

## BAB III PELAKSANAAN KERJA MAGANG

### 3.1. Kedudukan dalam Kerja Magang

Pada kerja magang ini masuk kedalam departemen *Operation divisi project team* dengan posisi sebagai *ERP Implementation Intern*, dibawah naungan supervisor lapangan yaitu Sinta Kartika dengan jabatan sebagai *ERP Consultant* dan *Head of Operation* Belatrix Wulandari. Dalam kerja magang ini mahasiswa masuk kedalam divisi *operation* bagian *project* yang beranggotakan:

**Tabel 3. 1 Tabel Anggota Team Project**

<b>Role</b>	<b>Banyak anggota</b>
<i>Project coordinator</i>	1
<i>Business analyst</i>	1
<i>Project manager</i>	1
<i>ERP Implementation Intern</i>	2
<i>Developer team India</i>	3
<i>Developer team Vietnam</i>	2
<i>Developer team Indonesia</i>	2
<i>Technical Writer</i>	2

Pada Minggu awal masuk kedalam *project team* mahasiswa disambut oleh *Project Coordinator* dan *Project manager* terkait peran yang akan dilakukan selama masa praktek kerja magang, setelah mengetahui peran yang akan dilakukan, mulai dimasukkan kedalam project yang akan diimplementasikan dalam hal ini implementasi sistem *Entrprise Resource Planning* pada PT XYZ untuk modul *Point of Sales(POS)* dan modul *Inventory*. Setelah mengetahui project apa yang akan di *assign* mahasiswa perlu memahami detail sistem dan proses bisnis seperti apa yang akan di implementasikan, maka seringkali *project manager* yang bertanggung jawab atas *project* ini melakukan *transfer knowledge* agar mahasiswa

dapat memahami modul *Point of Sales(POS)* dan modul *Inventory* yang akan di implementasikan serta proses bisnis PT XYZ.

Dalam project sistem ERP yang akan di implementasikan pada PT XYZ ini sudah melewati beberapa fase, yakni:

a. *Kick-off-meeting*

Pada fase kick off meeting ini, Tim Project perlu mempelajari SBD (*Sales Briefing Document*) yang isinya merupakan pemaparan modul dan fitur apa saja yang dibutuhkan oleh *client* yang dibuat oleh tim *Business Development*. Selama jalannya *kick-off-meeting* ini Tim *Project* perlu memperkenalkan tim yang akan menangani project ini, lalu memberi perkenalan tentang fase *development*, menjelaskan konsep FRD(*Functional Requirement Document*), menjelaskan pemahaman terkait permohonan perubahan, meminta *client* untuk dapat menjelaskan proses bisnis mereka secara umum sesuai dengan modul yang dibeli, serta mengatur untuk *meeting* selanjutnya untuk mengetahui secara lebih detail dan terperinci terkait kebutuhan *client*.

b. *Requirement Gathering*

Sebelum memulai fase ini *client* dipastikan sudah mendapatkan link demo terkait modul yang dibeli, agar pada fase ini Tim *project* dapat mengumpulkan dan mencatat kebutuhan *client* pada tiap modul yang dibeli, mulai dari gambaran umum hingga lebih rinci. Setelahnya Tim *project* menentukan anggaran *project* berdasarkan kebutuhan dari *client*, lalu membuat Tim *Project* membuat *Functional Requirement Document* dan *timeline* dengan melakukan estimasi jumlah kustomisasi yang diminta dan *workload* yang akan dihadapi tim *project*. Lalu, FRD tersebut perlu di presentasikan dan di tinjau oleh komite proyek dan *Business*

*analyst*, jika sudah di setujui oleh mereka maka akan ada penandatanganan dokumen FRD.

c. *Development Phase*

Pada fase *development phase* tim *project* meminta *developer team* yang akan menangani *project* ini kepada *project coordinator*. Lalu tim *project* membuat *developer tasklist* sesuai dengan dokumen FRD yang telah disepakati bersama, melakukan pembagian *developer tasklist* berdasarkan metodologinya *Waterfall*, *Agile* atau *Scrum*, melakukan testing terhadap tugas dari *developer*, melakukan penginputan ETA *developer* pada portal internal, memastikan *timeline* berjalan sesuai dengan kesepakatan diawal, menindaklanjuti berbagai hal terkait manajemen proyek, serta melakukan *test* secara menyeluruh untuk dapat menemukan *bugs*, *defact*, atau *error* pada sistem yang sedang di *develop*.

d. *Reguler Meeting*

*Reguler Meeting* merupakan fase yang dilakukan setiap minggunya oleh tim *project* agar dapat memaparkan kepada *client* terkait *progress project* yang sudah dikerjakan oleh tim *project* pada tiap minggu, berguna agar *client* dapat melakukan *tracking* terhadap *timeline* agar tetap dapat sesuai dengan target diawal.

e. *User Acceptance Testability (UAT)*

Pada fase *User Acceptance Testability* tim *project* bersama *client* melakukan *testing* sistem *Enterprise Resource Planning* modul *Point of Sales (POS)* dan *Inventory* yang telah di *develop* sesuai dengan FRD, lalu memastikan master data *client* telah masuk kedalam sistem. Setelah semua *case* telah dicoba dan disetujui oleh *client* maka akan masuk ketahap *traning* dan *Go-live*, namun jika masih ada ketidak sesuaiaan maka *client* dapat memberikan umpan balik kepada tim

*project* untuk dilakukan *bugs fixing* terhadap fitur yang dirasa masih perlu dilakukan perbaikan.

f. *Bugs Fixing*

Pada fase *bugs fixing* tim *project* merakipitulasi umpan balik yang disampaikan *client* saat proses *User Acceptance Testability (UAT)* untuk dapat diperbaiki oleh developer, maka dari itu tim *project* akan membuat *developer tasklist* berdasarkan umpan balik, membuat pembagaian metodologi yang digunakan antara *Waterfall*, *Agile* atau *Scrum* sesuai dengan urutan *depedency*nya dan di assign kepada developer team terkait untuk dapat diperbaiki dan diselesaikan, jika tim *Project* sudah merasa sesuai maka akan melanjutkan input ETA developer ke portal internal, apabila masih belum sesuai meminta *developer* untuk memperbaikinya lagi.

g. *Training*

Pada fase *training* tim *project* melatih dan memberitahukan *client* tentang cara penggunaan fitur-fitur yang ada didalam modul *Point of Sales* dan *Inventory*, yang sesuai dengan proses bisnis PT XYZ, dan memeriksa daftar kehadiran terkait siapa saja yang mengikuti *training* tersebut sebagai bukti kepada manajemen bahwa *training* telah dilaksanakan.

h. *Go-Live*

Pada fase *Go-Live* ini PT XYZ sudah menyetujui bahwa semua fitur sudah ter *deliver* dengan baik dan tim *project* akan menjadwlakan untuk melakukan pendampingan secara *onsite* untuk memastikan bahwa sistem yang di implementasikan berjalan dengan lancar, lalu nanti setelah 4 minggu pertama setelah melakukan *Go-live* maka akan melakukan serah terima *project* dari divisi *project* ke divisi *support*.

