebut.



Gambar 3.2. Flowchart sistem sebagai administrator pada halaman inventory

Berdasarkan Gambar 3.2, pengguna juga dapat melihat modul *inventory* sebelum membuka modul *point of sale*. Apabila pengguna membuka modul *inventory*, maka pengguna akan ditampilkan dengan *dashboard inventory* yang berisikan pergerakan barang setiap outlet. Jika pengguna ingin mengajukan atau melihat *ma-*

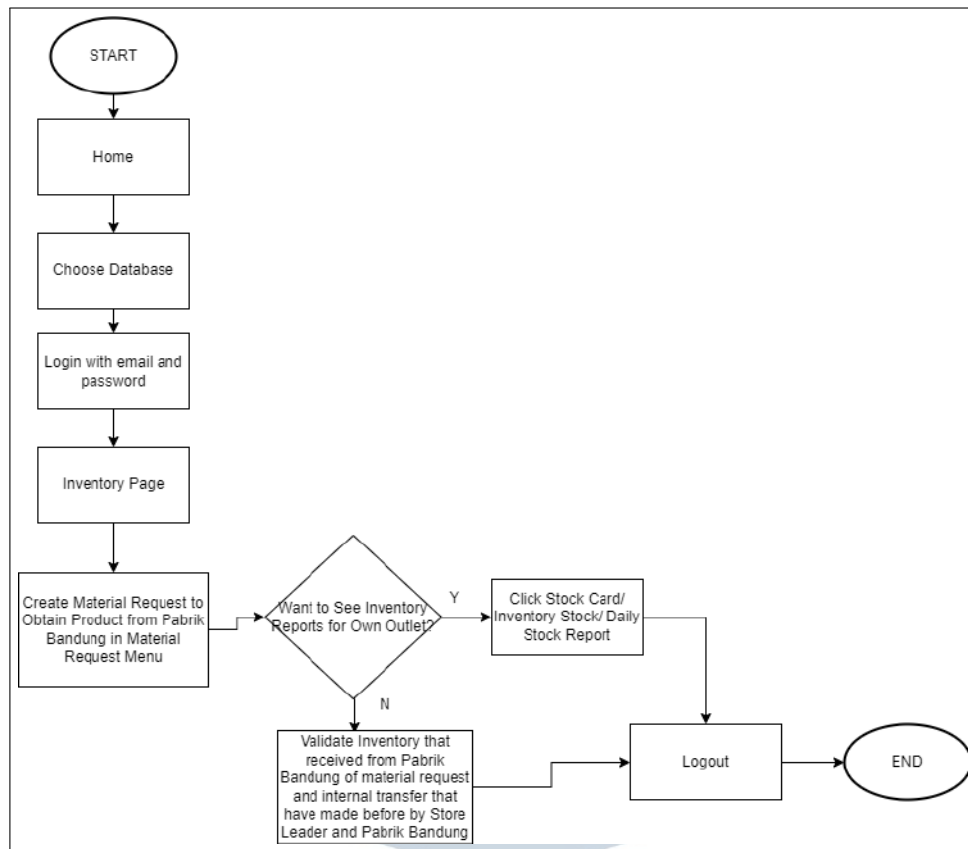
*terial request* yang dibuat oleh semua *store leader*, maka pengguna dapat membuka menu *material request*. Kemudian, jika pengguna ingin melihat / membuat *internal transfer*, maka pengguna dapat menekan menu *internal transfer*.

Kemudian, jika pengguna ingin menambah stok barang atau melihat penambahan stok barang semua outlet, maka pengguna dapat menekan tombol *inventory adjustment plus*. Kemudian, jika pengguna ingin melihat *report* pergerakan barang setiap outlet, maka pengguna memiliki beberapa pilihan untuk melihatnya. Pertama, pengguna dapat melihat *daily stock report* untuk melihat data stok barang di outlet yang diinginkan, lalu bisa diset per kategori barang.

Pengguna juga dapat mengubah tanggal untuk mengecek stok barang di tanggal yang dipilih. Kemudian, pengguna dapat melihat data pergerakan barang yang lebih lengkap pada *stock card report* dimana pada *report* ini, pengguna dapat melihat semua pergerakan barang mulai dari barang yang masuk hingga keluar. Sedangkan untuk *daily stock report*, hanya ditampilkan *stock balance* akhir saja.

Kemudian, pengguna juga dapat mengecek jumlah barang di semua outlet pada *inventory stock*. Di halaman ini, terpampang *available quantity stock* dari masing-masing outlet dan dipisah per produk. Apabila pengguna sudah mengecek semua dan ingin keluar dari sistem, maka pengguna dapat melakukan *logout*.