Pada praktek kerja magang ini mahasiswa masuk kedalam fase *Development Phase, Regular meeting, User Acceptance Testability (UAT), Bugs Fixing, Training, dan Go-live*. Selain itu, setelah mahasiswa mulai memahami akan modul *Point of Sales(POS)* dan modul *Inventory* yang akan di implementasikan serta proses bisnis PT XYZ, *Project Manager* mengenalkan mahasiswa kepada *developer team* yang menangani project ini, dalam hal ini terdapat *developer team* India, *developer team* Indonesia dan *developer team* Vietnam untuk dapat saling kenal dan dapat mengetahui nantinya akan *assign developer tasklist* ke *developer team* yang mana karena setiap dari *developer team* memiliki kemampuan yang berbeda-beda ada yang lebih ahli pada modul *Point of Sales(POS)* dan ada yang lebih ahli pada modul *Inventory*. *Developer tasklist* merupakan *list* tugas yang dibuat oleh mahasiswa untuk mengerjakan fitur yang akan dikustomisasi atau diperbaiki pada modul *Point of Sales(POS)* dan modul *Inventory* yang di dalamnya terdapat urutan prioritas pengerjaan yang terbagi kedalam 3 metodologi yakni *Waterfall, Agile* dan *Scrum* tergantung dari tingkat *dependency* dari tiap *task* dan juga tingkat priotitasnya.

Metodologi *Waterfall* merupakan metodologi klasik yang memiliki sifat sistematis, dimana perlu melakukan tugas secara berurutan dalam membangun software. Metodologi ini merupakan dilakukan secara satu arah yang dimulai dari tahap persiapan hingga perawatan. Metodologi *waterfall* juga merupakan metodologi yang sering digunakan secara umum. Inti dari penggunaan metode *waterfall* ini adalah dimana pengerjaan dari suatu tugas perlu dilakukan secara berurutan atau secara linear. Jadi jika langkah yang pertama belum dikerjakan, maka langkah kedua pun tidak dapat dikerjakan. Jika langkah kedua belum dikerjakan maka langkah ketiga juga tidak dapat dikerjakan, begitu seterusnya. Secara otomatis langkah ketiga akan dapat dilakukan ketika langkah pertama dan kedua telah dilakukan [3].

Metodologi Agile merupakan metodologi pengembangan perangkat lunak yang didasarkan pada pengembangan yang dilakukan secara iteratif, dimana *requirement* dan *solution* muncul melalui kolaborasi yang dilakukan oleh antar tim,

bisa juga dibidang metodologi agile cocok pada *development* dengan jangka waktu yang singkat dengan tingkat adaptabilitas yang cepat terhadap perubahan yang terjadi [5], menurut *agile manifesto* agile *development* memiliki tujuan, yakni:

1. *High-value & working App system*
2. *Iterative, incremental, evolutionary*
3. *Cost control & value-driven development*
4. *High-quality production*
5. *Flexible & risk management*
6. *Collaboration*
7. *Self-organizing, self-managing teams*

Metodologi Scrum merupakan metodologi turunan dari agile development yang memerlukan seorang *scrum master* untuk dapat membagi tugas kedalam beberapa tingkat prioritas dan tugas mana saja yang dapat dilakukan secara paralel, metodologi ini sendiri perlu melakukan *sprint meeting* di awal minggu dan review di akhir minggu untuk mengetahui tugas mana saja yang sudah selesai atau belum, kendala yang ditemukan serta berbagai hal yang menyangkut tugas agar tugas dapat diselesaikan dengan baik oleh *developer* dan tim *project* memiliki kontrol atas tugas yang sedang dikerjakan.

Lalu, karena dinilai oleh project manager bahwa mahasiswa sudah memiliki kapabilitas dan pengetahuan yang baik akan modul *Point of Sales (POS)* dan modul *Inventory* yang akan di implementasikan dan sudah memahami alur proses bisnis PT XYZ dengan baik, maka mahasiswa melakukan presentasi kepada *client* terkait *feature* yang telah dikembangkan atau diperbaiki agar *client* dapat mengetahui sudah sejauh mana *progress* dari sistem yang akan di implementasikan.

U N I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A

## 3.2. Tugas dan Uraian dalam Kerja Magang

### 3.2.1. Tugas dalam Kerja Magang

Kegiatan kerja magang berfokus pada manajemen proyek *ERP* yang diimplementasikan PT Hashmicro Solusi Indonesia pada PT. XYZ untuk modul *POS(Point of Sale)* dan *Inventory* pada tiap fase implementasi yakni *Development Phase, Regular meeting, User Acceptance Testability (UAT), Bugs Fixing, dan Training*. Kegiatan manajemen proyek pada tiap fase ini meliputi membuat, memperbarui, dan memelihara berbagai tindakan terkait manajemen proyek seperti membuat *developer tasklist* yang akan dibagi sesuai metodologinya antara *Waterfall, Agile* atau *Scrum* serta merencanakan untuk dibagi ke *developer team* yang memiliki kemampuan yang baik pada *task* yang akan diberikan, melakukan perorganisasian dan presentasi kepada client saat *regular meeting*, melakukan kontrol pengerjaan *developer tasklist* agar tidak melibihi *timeline* yang telah ditetapkan diawal, serta memastikan kepuasan *client* atas kinerja tim *project*, lalu Melacak status proyek, perubahan proyek, atau urutan tugas dalam proyek serta menetapkan, memantau, mengomunikasikan, serta memelihara jadwal proyek.

Selain dari melakukan manajemen implementasi proyek ERP pada PT XYZ mahasiswa juga mendapatkan bekal *training* dari perusahaan untuk modul *Inventory, Accounting, Purchasing, Sales, Leads dan Manufacturing* agar mahasiswa memiliki pemahaman yang baik akan banyak modul sehingga saat bertemu langsung dengan *client* dapat menjawab pertanyaan dengan baik, serta bertujuan untuk dapat melancarkan proses implementasi sistem ERP karena mahasiswa memiliki *knowledge* yang baik atas *product* perusahaan . Di bawah ini merupakan tabel pekerjaan yang dilakukan saat melaksanakan praktek kerja magang:

Tabel 3.2.1 Pekerjaan yang dilakukan

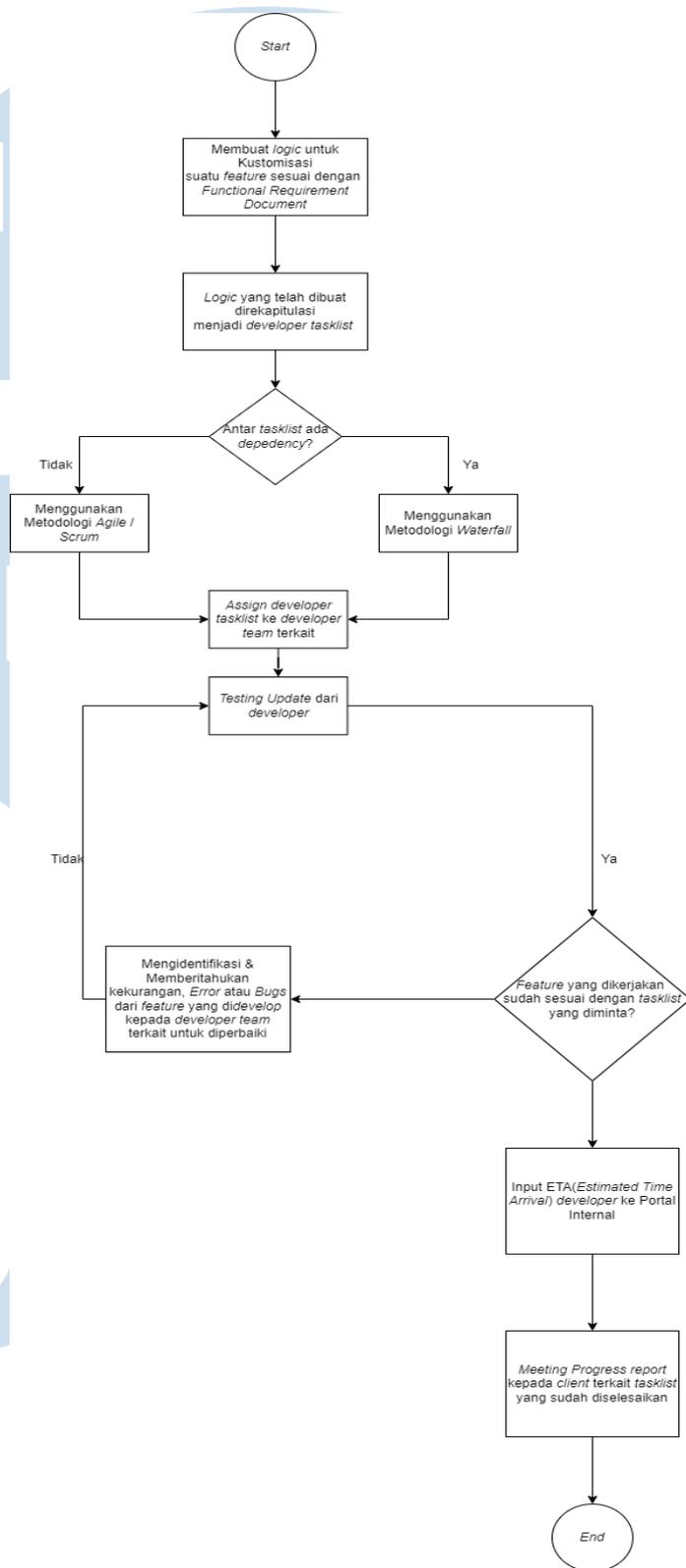
No	Pekerjaan yang dilakukan	Minggu	Tanggal mulai	Tanggal selesai
1	Manajemen <i>Development phase project</i> untuk modul POS ( <i>Point of Sale</i> ) dan <i>Inventory</i>	1-6	9 Agustus 2021	17 September 2021
2	<i>Training</i> modul <i>Inventory, Accounting, Purchasing, Sales, Leads dan Manufacturing</i>	4-6	30 Agustus 2021	17 September 2021
3	<i>Meeting Progress Report project</i> kepada <i>client</i>	1,2,3,6,10, 13,14,15,19	9 Agustus 2021	17 Desember 2021
4	UAT ( <i>User Acceptance Testability</i> ) ke-1 untuk modul POS ( <i>Point of Sale</i> ) dan <i>Inventory</i>	7	20 September 2021	24 September 2021
5	<i>Bug Fixing</i> setelah UAT ( <i>User Acceptance Testability</i> ) ke-1 untuk modul POS ( <i>Point of Sale</i> ) dan <i>Inventory</i>	8-15	27 September 2021	19 November 2021
6	UAT ( <i>User Acceptance Testability</i> ) ke-2 dan <i>Training</i> untuk modul POS ( <i>Point of Sale</i> ) dan <i>Inventory</i>	16	22 November 2021	26 November 2021
7	<i>Bug Fixing</i> setelah UAT ( <i>User Acceptance Testability</i> ) ke-2 untuk modul POS ( <i>Point of Sale</i> ) dan <i>Inventory</i>	17-19	29 November 2021	17 Desember 2021

### 3.2.2. Uraian dalam Kerja Magang

#### 3.2.2.1. Manajemen *Development phase project* modul POS (*Point of Sale*) dan *Inventory* (Minggu 1-6)

Pada fase ini mahasiswa melakukan manajemen terkait *development feature* yang dikustomisasi pada modul POS (*Point of Sale*) dan *Inventory* sesuai dengan kesepakatan antara client dan PT Hashmicro Solusi Indonesia yang tertuang di dalam *Functional Requirement Document*. Fase ini dilakukan pada minggu ke-1 hingga ke-6, pada minggu awal memasuki project ini mahasiswa diberi arahan oleh *project manager* terkait fungsionalitas *feature* pada Modul POS (*Point of Sale*) dan *Inventory* serta *feature* mana saja yang perlu di kustomisasi, selain itu pada fase *development* ini juga *project coordinator* akan memberitahukan *developer team* mana saja yang akan bertugas pada *project* ini, lalu *developer team* yang berasal dari India, Indonesia dan Vietnam yang diberikan untuk dapat mengerjakan *task* pada *project* ini. Setelah sudah mengetahui *Developer team* mana saja mahasiswa membuat *developer tasklist* sesuai dengan dokumen FRD yang telah disepakati bersama, lalu melakukan pembagian *developer tasklist* berdasarkan metodologinya *Waterfall*, *Agile* atau *Scrum*, melakukan *testing* terhadap tugas dari *developer*, melakukan penginputan ETA *developer* pada portal internal, serta melakukan test secara menyeluruh untuk dapat menemukan *bugs*, *defact*, atau *error* pada sistem yang sedang di *develop*. Pada gambar 3.2.2.1 dibawah merupakan *flow chart* rangkaian kegiatan yang dilakukan pada *development phase*.

U N I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A



Gambar 3.2.2.1 Flowchart Rangkaian Kegiatan *Development Phase*

Pada gambar 3.2.2.1 menjelaskan rangkaian kegiatan yang dilakukan pada fase *development* modul POS dan *Inventory* yang dilakukan mahasiswa, yakni :

- a. Membuat *logic* untuk Kustomisasi *feature* Modul POS dan *Inventory* sesuai dengan *Functional Requirement Document*

Di tahap ini mahasiswa membuat logika yang mudah dipahami oleh *developer* untuk dapat mengerti *feature* yang perlu di *bugs fixing* sesuai dengan *feedback* yang diberikan *client* saat UAT

- b. *Logic* yang telah dibuat direkapitulasi menjadi *developer tasklist*

*Developer team* terkait, berikut adalah *list* tugas *developer tasklist* pada *development phase*:

1. *Design Print Receipt Layout for Promotion Buy Product Free Product (Buy 3 get 1), make the free product on the last column of name, qty, but price and value =Rp.0 . Under the subtotal only ppn when this promo applied (Modul POS)*
2. *Design Print layout for compliment product, the total was right. Picture receipt down here was a compliment product, make the price and value Rp.0 and the subtotal only for non-compliment product. Under the subtotal only ppn when this applied. (Modul POS)*
3. *Design Print Receipt Layout for paket harga coret, disc 10% name it Discount in receipt layout. (Modul POS)*
4. *Design Print Receipt Layout for Product Package. Make the item that on that package display in the receipt, qty depends on settings that user input for (Modul POS)*