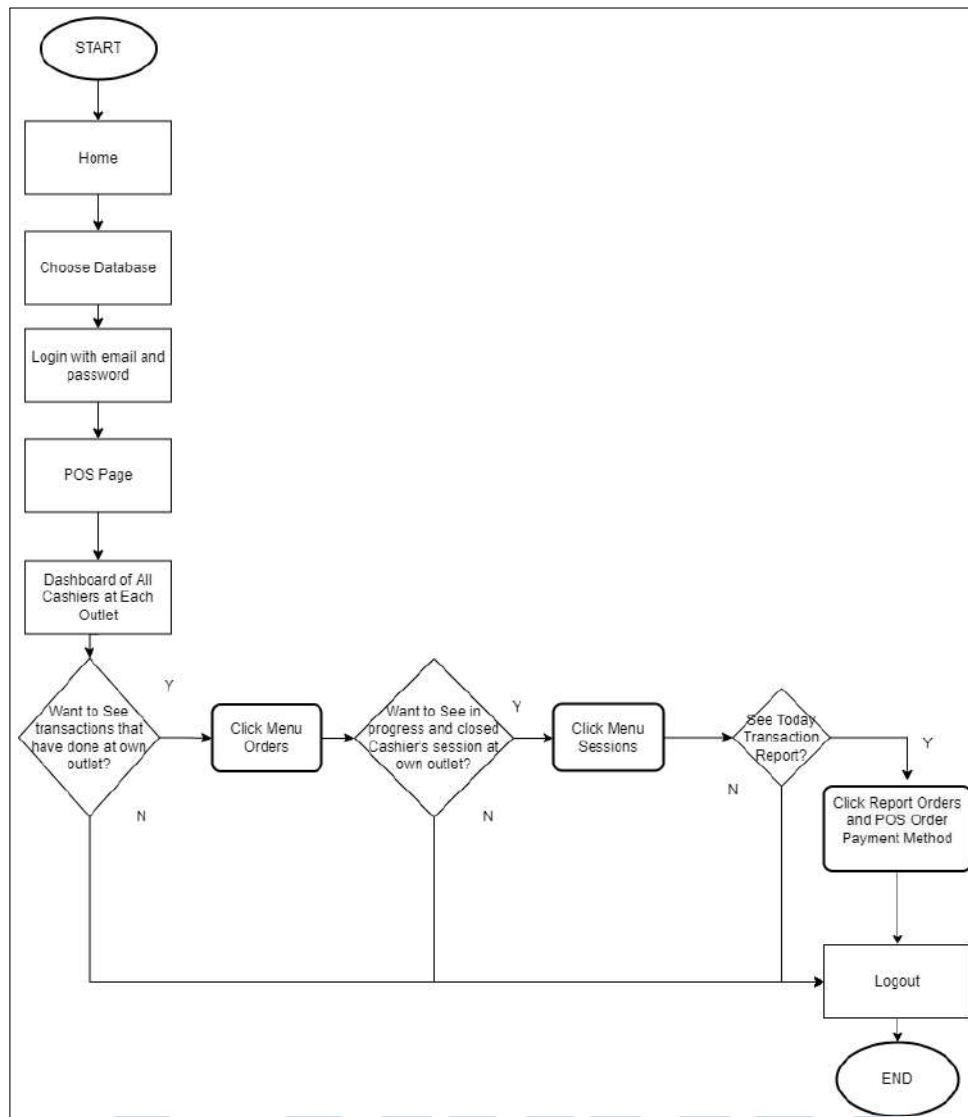




Gambar 3.3. Flowchart sistem sebagai *store leader* pada halaman *inventory*

Berdasarkan Gambar 3.3, pengguna pertama kali akan memilih *database* yang akan digunakan. Kemudian, pengguna sebagai *store leader* melakukan *input email* dan *password* yang telah didaftarkan oleh akun admin. Kemudian, pengguna sebagai *store leader* akan membuat *material request* untuk mendapatkan produk dari pabrik Bandung.

Kemudian, setelah Pabrik Bandung telah mengirimkan produk yang diminta, maka pengguna dapat melakukan *validate* jika barang sudah diperiksa dan sesuai jumlahnya yang dikirim oleh pabrik Bandung. Kemudian, untuk mengecek apakah barang sudah masuk ke outlet yang dihandle, pengguna dapat mengecek *report* pada *stock card*, *inventory stock*, ataupun *daily stock report*. Namun, apabila pengguna merasa tidak perlu memeriksa *report*, maka pengguna dapat langsung melakukan *logout*.



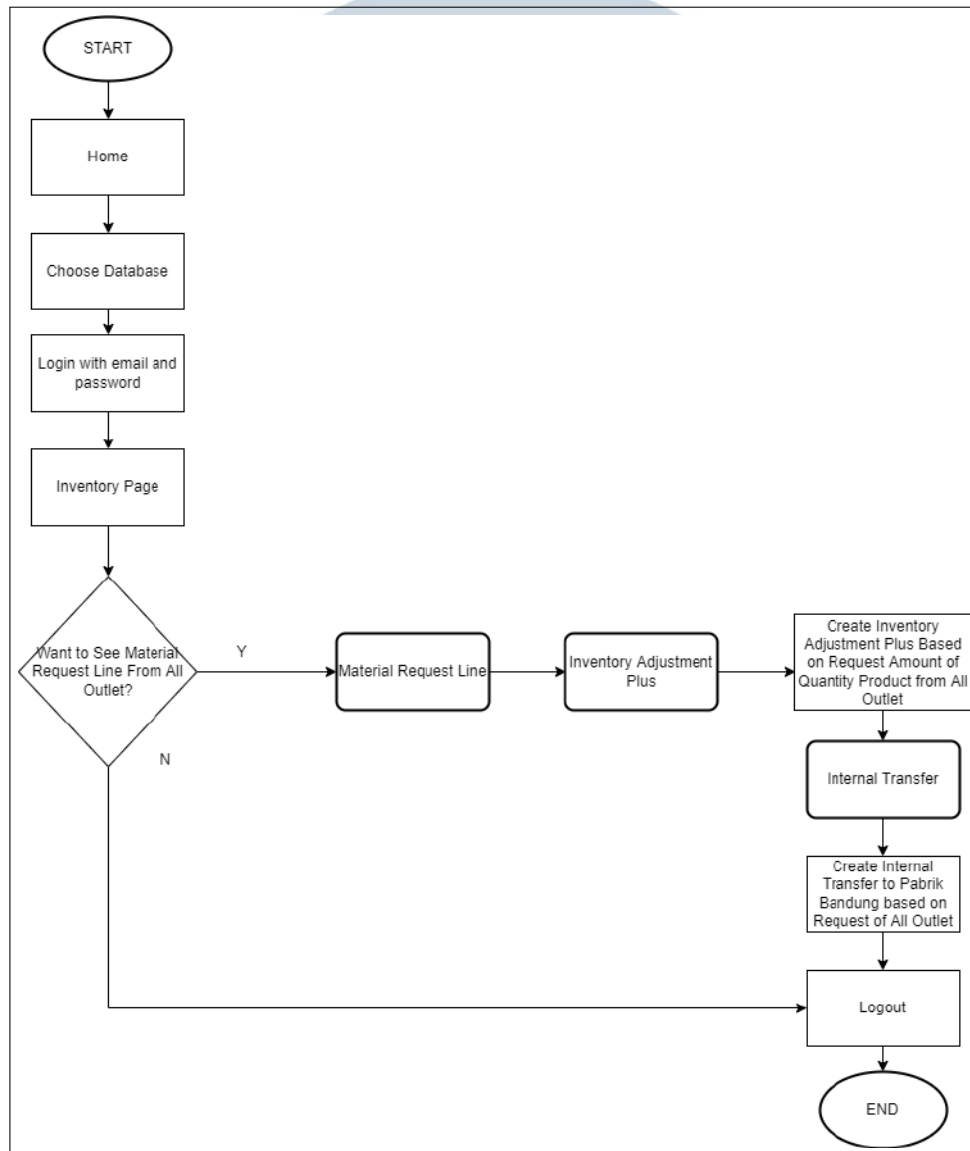
Gambar 3.4. Flowchart sistem sebagai *store leader* pada halaman *point of sale*

Berdasarkan Gambar 3.4, pengguna sebagai *store leader* akan memilih *database* yang akan digunakan. Kemudian, pengguna sebagai *store leader* melakukan *input email* dan *password* yang telah didaftarkan oleh akun admin. Pengguna akan ditampilkan dengan *dashboard point of sale*.