5. *Make the item that on the Product package display in the report order (Modul POS)*

6. *Promotion using price fixed amount promotion condition (Modul POS)*

7. *Promotion buy product free free product promotion condition (Modul POS)*

8. *Stock aging Report bugs custom layout (Modul Inventory)*

c. *Decision Making untuk penggunaan metodologi waterfall atau agile/scrum dengan faktor pembanding dependency antara task list.*

1. *Design Print Receipt Layout for Promotion Buy Product Free Product (Buy 3 get 1), make the free product on the last column of name, qty, but price and value =Rp.0 . Under the subtotal only ppn when this promo applied (Modul POS) terdapat dependency untuk task list lainnya maka menggunakan metodologi waterfall*

2. *Design Print layout for compliment product, the total was right. Picture receipt down here was a compliment product, make the price and value Rp.0 and the subtotal only for non-compliment product. Under the subtotal only ppn when this applied. (Modul POS) terdapat dependency untuk task list lainnya maka menggunakan metodologi waterfall*

3. *Design Print Receipt Layout for paket harga coret, disc 10% name it Discount in receipt layout. (Modul POS) terdapat dependency untuk task list lainnya maka menggunakan metodologi waterfall*

4. *Design Print Receipt Layout for Product Package. Make the item that on that package display in the*

- receipt, qty depends on settings that user input for (Modul POS) terdapat dependency untuk task list lainnya maka menggunakan metodologi waterfall*
5. *Make the item that on the Product package display in the report order (Modul POS) terdapat dependency untuk task list lainnya maka menggunakan metodologi waterfall*
  6. *Promotion using price fixed amount promotion condition (Modul POS) terdapat dependency untuk task list lainnya maka menggunakan metodologi waterfall*
  7. *Promotion buy product free free product promotion condition (Modul POS) terdapat dependency untuk task list lainnya maka menggunakan metodologi waterfall*
  8. *Stock aging Report bugs custom layout (Modul Inventory) tidak terdapat dependency maka dapat menggunakan metodologi agile (paralel dengan task lainnya)*

d. *Assign developer tasklist ke developer team terkait*

*Meng-assign tasklist no 1-7 kepada developer team vietnam, dan tasklist no 8 assign ke developer team Indonesia.*

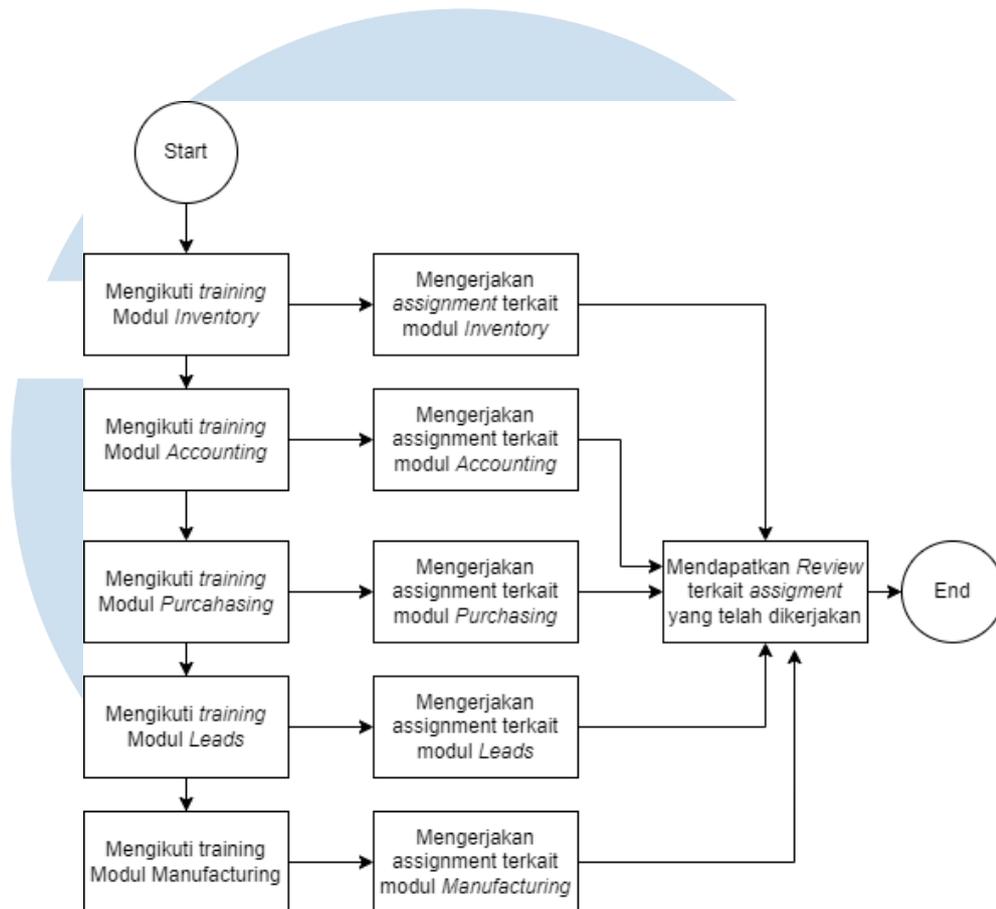
e. *Testing update dari developer*

*Melakukan pengecekan terhadap pekerjaan yang telah dilakukan oleh developer, apabila sudah sesuai maka tasklist yang sudah selesai dapat di input ke portal internal hashmicro dengan persetujuan dari Business Analyst, baru dapat melakukan meeting progress report dengan client, namun jika belum sesuai dan masih ditemukan error, bugs atau ketidaksesuaian feature yang didevelop developer dengan yang diminta maka perlu diberitahukan kepada developer untuk diperbaiki.*

### **3.2.2.2. Training modul *Inventory, Accounting, Purchasing, Sales, Leads dan Manufacturing* (Minggu 4-6)**

*Training* dilakukan paralel di minggu ke 4-6 dengan fase *development phase project, training* yang dilakukan adalah untuk modul *Inventory* (manajemen gudang), *accounting* (pencatatan keuangan), *Purchasing* (pembelian barang) , *Sales* (penjualan barang), *Leads* (pencatatan customer yang berpotensi untuk dihubungi), dan *Manufacturing* (pembuatan barang manufaktur). *Training* untuk modul *accounting* dan *manufacturing* dilakukan lewat *online meeting*, dan untuk modul *inventory, purchasing, sales* dan *leads* dilakukan dengan menonton video penjelasan modul yang terdapat di portal internal. Setiap selesai sesi *Training* akan diberikan *assignment* untuk mengetes pemahaman terkait modul yang sudah dipelajari, setelahnya akan mendapatkan review terkait *assignment* yang telah dikerjakan. Pada saat pengerjaan jika terdapat hal yang kurang dimengerti supervisor lapangan yakni Sinta Kartika akan membantu mahasiswa untuk dapat memahami akan maksud dari *assignment* yang dikerjakan serta tujuannya dari training yakni membuat *trainee* memahami modul *Inventory, Accounting, Purchasing, Sales, Leads dan Manufacturing* dengan lebih baik agar memiliki pemahaman yang lebih mendalam terkait *product knowledge* dari modul-modul yang sudah dipelajari. Gambar 3.2.2.2 dibawah ini merupakan flowchart rangkaian kegiatan pada saat *training*.