Jika ingin melihat transaksi yang berjalan, maka pengguna dapat melihat pada menu *orders*. Transaksi akan muncul sesuai dengan *assigned branch* masing-masing. Jika pengguna ingin melihat *session* yang aktif atau tutup, maka pengguna dapat masuk ke menu *sessions*.

Kemudian, pengguna dapat melihat *report* penjualan yang sedang berjalan, maka pengguna dapat menekan menu *report orders* dan *POS order payment*

*method.* Jika pengguna ingin keluar dari sistem, maka pengguna dapat menekan tombol *logout*.



Gambar 3.5. Flowchart sistem sebagai *product packaging*

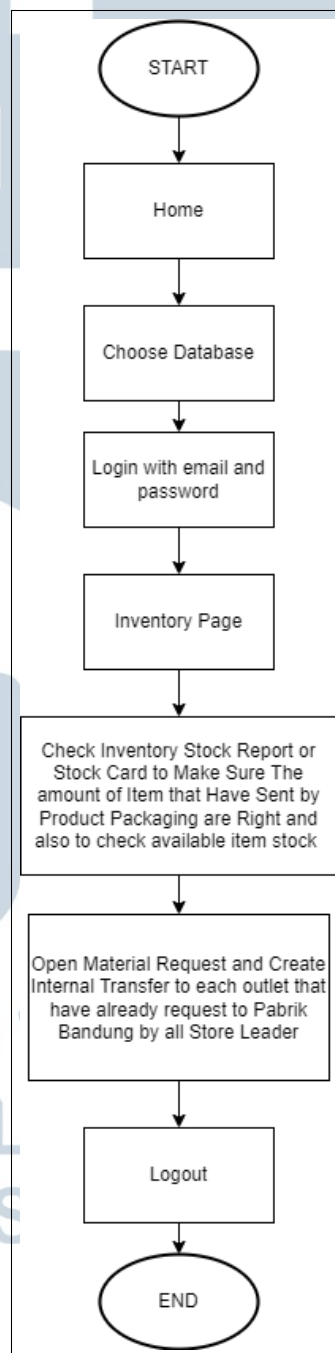
Berdasarkan Gambar 3.5, pengguna sebagai *product packaging* akan melakukan pemilihan *database* yang diinginkan. Kemudian, pengguna melakukan *input email* dan *password* yang telah didaftarkan oleh akun admin. Pengguna pertama kali akan ditampilkan dengan halaman *material request line*.

Pengguna akan melihat semua produk yang dilakukan *material request* pada halaman *material request line* yang disertai dengan berapa jumlahnya. Kemudian, setelah melihat *material request line*, maka pengguna akan melakukan *inventory ad-*



*justment plus* untuk menambahkan produk yang dibutuhkan untuk dikirim ke pabrik Bandung.

Kemudian, terakhir, pengguna akan melakukan *internal transfer* ke *pabrik bandung* untuk mengirimkan produk yang telah dipesan oleh setiap *store leader* dari masing-masing outlet. Kemudian, pengguna dapat keluar dari sistem dengan menekan tombol *logout*.

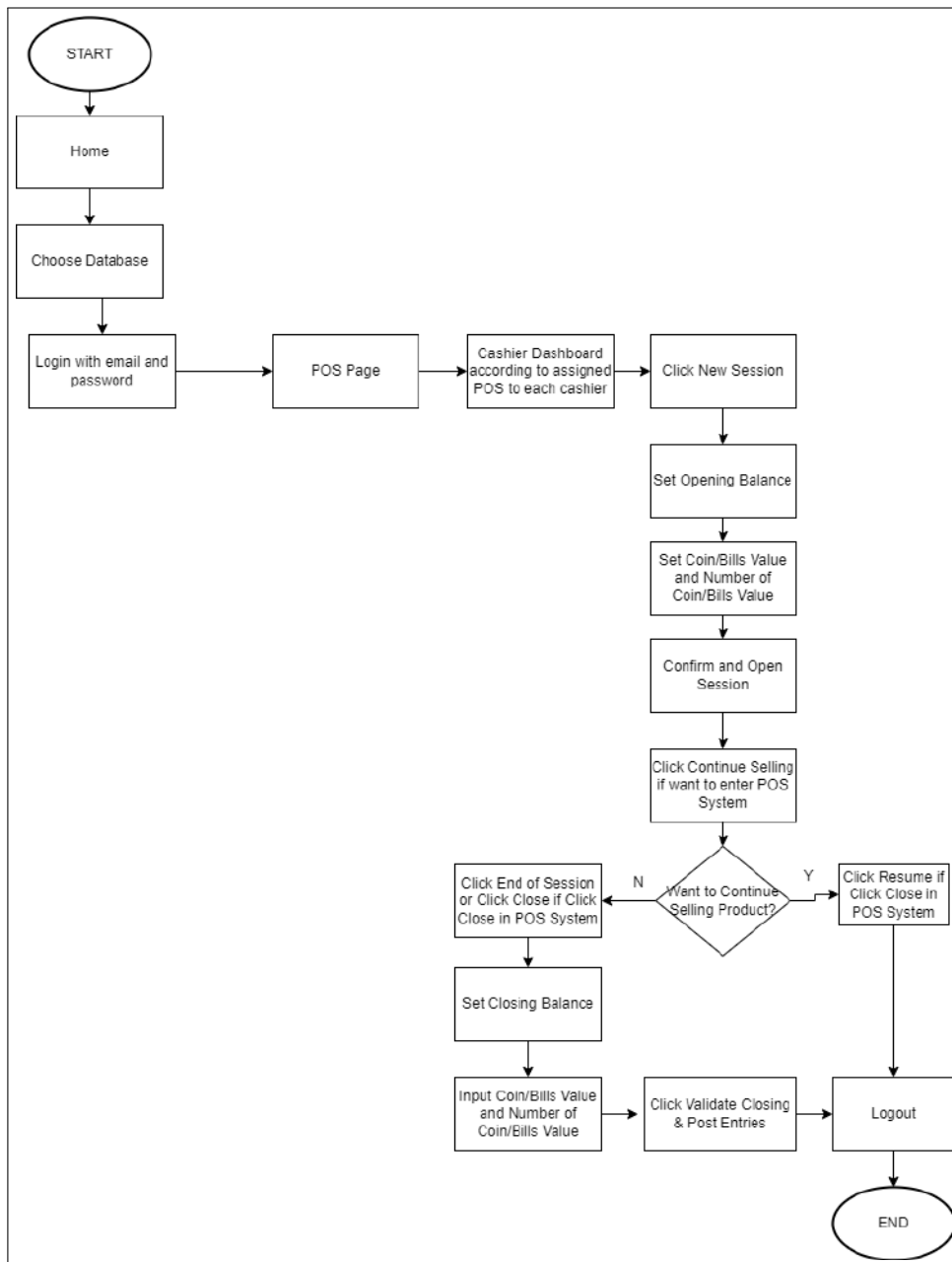


Gambar 3.6. Flowchart sistem sebagai *product distribution*

Berdasarkan Gambar 3.6, pengguna sebagai *product distribution* akan memilih *database* yang akan digunakan. Kemudian, pengguna akan memasukkan *email* dan *password* yang telah didaftarkan pada akun admin. Kemudian, pengguna mengecek *inventory stock* atau *stock card* untuk memastikan jumlah barang yang dikirimkan oleh *product packaging* sesuai jumlahnya dengan barang yang dilakukan *material request* oleh *store leader* dari masing-masing outlet.

Kemudian, pengguna akan membuka setiap *material request* yang masuk ke sistem pengguna sebagai *product distribution*. Kemudian, pengguna akan melakukan *internal transfer* sesuai dengan jumlah yang diminta oleh masing-masing *store leader*. Kemudian, pengguna dapat keluar dari sistem dengan melakukan *logout*.





Gambar 3.7. Flowchart sistem sebagai store cashier

Berdasarkan Gambar 3.7, pengguna memilih *database* yang akan digunakan. Kemudian, pengguna akan memasukkan email dan *password* sesuai dengan yang telah diregistrasi oleh akun admin. Kemudian, pengguna akan ditampilkan halaman *dashboard* kasir sesuai dengan yang diassign.

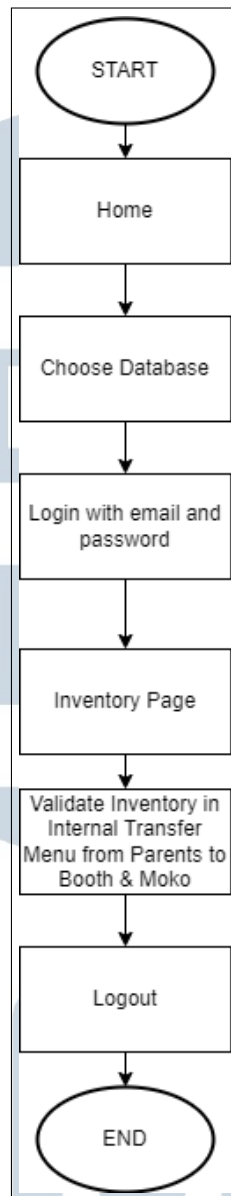
Kemudian, pengguna menekan tombol *new session* untuk membuka sesi baru. Lalu, pengguna wajib memasukkan *opening balance* untuk memulai sesi baru. Kemudian, pengguna memasukkan nilai *coin/bills*. Kemudian, pengguna

melakukan *confirm* dan *open session*.

Selanjutnya, menekan tombol *continue selling* jika ingin memasuki sistem *point of sale*. Kemudian, jika ingin melakukan penutupan *session*, maka pengguna dapat menekan tombol *end of session*. Jika pengguna terlanjur menekan tombol *close* pada sistem POS, maka pengguna dapat menekan tombol *resume* jika ingin melanjutkan penggunaan sistem *point of sale*.

Apabila pengguna benar ingin mengakhiri *session*, maka pengguna dapat menekan *close*. Kemudian, pengguna diwajibkan untuk memasukkan nilai *closing balance* dan memasukkan nilai *coin and bills*. Apabila pengguna sudah memasukkan nilai *closing balance* dengan benar, maka pengguna dapat langsung menekan tombol *validate closing and Post Entries*. Jika pengguna ingin keluar dari sistem, maka pengguna dapat melakukan *logout*.