U N I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A



Gambar 3.2.2.2 flowchart rangkaian kegiatan pada saat mendapatkan *Training*

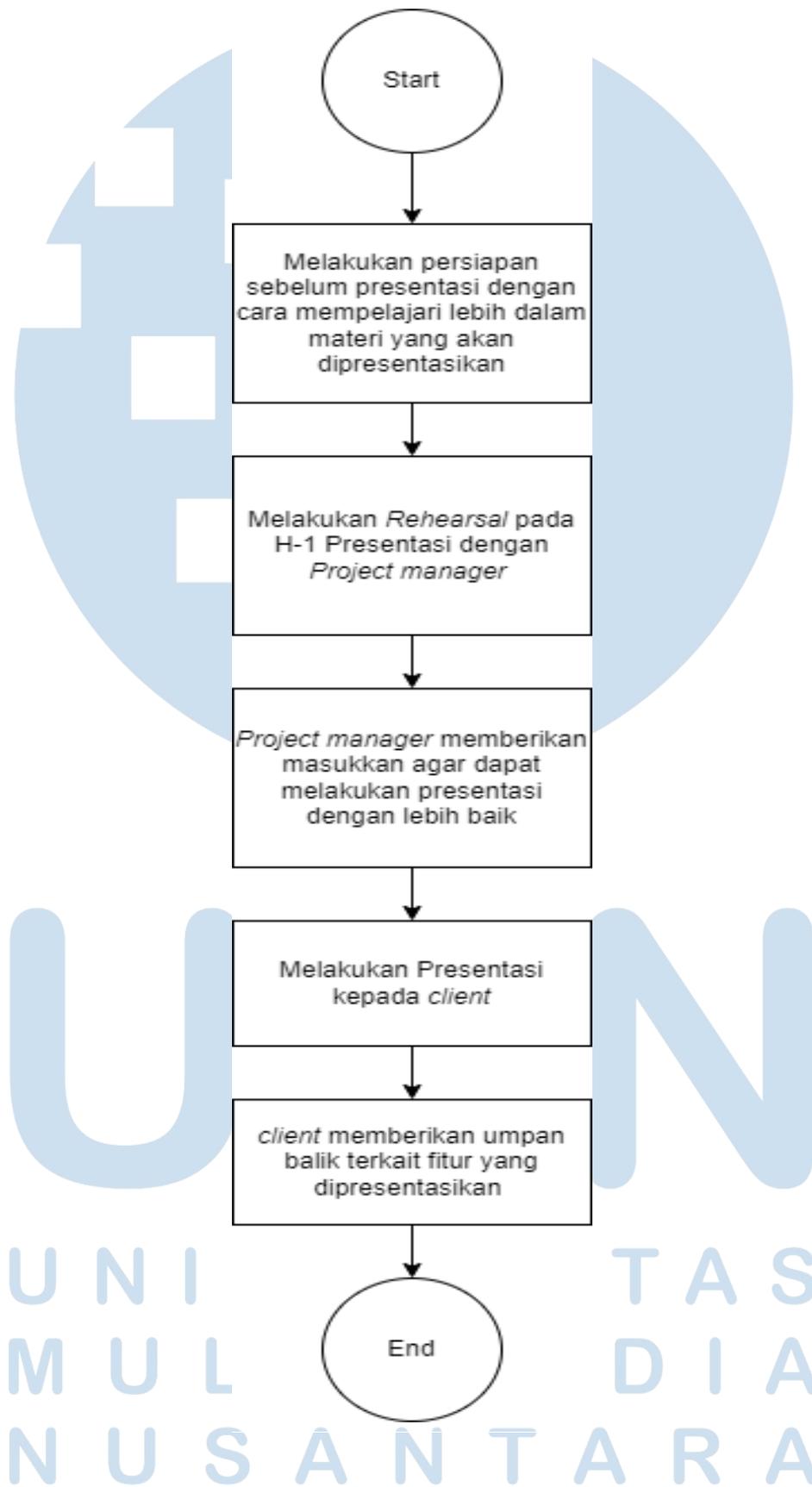
### 3.2.2.3. *Meeting Progress Report* kepada *client* (Minggu 1, 2, 3, 6, 10, 13, 14, 15, 19)

*Meeting progress report* dilakukan pada minggu ke 1,2,3,6,10,13,14,15,19 secara *online*, yang bertujuan untuk mempresentasikan *update* kepada *client* terkait feature modul POS dan Inventory yang telah *develop* dan juga telah di *bug fixing* agar *progress* yang dikerjakan oleh *team project* dapat diketahui oleh *client*. Pada *meeting progress report* ini pada minggu ke-1-2 masih dilakukan oleh *project manager*, pada minggu seterusnya yakni pada minggu ke 3, 6, 10, 13, 14, 15, dan 19 mahasiswa diberi kepercayaan untuk memaparkan dan mempresentasikan hasil yang telah dikerjakan tiap minggunya kepada *client*. Tentunya setiap sebelum melakukan presentasi mahasiswa harus menguasai fitur

yang akan dipresentasikan, biasanya H-1 sebelum melakukan presentasi mahasiswa dan *project manager* akan melakukan *rehearsal* sebelum presentasi kepada *client* agar pada saat presentasi terlihat menguasai fitur yang akan dipresentasikan. Selain itu, *project manager* juga memberikan saran setelah melakukan *rehearsal* terkait hal-hal yang perlu diperbaiki saat menyampaikan presentasi. Setelah melakukan presentasi pada *client* biasanya terdapat umpan balik yang diberikan terkait fitur yang dipresentasikan, mahasiswa perlu mencatatnya dan mendiskusikannya dengan *project manager* untuk menemukan solusi atas umpan balik yang diberikan oleh *client*. Gambar 3.2.2.3 dibawah ini merupakan *flowchart* rangkaian kegiatan yang dilakukan saat *meeting progress report*.