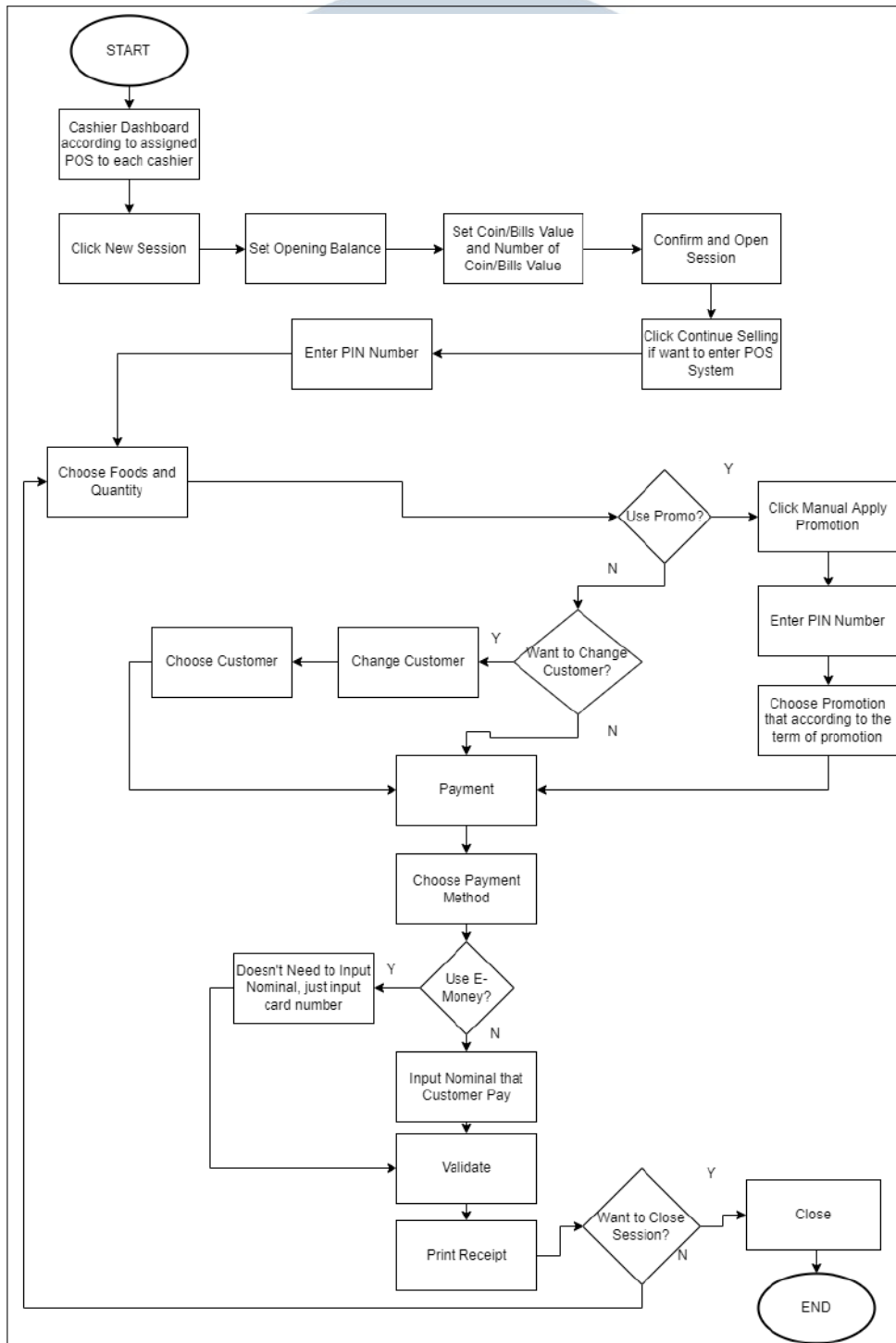




Gambar 3.8. Flowchart sistem sebagai *store cashier* booth dan moko

Berdasarkan Gambar 3.8, pengguna sebagai kasir booth moko, akan memilih *database* sesuai dengan yang digunakan. Kemudian, pengguna memasukkan *email* dan *password* sesuai dengan yang teregistrasi pada akun admin. Untuk sistem POS, sama dengan milik *store cashier*. Namun, terdapat perbedaan dimana booth moko memiliki modul *inventory* untuk melakukan validasi barang pada *internal transfer* yang dikirimkan oleh *parents* ke *child*.

## B. Flowchart Halaman Fitur



Gambar 3.9. Flowchart proses sistem POS

Berdasarkan Gambar 3.8, seperti langkah yang dilakukan oleh pengguna sebagai *store cashier*. Namun, disini akan membahas proses setelah pengguna sebagai *store cashier* atau booth dan moko mengklik *continue selling*. Setelah pengguna menekan tombol *continue selling*, maka pengguna akan diminta untuk memasukkan *PIN number*.

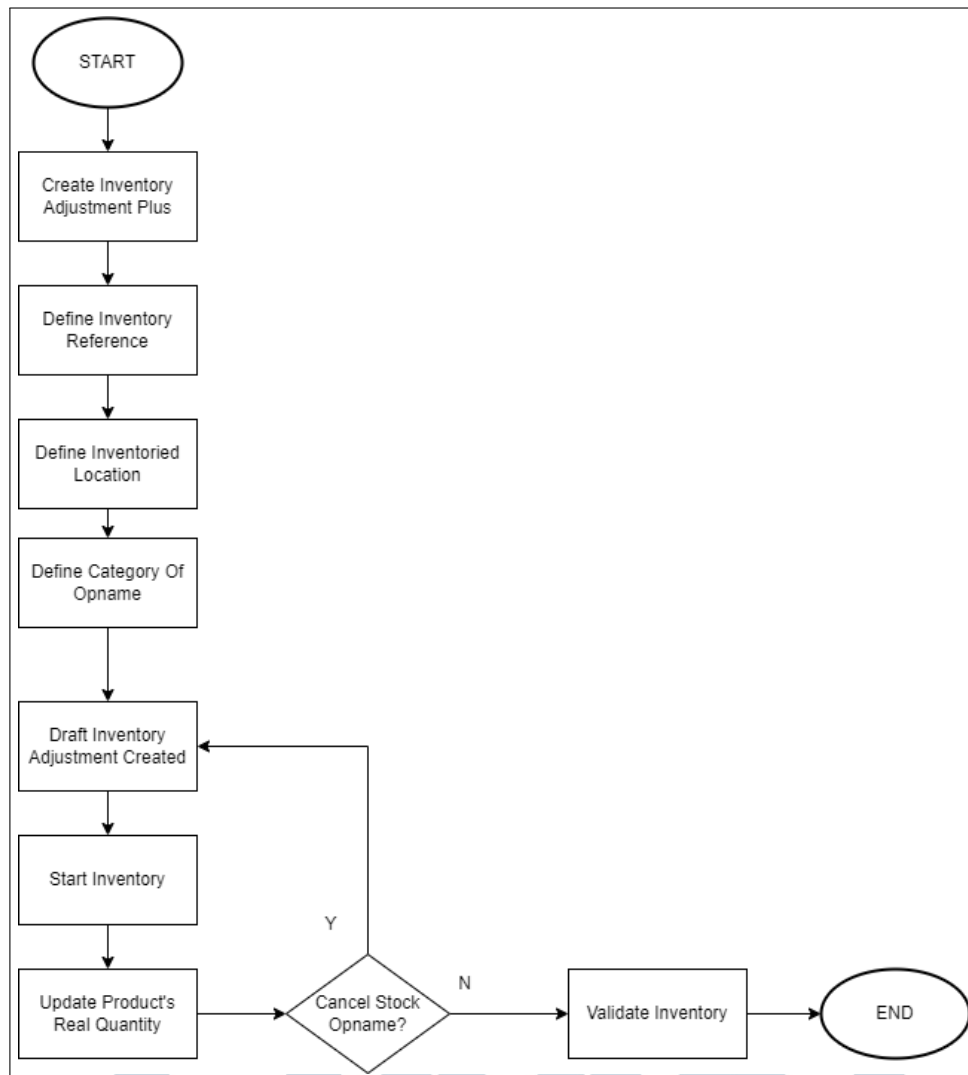
*PIN number* dibuat pada *setting users* yang dilakukan oleh akun admin. Setelah memasukkan *PIN number*, pengguna sebagai kasir dapat memilih produk yang ingin dibeli oleh pelanggan dan memasukkan jumlah produk sesuai dengan keinginan pelanggan. Kemudian, pengguna dapat memasukkan promosi dengan menekan tombol *manual apply promotion* apabila pembelian pelanggan memasuki syarat ketentuan promosi yang dibuat.

Jika pesanan dapat dimasukkan promosi, maka pengguna diminta untuk memasukkan *PIN number* lagi untuk memilih promosi yang dapat digunakan. Kemudian, pengguna dapat memilih *customer* lain selain *walk in customer* dengan menekan *customer* dan memilih *customer* selain *walk in customer*. Kemudian, pengguna menekan tombol *change customer*.

Selanjutnya, jika pengguna merasa semua yang dilakukan sebelumnya sudah benar, maka pengguna dapat melanjutkan ke pembayaran dengan menekan tombol *payment*. Kemudian, pengguna diminta untuk memilih metode pembayaran yang digunakan oleh pelanggan dalam melakukan pembelian. Jika pelanggan menggunakan *E-money*, maka pengguna hanya perlu memasukkan *card number* tanpa perlu memasukkan nominal pembayaran.

Sedangkan, jika pelanggan menggunakan pembayaran *cash*, maka pengguna harus memasukkan nominal pembayaran sesuai dengan total transaksi yang dilakukan atau sesuai dengan total yang dibayar oleh pelanggan. Kemudian, pengguna menekan tombol *validate* dan melakukan *print* struk pembelian.

Jika pengguna ingin mengakhiri *session*, maka pengguna dapat menekan *close* dan *confirm*. Namun, jika pengguna ingin meneruskan penjualan, maka pengguna akan diminta memilih produk lagi dan memasukkan jumlah barang seperti yang dilakukan sebelumnya.



Gambar 3.10. Flowchart proses halaman inventory adjustment plus

Berdasarkan Gambar 3.9, pengguna sebagai admin atau *product packaging* akan melakukan *create inventory adjustment plus* untuk menambahkan produk yang ada di pabrik. Kemudian, pengguna wajib untuk memasukkan *inventory reference* sebagai nama dari *inventory adjustment plus* dari outlet mana.

Kemudian, pengguna diminta untuk memasukkan *inventoried location* sebagai lokasi yang dituju untuk penambahan stok barang. Selanjutnya, pengguna diminta untuk memasukkan kategori opname pada *field inventory of*. Pada *field* tersebut, terdiri atas *all products* yang digunakan untuk menambahkan semua produk yang tersedia dalam sistem POS.

Kemudian, ada *one product category* yang digunakan untuk menambahkan satu produk kategori barang saja. Kemudian, ada *one product only* dimana ini di-