UMMN

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA



#### **3.2.2.4. UAT (User Acceptance Testability) ke-1 untuk modul POS (Point of sales) dan Inventory (Minggu ke-7)**

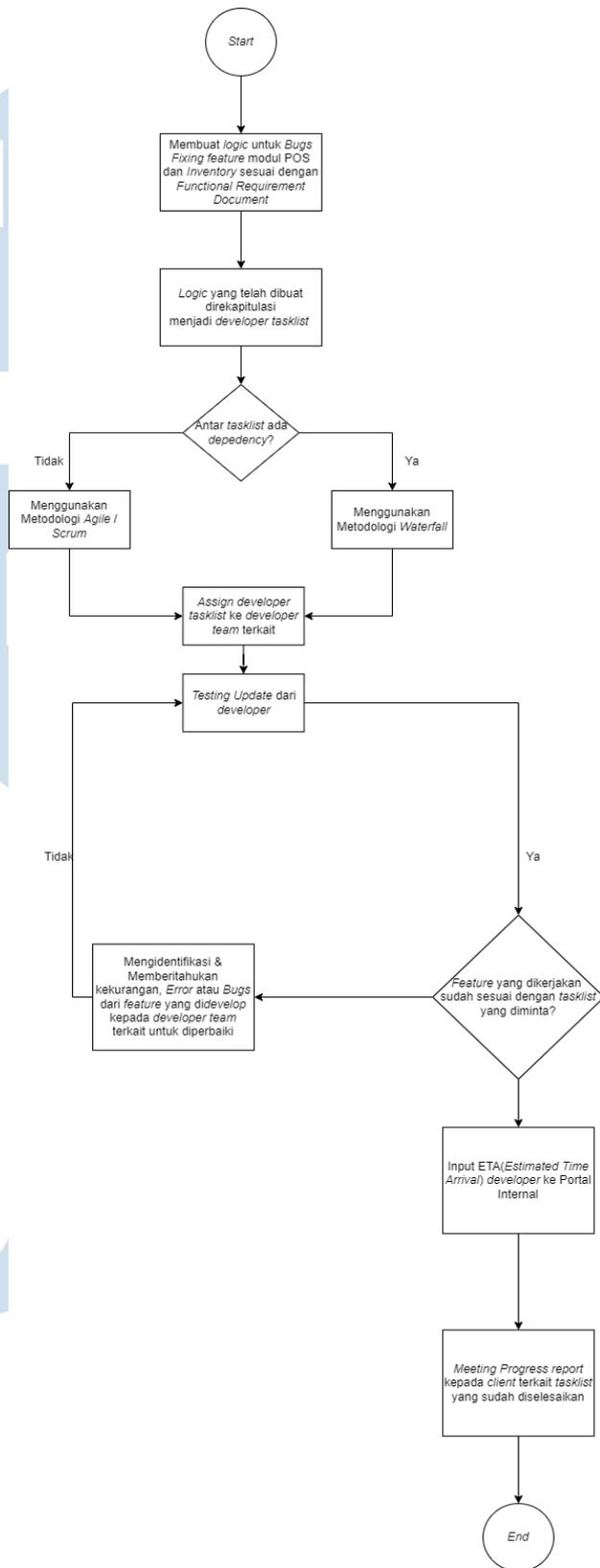
UAT (*User Acceptance Testability*) dilakukan pada minggu ke-7 dengan mahasiswa beserta *project manager* dan juga Rekan *ERP Implementation intern* mendatangi tempat client yang berada di Luar Kota untuk melakukan *On-site meeting* dan *testing* terkait *feature* yang telah di *develop* pada fase sebelumnya dimana pada fase *development testing* dilakukan oleh team *project*, namun pada fase ini dilakukan oleh *Client* dengan *test cases* yang lebih banyak dan unik untuk menemukan *Bugs, Defect*, atau *error* dalam *feature* di modul POS dan *Inventory*.

#### **3.2.2.5. Bug Fixing setelah UAT (User Acceptance Testability) ke-1 untuk modul POS (Point of Sale) dan Inventory (Minggu 8-15)**

Pada fase *Bug fixing* ini yang dilaksanakan pada minggu ke-8 sampai dengan minggu ke-15 mahasiswa menganalisis *feedback* yang diberikan *client* terhadap *Bugs, Defect* dan *Error* yang ditemukan pada fase *User Acceptance Testability (UAT)*, lalu menjalankan langkah-langkah *flow* kerja seperti pada gambar 3.2.2.5 dibawah ini untuk melakukan:

UMMN

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA



**Gambar 3.2.2.5 Flowchart Rangkaian kegiatan pada fase Bugs Fixing**

- a. Membuat *logic* untuk *Bugs Fixing feature* Modul POS dan *Inventory* sesuai dengan *Functional Requirement Document*

Di tahap ini mahasiswa membuat logika yang mudah dipahami oleh *developer* untuk dapat mengerti *feature* yang perlu di *bugs fixing* sesuai dengan *feedback* yang diberikan *client* saat UAT.

- b. *Logic* yang telah dibuat direkapitulasi menjadi *developer tasklist* berikut *list developer tasklist* yang dibuat mahasiswa pada fase *bug fixing*:

1. *Maximum Discount amount for price fixed amount promotion* (Modul POS)

2. *Search product by initial press enter masuk ke shopping cart* (Modul POS)

3. *Scan Barcode Fixing* (Modul POS)

4. *Font size and logo adjustment for receipt* (Modul POS)

5. *Product exchange* (Modul POS)

6. *Receipt ref in receipt and order details does not same* (Modul POS)

7. *Edit create internal transfer pop up show (source location, current quantity, quantity)* (Modul Inventory)

8. *Send QTY at material request line* (Modul Inventory)

9. *Add field note, next to card number in payment screen, and it will show in receipt and in field order detail*

(Modul POS)

10. *Add field best before date in inventory adjustment plus* (Modul Inventory)

11. *Inventory adjustment date for stock opname* (Modul Inventory)

12. *Inventory adjustment minus fixing* (Modul *Inventory*)

13. *Bugs price* (Modul *POS*)

14. *Negative stock for product pack* (Modul *POS*)

15. *POS Offline loading* (Modul *POS*)

c. *Decision Making* untuk penggunaan metodologi *waterfall* atau *agile/scrum* dengan faktor pembanding *dependency* antara *task list*.

1. *Maximum Discount amount for price fixed amount promotion* (Modul *POS*) menggunakan metodologi *agile* karena dapat dilakukan secara paralel dengan *task* yang lain.

2. *Search product by initial press enter masuk ke shopping cart* (Modul *POS*) menggunakan metodologi *agile* karena dapat dilakukan secara paralel dengan *task* yang lain.

3. *Scan Barcode Fixing* (Modul *POS*) menggunakan metodologi *agile* karena dapat dilakukan secara paralel dengan *task* yang lain.

4. *Font size and logo adjustment for receipt* (Modul *POS*) menggunakan metodologi *agile* karena dapat dilakukan secara paralel dengan *task* yang lain.

5. *Product exchange* (Modul *POS*) menggunakan metodologi *agile* karena dapat dilakukan secara paralel dengan *task* yang lain.

6. *Receipt ref in receipt and order details does not same* (Modul *POS*) menggunakan metodologi *agile* karena dapat dilakukan secara paralel dengan *task* yang lain.

7. *Edit create internal transfer pop up show (source location, current quantity, quantity)* (Modul

- Inventory*) menggunakan metodologi *scrum* karena dapat dilakukan secara paralel dan perlu dilakukan *stand up meeting* terlebih dahulu.
8. *Send QTY at material request line* ) (Modul *Inventory*) menggunakan metodologi *scrum* karena dapat dilakukan secara paralel dan perlu dilakukan *stand up meeting* terlebih dahulu.
  9. *Add field note, next to card number in payment screen, and it will show in receipt and in field order detail* (Modul POS) menggunakan metodologi *agile* karena dapat dilakukan secara paralel dengan *task* yang lain.
  10. *Add field best before date in inventory adjustment plus* (Modul *Inventory*) menggunakan metodologi *scrum* karena dapat dilakukan secara paralel dan perlu dilakukan *stand up meeting* terlebih dahulu.
  11. *Inventory adjustment date for stock opname* (Modul *Inventory*) menggunakan metodologi *scrum* karena dapat dilakukan secara paralel dan perlu dilakukan *stand up meeting* terlebih dahulu.
  12. *Inventory adjustment minus fixing* (Modul *Inventory*) menggunakan metodologi *scrum* karena dapat dilakukan secara paralel dan perlu dilakukan *stand up meeting* terlebih dahulu.
  13. *Bugs price* (Modul POS) menggunakan metodologi *waterfall* karena terdapat *dependency* dengan *task* lainnya.
  14. *Negative stock for product pack* (Modul POS) menggunakan metodologi *waterfall* karena terdapat *dependency* dengan *task* lainnya.

15. *POS Offline loading* (Modul POS) menggunakan metodologi *waterfall* karena terdapat *dependency* dengan task lainnya.

d. Assign *developer tasklist* ke *developer team* terkait Untuk *tasklist* no 1,2,3 assign tugas tersebut ke *developer team* vietnam, *tasklist* 4, 5, 6 ,13, 14, 15 assign tugas tersebut *developer team* india dan untuk *tasklist* no 7,8,9,10,11, 12, assign tugas tersebut ke *developer team* Indonesia. Penggunaan metodologi *agile* diterapkan karena pada *tasklist* tersebut dapat dijalan secara paralel dengan task lainnya, untuk metodologi *scrum* dijalankan karena perlu melakukan *standup meeting* untuk menjelaskan tasknya dan di *review* pada akhir minggu terkait *tasklist* yang dikerjakan, serta penggunaan metodologi *waterfall* harus menunggu *tasklist* dari metodologi *agile* dan *scrum* selesai terlebih dahulu karena memiliki *dependency code* dalam *functionnya*.

e. *Testing update dari developer*

Melakukan *testing* dari *update developer* dan mengkomunikasikan ke *developer* jika masih terdapat *bugs*. Pada fase ini mahasiswa juga assign *task* ke *technical writer* untuk membuat *user manual* modul POS dan *Inventory* terkait penjelasan penggunaan fitur yang dikustomisasi, lalu melakukan pengecekan *user manual*, apakah sudah sesuai dengan penjelasan yang diberikan, jika belum maka meminta *technical writer* untuk merevisinya.

U N I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A

### **3.2.2.6. UAT (*User Acceptance Testability*) ke-2 dan *Training* untuk modul POS (*Point of Sale*) dan *Inventory* (Minggu 16)**

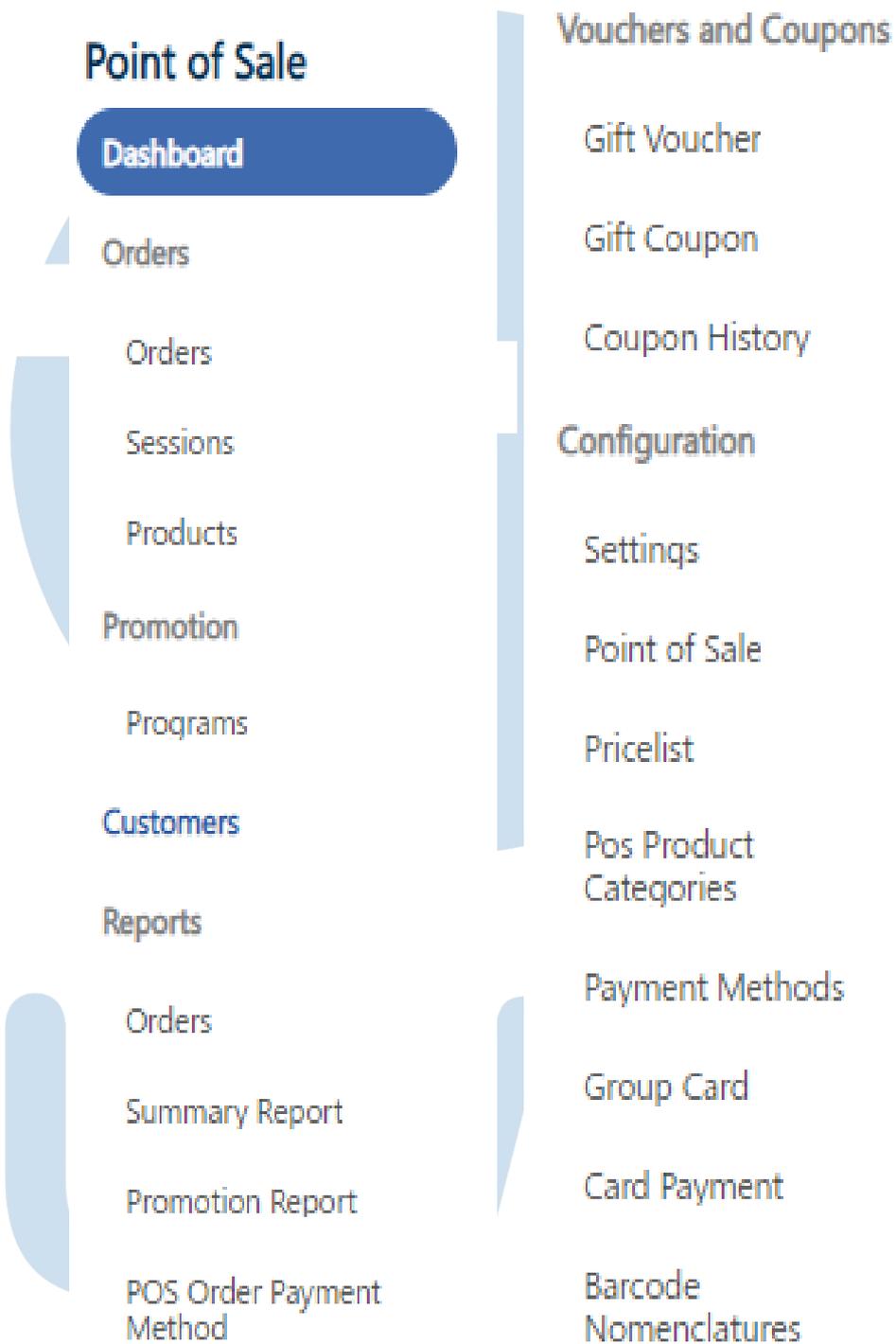
UAT (*User Acceptance Testability*) ke-2 dilakukan pada minggu ke-16 dengan mahasiswa beserta *project manager* dan juga Rekan *ERP Implementation intern* mendatangi ke tempat *client* yang berada di Luar Kota untuk melakukan *On-site meeting* dan *testing* terkait *feature* yang telah di *Bugs Fixing* dimana pada fase *Bugs fixing testing* dilakukan oleh *team project*, namun pada fase ini dilakukan oleh *Client* dengan *testcase* yang lebih banyak dan unik untuk menemukan *Bugs*, *Defect*, atau *error* dalam *feature* di modul POS dan *Inventory*, setelah proses *User Acceptance Testability* terdapat beberapa poin pada *Functional Requirement Document* yang dapat disetujui secara bersama oleh pihak *client* dan pihak tim *project* maka poin yang sudah disetujui dan disepakati tersebut dapat di buat *checklist* sebagai bukti untuk pertanggung jawaban kepada manajemen HashMicro, selain itu *team project* juga melakukan *training* kepada *End-User* terkait cara penggunaan sistem yang akan dimplementasikan ini untuk modul *Point of Sales* (POS) dan *Inventory*. Pada Gambar dibawah ini merupakan menu *feature* pada modul *Point of Sales* (POS) dan *Inventory* yang dijelaskan pada saat melakukan sesi *training* berlangsung.