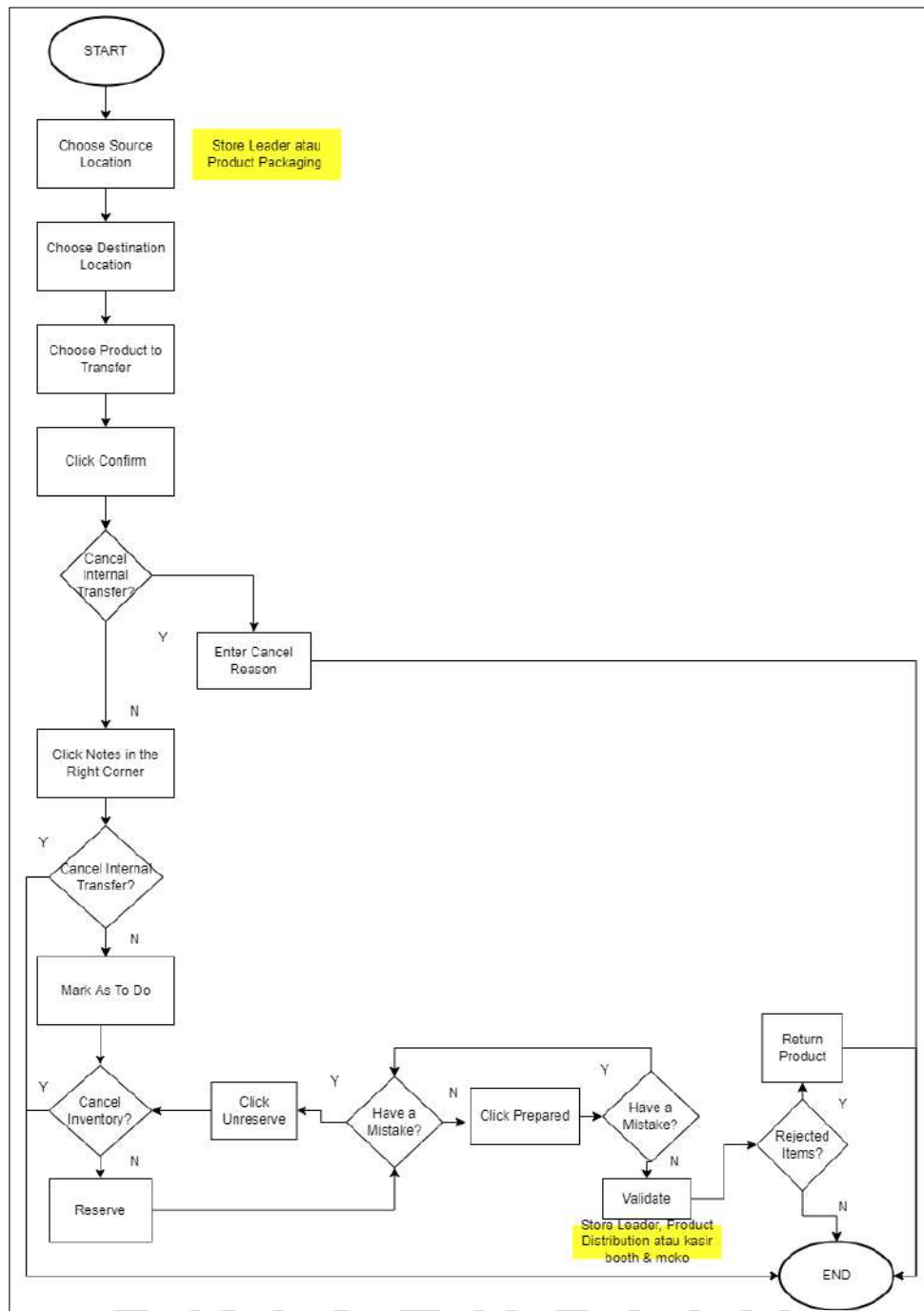


gunakan hanya untuk menambah satu jenis produk saja. Selanjutnya, ada *select product manually*, ini digunakan untuk memilih jenis barang secara manual oleh pengguna. Terakhir, ada *one lot/serial number*, ini digunakan untuk menambahkan dengan satu jenis *serial number* yang sama.

Kemudian, setelah *draft inventory adjustment* telah dibuat, maka pengguna dapat menekan tombol *start inventory* untuk meneruskan proses selanjutnya. Kemudian, pengguna dapat memasukkan jumlah barang dari masing-masing produk sesuai dengan permintaan dari *store leader* masing-masing outlet.

Apabila pengguna ingin melakukan *cancel*, maka pengguna akan dibalikkan lagi ke tahap *draft* kembali. Apabila jumlah dan jenis produk sudah benar, maka pengguna dapat menekan tombol *validate inventory* untuk menyelesaikan tahapan dari pembuatan *inventory adjustment plus*.





Gambar 3.11. Flowchart halaman internal transfer

Berdasarkan Gambar 3.10, pengguna yang ingin melakukan *internal transfer*, hal pertama yang dilakukan adalah memilih *source location* sebagai sumber lokasi barang yang akan dikirim. Kemudian, pengguna wajib untuk memasukkan *destination location* sebagai lokasi tujuan pengiriman barang. Kemudian, pengguna memilih produk apa saja yang ingin dilakukan pengiriman beserta dengan jumlah-

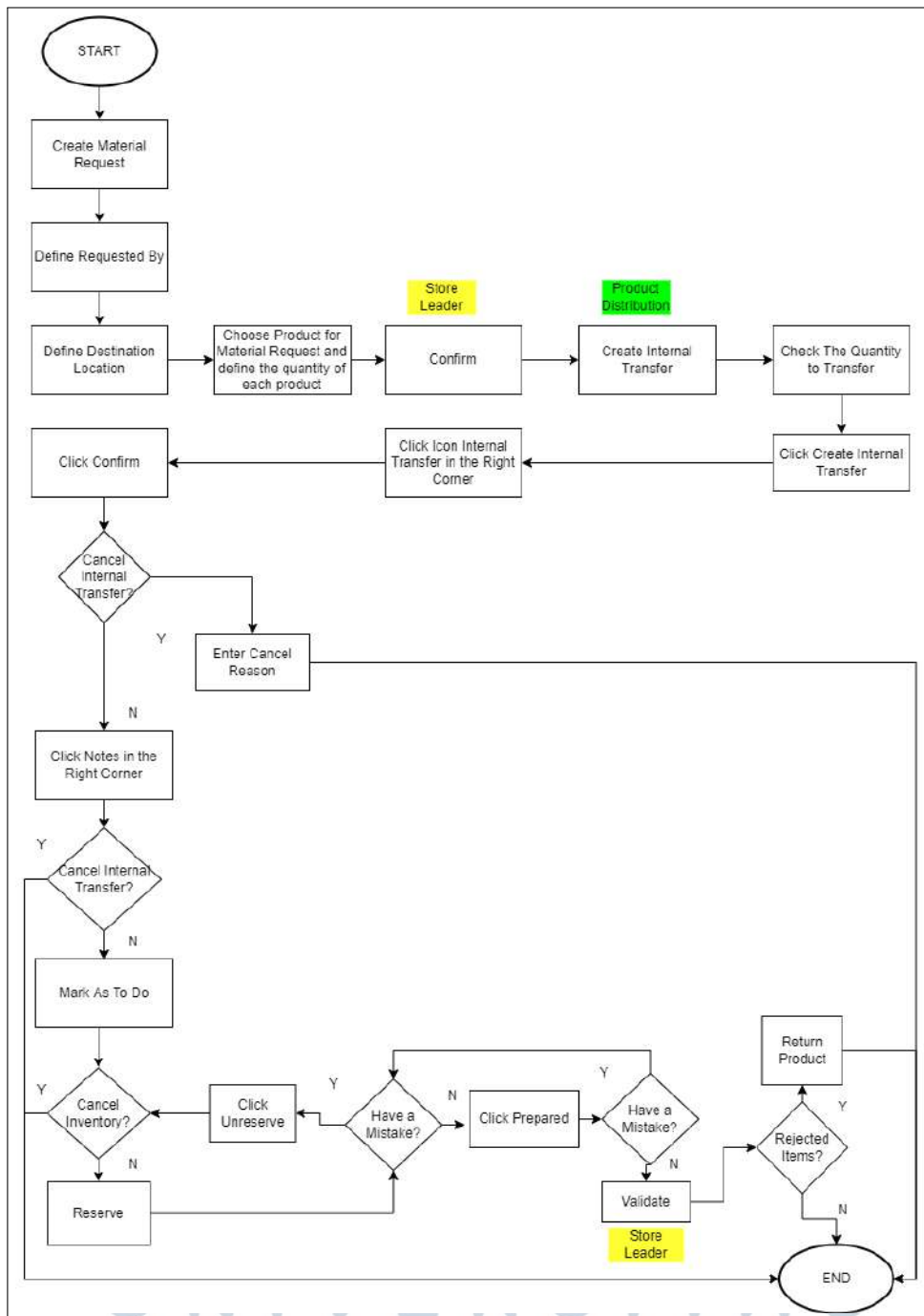
nya.

Lalu, jika sudah benar, maka pengguna dapat menekan tombol *confirm*. Apabila pengguna ingin melakukan pembatalan setelah dilakukan konfirmasi, maka pengguna akan diminta untuk memasukkan alasan pembatalan. Namun, jika pengguna merasa barang yang dikirim benar, maka pengguna dapat meneruskan tahapan selanjutnya dengan menekan tombol *notes* yang berbentuk seperti truk.

Kemudian, apabila pengguna merasa *internal transfer* yang dibuat ada kesalahan dan ingin dibatalkan maka pengguna harus membuat *internal transfer* baru. Namun, jika tidak ingin dibatalkan, maka pengguna dapat menekan tombol *mark as to do*. Selanjutnya, jika pengguna merasa produk yang akan dikirim sudah benar, maka pengguna dapat menekan tombol *reserve*.

Namun, apabila ada kesalahan dan tidak ingin meneruskan tahapan selanjutnya, maka pengguna dapat menekan tombol *cancel* dan harus membuat *internal transfer* baru. Namun, jika pengguna ingin meneruskan, maka pengguna dapat menekan tombol *reserve*. Jika merasa ada kesalahan, pengguna dapat menekan tombol *unreserve* dan memperbaiki kesalahan tersebut dan jika sudah benar bisa menekan tombol *reserve* lagi.

Kemudian, pengguna dapat menekan tombol *prepared* jika sudah benar. Namun, apabila ada kesalahan, pengguna bisa menekan tombol *unprepared* dan kembali ke tahapan *reserve* atau *unreserve*. Jika sudah, maka nanti pengguna yang menjadi *store leader* akan melakukan validasi dengan menekan tombol *validate*. Namun, jika ada kecacatan produk atau kesalahan dalam pengiriman, pengguna yang menjadi *store leader* dapat mengembalikan kiriman barang tersebut dengan cara menekan tombol *return product*.



Gambar 3.12. Flowchart halaman material request

Berdasarkan Gambar 3.11, ketika pengguna sebagai *store leader* ingin membuat *material request*, maka pengguna pertama kali menekan tombol *create* pada halaman *material request*. Kemudian, pengguna memasukkan nama yang mengajukan *material request*. Lalu, pengguna wajib untuk mengisi *destination location* sebagai lokasi yang dituju untuk pengiriman barang.

Kemudian, pengguna yang bersangkutan memasukkan produk apa saja yang disertai dengan jumlah yang diminta. Kemudian, pengguna sebagai *store leader* dapat menekan *confirm*. Kemudian, pada akun *product distribution* di halaman *material request* yang telah dibuat oleh *store leader* yang bersangkutan, pengguna dapat menekan tombol *create internal transfer*.

Kemudian, pengguna mengecek apakah barang yang ingin dilakukan *internal transfer* sesuai atau tidak. Kemudian, pengguna menekan tombol *create internal transfer*. Kemudian, setelah *internal transfer* dibuat, maka pengguna dapat menekan tombol *internal transfer* di pojok kanan atas.

Kemudian, pengguna dapat menekan tombol *confirm*. Apabila pengguna ingin melakukan pembatalan setelah dilakukan konfirmasi, maka pengguna akan diminta untuk memasukkan alasan pembatalan. Namun, jika pengguna merasa barang yang dikirim benar, maka pengguna dapat meneruskan tahapan selanjutnya dengan menekan tombol *notes* yang berbentuk seperti truk.

Kemudian, apabila pengguna merasa *internal transfer* yang dibuat ada kesalahan dan ingin dibatalkan maka pengguna harus membuat *internal transfer* baru. Namun, jika tidak ingin dibatalkan, maka pengguna dapat menekan tombol *mark as to do*. Selanjutnya, jika pengguna merasa produk yang akan dikirim sudah benar, maka pengguna dapat menekan tombol *reserve*.

Namun, apabila ada kesalahan dan tidak ingin meneruskan tahapan selanjutnya, maka pengguna dapat menekan tombol *cancel* dan harus membuat *internal transfer* baru. Namun, jika pengguna ingin meneruskan, maka pengguna dapat menekan tombol *reserve*. Jika merasa ada kesalahan, pengguna dapat menekan tombol *unreserve* dan memperbaiki kesalahan tersebut dan jika sudah benar bisa menekan tombol *reserve* lagi.

Kemudian, pengguna dapat menekan tombol *prepared* jika sudah benar. Namun, apabila ada kesalahan, pengguna bisa menekan tombol *unprepared* dan kembali ke tahapan *reserve* atau *unreserve*. Jika sudah, maka nanti pengguna yang menjadi *store leader* akan melakukan validasi dengan menekan tombol *validate*. Namun, jika ada kecacatan produk atau kesalahan dalam pengiriman, pengguna yang menjadi *store leader* dapat mengembalikan kiriman barang tersebut dengan cara menekan tombol *return product*.

### 3.4 Kendala dan Solusi yang Ditemukan

Kendala yang ditemukan selama melakukan kerja magang adalah sebagai berikut.

1. Pengembangan fitur yang digunakan adalah bahasa Odoo, Python, dan Javascript. Bahasa pemrograman Odoo belum pernah dipelajari selama di perkuliahan dan Python sendiri di perkuliahan belum mempelajari terlalu dalam, sehingga solusi yang dapat dilakukan adalah harus lebih banyak membaca dokumentasi serta bertanya kepada mentor maupun kepada teman yang pernah mempelajarinya dan juga mempelajarinya melalui internet.
2. Membutuhkan waktu yang cukup lama dalam mempelajari alur proses bisnis ERP (*Enterprise Resource Planning*) dikarenakan dalam perkuliahan tidak diajarkan dalam keadaan *real* sehingga solusi yang dapat dilakukan adalah mengikuti *training* yang diadakan oleh perusahaan dan juga menonton *youtube* untuk mengetahui secara jelas alur proses bisnis sistem ERP (*Enterprise Resource Planning*).
3. Terlalu banyak permintaan dari klien yang dikejar oleh *timeline* yang menyebabkan muncul banyak *issue* yang tidak tertebak sebelumnya sehingga solusi yang dapat ditawarkan adalah dengan menganalisa secara baik untuk melakukan *reproduce* suatu masalah secara mendetail supaya tidak muncul *issue* atau masalah yang dapat menyebabkan mundurnya *timeline* yang telah dibuat.

UIMN  
UNIVERSITAS  
MULTIMEDIA  
NUSANTARA