U M M N

U N I V E R S I T A S

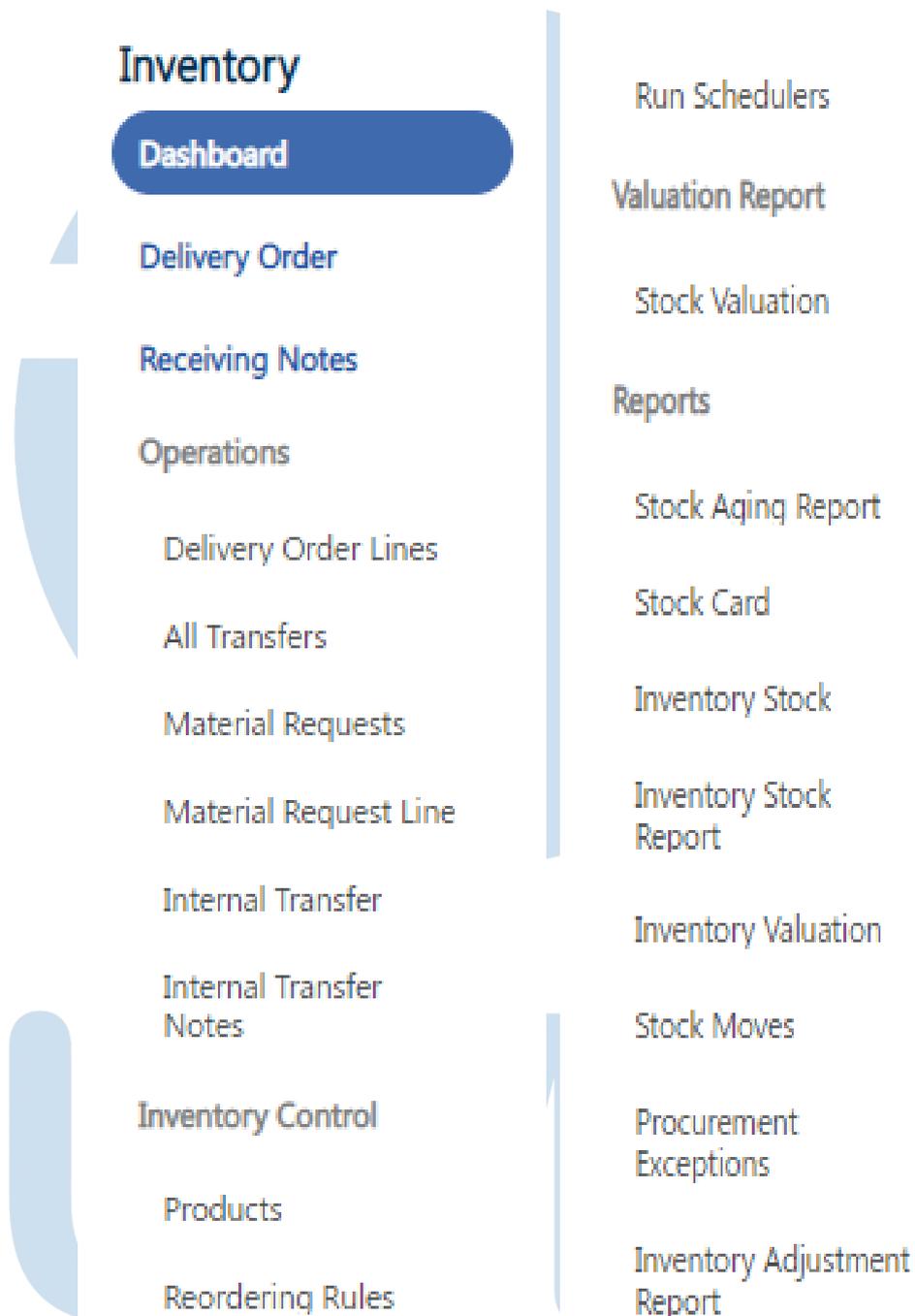
M U L T I M E D I A

N U S A N T A R A



Gambar 3.2.2 2 Tampilan menu Modul POS

U N I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A



Gambar 3.2.2 3 Tampilan menu modul *Inventory*

U N I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A

### 3.2.2.7. Bug Fixing setelah UAT (User Acceptance Testability) ke-2 untuk modul POS (Point of Sale) dan Inventory (Minggu 17-19)

Pada fase ini mahasiswa menganalisis *feedback* yang diberikan *client* terhadap *Bugs*, *Defect* dan *Error* yang ditemukan pada fase UAT ke-2, lalu mengulangi langkah-langkah flow kerja seperti pada gambar 3.2.2.5 untuk melakukan:

- a) Membuat *logic* untuk *Bug Fixing feature* Modul POS dan *Inventory* sesuai dengan *Functional Requirement Document*

Di tahap ini mahasiswa membuat logika yang mudah dipahami oleh *developer* untuk dapat mengerti *feature* yang perlu di *bug fixing* sesuai dengan *feedback* yang diberikan *client* saat UAT ke-2.

- b) *Logic* yang telah dibuat direkapitulasi menjadi *developer tasklist*

berikut *list developer tasklist* yang dibuat mahasiswa pada fase *bug fixing* ke-2.

1. *Report order bugs* (Modul POS)
2. *POS Offline bugs* (Modul POS)

- c) *Decision Making* untuk penggunaan metodologi *waterfall* atau *agile/scrum* dengan faktor pembandingan *dependency* antara *task list*.

1. *Report order bugs* (Modul POS) menggunakan metodologi *Scrum* karena bisa dilakukan secara paralel dan perlu melakukan *stand up meeting* terlebih dahulu.
2. *POS Offline bugs* (modul POS) menggunakan metodologi *agile* karena bisa dilakukan secara paralel dengan *task* lainnya.

- d) *Assign developer tasklist* ke *developer team* terkait  
Untuk *tasklist* no 1 dan 2 di *assign* ke *developer team* india, menggunakan 2 metodologi yakni *scrum* dan *agile*.

e) *Testing update dari developer*

Melakukan *testing* terhadap *update* dari *developer* untuk dapat menemukan *Bugs, defect* atau *error* pada *feature* yang di *bug fixing*, lalu mengkomunikasikannya ke *developer* jika mahasiswa masih menemukannya untuk dapat diperbaiki.

**3.2.3. Kendala yang Ditemukan**

Kendala yang dihadapi saat kerja magang ini adalah perlu adanya penyesuaian terlebih dahulu untuk memahami sistem *ERP* berbasis Odoo, karena selama ini yang dipelajari di kampus merupakan sistem *ERP* berbasis SAP, meskipun dari segi alur cara kerja dan proses bisnis terdapat kemiripan, namun memiliki fungsionalitas *feature* yang berbeda.

**3.2.4. Solusi atas Kendala yang ditemukan**

Solusi dari kendala yang dihadapi dengan mempelajari cara kerja sistem *ERP* berbasis Odoo lewat video *Training* yang ada pada portal internal HashMicro, serta bertanya secara proaktif kepada *Supervisor Lapangan, Business Analyst, Project Manager* terkait hal-hal yang kurang dimengerti.

UMMN

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA