

## BAB 3

### PELAKSANAAN KERJA MAGANG

#### 3.1 Kedudukan dan Organisasi

Selama proses *internship* dilaksanakan, praktik *internship* dilaksanakan sebagai *Software Engineer* yang ditempatkan pada divisi *Retail: Non-Digital*. *Retail: Non-Digital* memiliki tanggung jawab akan berbagai aplikasi serta layanan B2C (*Business to Customer*) yang dimiliki PT Indobest Artha Kreasi. Berbagai lini layanan yang dimaksud diantaranya, Whitelabel, Mobilepulsa, dan Mobilepulsa Next. Selama proses *internship* dilaksanakan, tim *Retail:Non-Digital* dikepalai oleh Jansen Sampurna selaku *Product Owner*, yang pada saat bersamaan menjadi pembimbing lapangan untuk mahasiswa yang sedang dalam masa *internship*. Eddy Christiandy, selaku CTO (*Chief Technology Officer*) PT Indobest Artha Kreasi, mengatasi seluruh tim yang ada, termasuk tim *retail*. Sebagai sebuah tim, tentunya tim *Retail:Non-Digital* memiliki beberapa anggota, diantaranya Brigitta Rubianti dan Kevin Hendy.

Dalam rangka menyokong penugasan serta penyampaian laporan harian, PT Indobest Artha Kreasi menggunakan aplikasi manajemen proyek bernama *Basecamp*. Dengan menggunakan aplikasi *Basecamp*, keperluan seperti *daily report*, *weekly target*, pemecahan tugas, pembagian jadwal, *cloud storage service*, hingga keperluan penunjang manajemen proyek lainnya seperti forum diskusi dan *wiki* untuk mencatat ataupun mendapatkan *product knowledge*, dapat terpenuhi dengan baik. Untuk mendukung alur komunikasi dan diskusi, PT Indobest Artha Kreasi menggunakan aplikasi pesan instan *Telegram*. Pembagian informasi seperti rapat bulanan, jadwal libur, hingga *update* terkini terkait teknologi maupun masalah produk yang perlu diberikan perhatian khusus dapat disebarluaskan melalui *group telegram*.

Tim *Retail:Non-Digital* memiliki jadwal *weekly meeting* setiap hari Senin pukul 11.00 WIB. Pertemuan mingguan ini biasanya dilaksanakan secara *online*, sejalan dengan sistem kerja hibrida yang diterapkan oleh divisi IT PT Indobest Artha Kreasi. Tim *retail* menjalankan WFH pada hari Senin, Selasa, dan Kamis. Selain *weekly meeting*, selama program *internship*, PT Indobest Artha Kreasi juga mengadakan pertemuan bulanan setiap hari Jumat di minggu ketiga setiap tiga bulan. Rapat yang disebut *Feast Meeting* ini melibatkan seluruh pimpinan dan

karyawan untuk mendengarkan pembaruan mengenai pertumbuhan perusahaan dan kendala yang dihadapi.

Proses kerja yang diadakan pada tim *Retail* dibagi menjadi 2 tim, yaitu *Small Batch* dan *Big Batch*. Tim *Small Batch* bertanggung jawab terhadap pengembangan fitur yang memiliki lingkup relatif lebih kecil, rentang batas pengerjaan tim *Small Batch* adalah 2 sampai 3 minggu. Sedangkan tim *Big Batch* memiliki tanggung jawab terhadap pengembangan fitur yang memiliki lingkup relatif besar, sehingga proses pengerjaannya diberikan waktu 6 minggu untuk fase pengembangan dan 2 minggu untuk proses QA (*Quality Assurance*). Satu *cycle* terdiri dari 8 minggu, yang biasanya berisikan 1 proyek *Big Batch* dan 2 proyek *Small Batch*. Untuk setiap perkembangan yang berjalan dapat dipantau serta dikelola melalui aplikasi *Basecamp*.

Sejalan dengan proses koordinasi serta kinerja tim yang lancar dalam menyelesaikan suatu proyek, tim *Retail:Non-Digital* menggunakan *version control* Git dengan *platform* Github. Pada awal proyek *Whitelabel* diciptakan, proyek tersebut disimpan menggunakan *platform* Gitlab. Setelah melakukan pertimbangan akan beberapa hal, pada Maret 2023, *repository* pada Gitlab dimigrasikan ke *platform* baru, yaitu Github. Sebagai alat bantu dalam penggunaan Git GUI untuk berbagai keperluan kontrol *repository*, aplikasi Fork dapat digunakan untuk melakukan berbagai kontrol *remote repository*, seperti membuat *branch* baru, mengajukan *merge request*, serta menjalankan perintah Git lainnya. Arsitektur *repository* perusahaan disusun secara terpadu, sehingga setiap kali ada pembaruan, fitur baru, atau perbaikan, diwajibkan untuk membuat *branch* baru yang akan diawasi oleh Jansen Sampurna selaku PO sebelum digabungkan dengan *branch staging* atau *development*. Sebelum pembaruan diluncurkan, tahap *staging* melibatkan tim QA (*Quality Assurance*) untuk memastikan fitur sesuai dengan standar perusahaan, serta tidak ada kesalahan atau *bug* dalam pembaruan yang akan dimasukkan ke *production branch*.

### 3.2 Tugas yang Dilakukan

Selama proyek *internship* berlangsung, tugas yang dikerjakan berupa perbaikan terhadap *bug* dan juga pengembangan fitur yang terdapat dalam layanan *Whitelabel*. Salah satu fitur yang dikembangkan adalah fitur *bulk import* pada *shop*. Pengembangan ini termasuk ke dalam penugasan tim *Big Batch* yang dilaksanakan selama 1 *cycle*. Pada awal pengerjaan, dilakukan *research* terhadap *library* yang

akan dipakai untuk melakukan pembacaan *file excel*. Setelah diputuskan untuk menggunakan *library* Openspout, proses pengembangan dimulai. Pada awal proses pengembangan, dibuat sebuah *Base Service* yang fungsinya untuk membaca *file excel*, dengan adanya *service* tersebut, kedepannya proses pengembangan yang serupa akan lebih mudah diimplementasikan.

Setelah *Base Service* selesai diciptakan, *Base Service* tersebut diimplementasikan pada sistem *Bulk Upload Product Shop*. Langkah selanjutnya adalah melakukan penyesuaian *service Bulk Upload Product Shop* agar sesuai dengan ketentuan *upload product* seperti ketentuan harga, kuantitas, SKU, dan lain-lain. Selain itu, dibuat juga mekanisme *error handling* terhadap produk yang gagal diunggah karena masalah validasi. Produk yang gagal diunggah akan disimpan dan ditampilkan kepada pengguna, agar pengguna mengetahui produk mana yang gagal diunggah dan alasan kegagalannya.

Untuk memastikan tidak ada gangguan pada *queue server*, penggunaan *worker server* disesuaikan dan ditempatkan pada *worker* yang tidak memiliki beban besar. Selain itu, *websocket* diintegrasikan untuk melakukan *broadcasting* notifikasi terhadap produk yang berhasil diunggah maupun yang gagal diunggah. Seluruh proses ini diakhiri dengan melakukan *quality assurance* melalui serangkaian *test case* yang telah disusun oleh tim QA, guna memastikan bahwa fitur berjalan dengan baik dan memenuhi kebutuhan pengguna.

### 3.3 Uraian Pelaksanaan Magang

Proses *internship* di PT Indobest Artha Kreasi berlangsung selama 640 jam. Pelaksanaan kerja magang terurai pada Tabel 3.1.

Tabel 3.1. Tugas yang dilakukan setiap minggu selama magang

Minggu Ke-	Pekerjaan yang dilakukan
1	<i>Onboarding</i> , menyiapkan proyek Mobilepulsa, Mobilepulsa Admin Panel, dan Whitelabel, serta mempelajari kaidah penulisan program untuk setiap produk dari tim <i>Retail:Non-Digital</i> . Memulai proyek fitur duplikasi produk <i>shop</i> , dengan menyusun alur program berdasarkan sistem <i>shop</i> yang sudah ada.
Lanjut pada halaman berikutnya	

Tabel 3.1 Tugas yang dilakukan setiap minggu selama magang (lanjutan)

Minggu Ke-	Pekerjaan yang dilakukan
2	Membuat tampilan serta merancang alur sistem pada <i>backend</i> , meliputi <i>Controller</i> dan <i>Database</i> untuk fitur duplikasi produk <i>shop</i> .
3	Melakukan pengecekan ulang terhadap fitur yang telah dibuat serta meminta tim QA untuk memastikan fitur yang dibuat telah memenuhi seluruh kriteria dan standar perusahaan.
4	Melakukan inisiasi proyek pengembangan fitur untuk membuat <i>custom layout</i> menggunakan Tailwind. Membuat <i>components frontend</i> menggunakan Tailwind, dan mengintegrasikannya dengan sistem <i>backend</i> , meliputi <i>Routes</i> , <i>Validation</i> , dan <i>Controller</i> .
5	Diskusi dengan PO terkait perbaikan fitur produk MLO dan melakukan <i>setup</i> proyek MLO. Menambahkan detail fitur terkait perbaikan, yakni menambahkan detail <i>region</i> untuk fitur POS, menghilangkan kolom <i>input</i> yang sudah tidak digunakan pada <i>input</i> transaksi MLO.
6	Diskusi dengan PO terkait penambahan fitur pada <i>dashboard</i> CMS, menambahkan jumlah pembeli menggunakan <i>one page checkout</i> pada <i>dashboard</i> CMS. <i>Setup</i> proyek MLO NIPOS, menambahkan tabel estimasi SLA <i>days</i> pada NIPOS, menambahkan <i>automatic redirect</i> menggunakan QR pada resi NIPOS. Melakukan <i>setup</i> integrasi proyek MLO SiCepat. Membuat sistem <i>manifest</i> untuk MLO SiCepat, membuat <i>automated closing system</i> untuk <i>manifest</i> SiCepat.
7	Membuat sistem catatan untuk pelanggan yang ingin melakukan deposito. Mengintegrasikan sistem catatan dengan seluruh produk prabayar MobilePulsa, meliputi listrik, prabayar internasional, pertagas, <i>voucher</i> , dan PPOB. Membuat tabel laporan pada sistem admin terhadap deposito yang telah dilakukan pengguna.
Lanjut pada halaman berikutnya	

Tabel 3.1 Tugas yang dilakukan setiap minggu selama magang (lanjutan)

Minggu Ke-	Pekerjaan yang dilakukan
8	Membuat <i>query</i> untuk data deposito berdasarkan <i>user query builder</i> , menambahkan <i>request</i> API untuk deposito pada <i>frontend</i> , memodifikasi <i>model eloquent</i> pascabayar dan prabayar untuk mendapatkan data deposito secara terpisah.
9	Diskusi dengan PO terkait project <i>bulk upload</i> produk <i>shop</i> , meninjau <i>library</i> yang akan digunakan untuk membaca <i>file excel</i> , membuat sistem dasar untuk membaca <i>file excel</i> , mengubah algoritma menjadi <i>dispatchable job</i> , membuat <i>base importer</i> dan mengimplementasikan <i>base importer</i> pada <i>class bulk upload product shop</i> , membuat algoritma untuk melakukan <i>mapping</i> terhadap produk dengan kolom detail produk, memodelisasi data produk, dan membuat algoritma untuk melakukan <i>bulk insert</i> foto produk.
10	Membuat validasi data produk, melakukan optimasi terhadap algoritma validasi, dan mencari cara untuk melakukan <i>bulk insert</i> terhadap produk yang telah divalidasi.
11	Memperbaiki <i>bugs</i> pada validasi data produk, membuat variabel untuk memisahkan data yang tervalidasi dan tidak tervalidasi. Merancang algoritma <i>backend</i> pada <i>Controller</i> untuk melakukan <i>bulk insert</i> produk <i>shop</i> . Memuat pesan validasi produk untuk keperluan <i>error handling</i> dan pemberitahuan kepada <i>user</i> . Melakukan <i>case test</i> menggunakan 500 data.
12	Membuat <i>function</i> untuk melakukan konversi berat gram menjadi kilogram, merancang <i>service</i> untuk menghasilkan <i>product slug</i> , mengimplementasikan kinerja <i>job</i> pada <i>service class bulk import product shop</i> , membuat <i>table database</i> untuk menampung data yang tidak tervalidasi, dan menggunakan <i>websocket</i> untuk melakukan <i>broadcasting</i> notifikasi terhadap aksi yang berlangsung selama proses <i>bulk insert</i> , seperti proses berhasil dan proses gagal.
Lanjut pada halaman berikutnya	

Tabel 3.1 Tugas yang dilakukan setiap minggu selama magang (lanjutan)

Minggu Ke-	Pekerjaan yang dilakukan
13	Membuat <i>Controller</i> dan <i>Routes</i> untuk menarik data produk yang tidak tervalidasi menggunakan API, membuat <i>processing state</i> untuk mencegah <i>query user</i> secara bersamaan, membuat <i>flag</i> pada <i>frontend</i> untuk menonaktifkan tombol unggah berdasarkan <i>state</i> yang dikirim oleh <i>backend</i>
14	Membuat <i>broadcast</i> notifikasi menggunakan <i>websocket</i> untuk menandakan proses <i>import</i> , memperbaiki <i>issue typecasting</i> pada proses modelisasi data produk, melakukan proses QA dengan tim QA dan PO. Melakukan <i>deploy</i> fitur ke dalam <i>production</i> .
15	Berdiskusi dengan Jansen Sampurna selaku PO terkait masalah <i>timeout</i> pada saat pengunggahan foto produk. Memisahkan <i>queue job</i> untuk <i>bulk insert</i> data produk dengan <i>bulk insert</i> foto produk. Memperbaiki <i>bugs</i> yang ada pada fitur <i>bulk upload</i> produk <i>shop</i> , meliputi <i>null values</i> pada harga produk jenis varian, <i>gross values</i> pada produk jenis grosir, dan opsi pada produk jenis varian. Memperbaiki isu <i>error upload</i> gagal. Membuat validasi untuk melakukan pengecekan kategori, sub-kategori, dan super sub-kategori.
16	Memperbaiki <i>bug upload</i> terhenti saat ingin melakukan <i>upload</i> lebih dari 20 produk. Membuat validasi jumlah sku, harga, berat, dan detail produk lainnya, terhadap jumlah produk yang memiliki varian.
17	Melakukan setup untuk proyek integrasi MLO JNT Express, membuat tampilan antarmuka untuk <i>manifest</i> JNT Express, merancang <i>backend</i> , meliputi <i>Controller</i> , <i>Eloquent</i> , <i>Repository</i> , <i>Routes</i> , dan <i>Traits</i> untuk keperluan integrasi JNT Express.
Lanjut pada halaman berikutnya	

Tabel 3.1 Tugas yang dilakukan setiap minggu selama magang (lanjutan)

Minggu Ke-	Pekerjaan yang dilakukan
18	Menyelesaikan <i>backend</i> , meliputi <i>Controller</i> , <i>Eloquent</i> , <i>Repository</i> , <i>Routes</i> , dan <i>Traits</i> untuk keperluan integrasi JNT Express. Menyesuaikan <i>service</i> sesuai dengan ketentuan yang dimiliki JNT Express, menyesuaikan data pada <i>Region Controller</i> dan <i>User Agen JNT Express Resource</i> agar sesuai dengan <i>request body</i> untuk mekanisme <i>get fee</i> . Melakukan <i>hit</i> API untuk mekanisme <i>get fee</i> , menyesuaikan <i>form frontend</i> untuk keperluan <i>get fee</i> , dan menyesuaikan <i>validation rules</i> sesuai dengan <i>request</i> yang dilakukan <i>form</i> pada <i>frontend</i> .
19	Mempelajari alur penggunaan API JNT EXPRESS untuk membuat <i>order transaction</i> , menyesuaikan <i>form</i> transaksi pada <i>frontend</i> , menyesuaikan <i>Transaction Service</i> dan <i>traits</i> MLO JNT Express dengan peraturan <i>request</i> yang ada. Melakukan <i>binding service</i> dengan <i>eloquent</i> yang telah dibuat untuk keperluan transaksi JNT EXPRESS.
20	Membuat <i>migration</i> untuk <i>master product</i> mlo JNT EXPRESS, <i>tenant product</i> mlo JNT EXPRESS, dan <i>cashback</i> mlo JNT EXPRESS. Membuat <i>Cashback Controller</i> , <i>Model</i> , dan <i>resource</i> JNT EXPRESS. Mengintegrasikan <i>transaction service</i> untuk melakukan <i>hit</i> API transaksi JNT EXPRESS.

### 3.4 Perangkat Penunjang

Selama melaksanakan praktik *internship*, terdapat berbagai *software* dan *hardware* yang digunakan. Berikut ini adalah beberapa *software* yang digunakan dalam menyelesaikan praktik *internship*.

- a. Visual Studio Code v1.89.1
- b. PHP Zend Opache v7.4.33
- c. Laravel v5
- d. Composer v1.10.26

- e. Vue.js v2.6.12
- f. Node v12.22.12
- g. NPM v6.14.6
- h. NVM v0.39.7
- i. Yarn v1.22.19
- j. Redis v7.0.7
- k. Openresty v1.25.3.1
- l. MongoDB-Community v5.0
- m. Datagrip v2023.3.2
- n. MySQL v5.7.40
- o. Git v2.34.1
- p. Fork v2.27.1
- q. Isomnia v9.2.0
- r. OpenVPN Connect v3.4.4
- s. Google Chrome v124.0.6367.210
- t. Basecamp 3 v2.3.14
- u. Kaspersky Endpoint Security For Windows v12.3.0.493
- v. Windows Subsystem for Linux v2.0.14.0
- w. Windows 11 Home Single Language v23H2
- x. Ubuntu 22.04.3 LTS

Selain itu, berikut ini adalah beberapa perangkat keras yang digunakan.

- a. Lenovo Legion 5
- b. AMD 5 5600H with Radeon Graphics CPU; RTX 3050 4GB GPU; 18-core (6-core dan 12-thread)



- c. RAM 16 GB DDR4-3200
- d. SSD WD-BLACK WDC PC SN730 SDBPNTY-512G-1101

### 3.5 Proses Pelaksanaan Magang

Pada situs web layanan Whitelabel, terdapat dua akses pengguna, yaitu *End-User* yang bisa bertindak sebagai agen maupun pembeli, serta administrator yang bermitra dengan PT Indobest Artha Kreasi. Setiap akses dalam sistem Whitelabel memiliki peran dan fungsinya masing-masing, dengan fitur yang dapat digunakan sesuai kebutuhan mereka. Penjelasan lebih lanjut mengenai peran dan fungsi kedua akses dijelaskan di bawah ini.

Berikut adalah beberapa fungsi yang dapat dilakukan oleh *End-User*:

- a. Pendaftaran
- b. *Login*
- c. Mengubah atau mengatur profil
- d. Mengisi ulang saldo
- e. Mentransfer saldo ke *End-User* lain
- f. Melakukan transaksi PPOB atau berbelanja
- g. Menerima notifikasi transaksi
- h. Melihat laporan transaksi yang telah dilakukan
- i. Mengunduh struk transaksi
- j. Mengunduh laporan transaksi (PDF/Excel)
- k. Mengakses halaman daftar harga (*Pricelist*)
- l. Membuat gambar daftar harga yang telah disesuaikan
- m. Menaikkan harga produk (mengatur margin keuntungan)
- n. Mendaftarkan *downline* atau agen di bawah pengguna tersebut

Untuk Admin, berikut adalah beberapa fitur yang dapat diakses melalui *admin panel*:

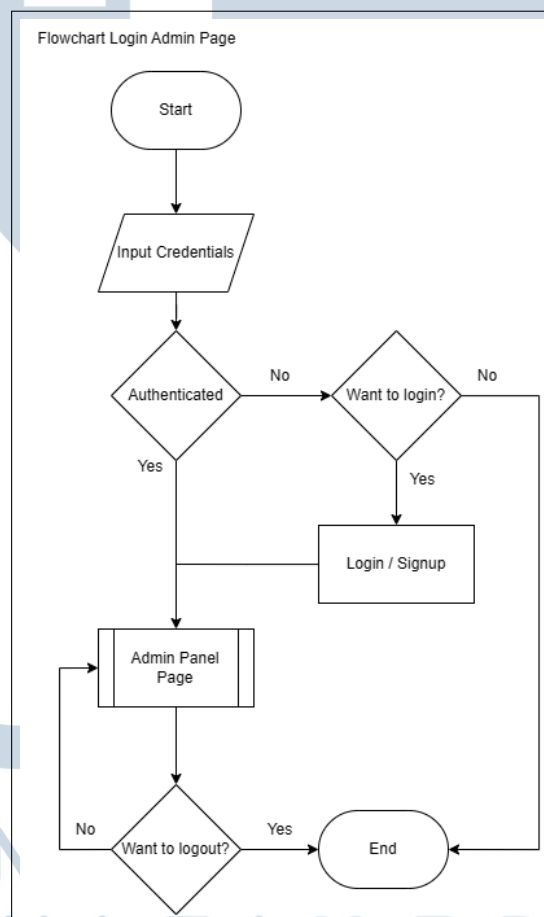
- a. Registrasi
- b. *Login*
- c. Membuka dashboard pada *admin panel*
- d. Membuka audit log pada *admin panel*
- e. Mengelola produk *shop* dan PPOB
  - i. Mengelola transaksi
  - ii. Mengelola daftar harga
  - iii. Menetapkan produk *Resell* untuk *shop*
  - iv. Menetapkan *cashback* untuk PPOB
  - v. Menetapkan operator untuk produk PPOB
- f. Mengelola laporan
  - i. Laporan keuntungan
  - ii. Laporan deposit Admin
  - iii. Laporan transaksi agen
  - iv. Laporan deposit agen
  - v. Laporan *cashback* agen
  - vi. Laporan penjualan produk
- g. Mengelola fasilitas web
  - i. Pengaturan utama web
  - ii. *Platform* pembayaran
  - iii. Melakukan pendaftaran pengguna
  - iv. Verifikasi *login user*
  - v. Halaman utama dan *blog* untuk situs utama web.
- h. Mengelola konten web
  - i. Membuat halaman kustom
  - ii. Membuat tampilan *carousel* untuk iklan pada halaman utama

- iii. Konten *blog*
- iv. Kustomisasi media untuk ikon dan logo web
- v. Kustomisasi menu sidebar
- vi. Memberi tagar untuk halaman kustom
- i. Mengelola deposit
  - i. Mengelola transaksi deposit
  - ii. Mengelola pilihan bank untuk pengisian saldo deposit *End-User*
  - iii. Penambahan saldo deposit *admin*
- j. Mengelola pengguna
  - i. Mengelola akses dan izin pengguna
  - ii. Mengatur peran pengguna
- k. Mengatur profil dan pengaturan toko
  - i. Kustomisasi domain web
  - ii. Kustomisasi tampilan web
  - iii. Mengatur akses web
- l. Mengatur notifikasi
  - i. Membuat dan mengirim notifikasi ke pengunjung web
  - ii. Membuat dan mengirim notifikasi ke agen
- m. Mengatur integrasi
  - i. Mengelola integrasi web dengan layanan pihak ketiga
  - ii. Memproduksi aplikasi *native* Android

Fitur yang tertera pada sub bab 3.5 adalah fitur yang sudah berjalan dan dapat dinikmati oleh pengguna yang berlangganan layanan Whitelabel. Salah satu fitur terbaru Whitelabel adalah *shop*. Sebelumnya Whitelabel hanya memiliki layanan untuk berjualan PPOB, namun dengan hadirnya fitur *shop*, pengguna dapat memulai bisnis *e-commerce* secara pribadi, dan tentunya memiliki kemampuan kustomisasi yang lebih fleksibel. Mengingat layanan *shop* yang

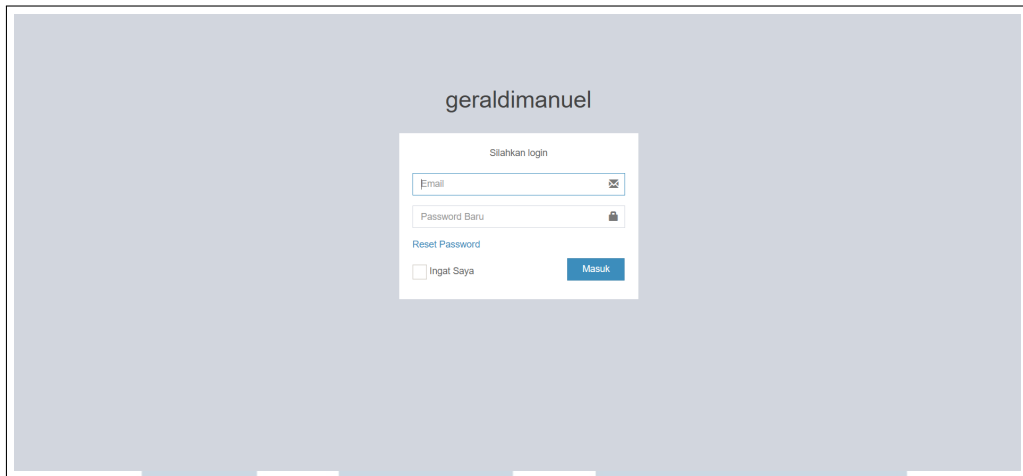
baru saja dikembangkan, tentunya fitur yang berada didalamnya akan terus dikembangkan sesuai dengan perkembangan teknologi dan kebutuhan pengguna. Dalam kesempatan proses kerja magang ini, PT Indobest Artha Kreasi memutuskan untuk menambahkan fitur *bulk upload*, agar pengguna dapat melakukan *input* produk dalam jumlah banyak dalam sekali waktu, serta meringankan kerja *worker server* dengan menurunkan jumlah *query*. Fitur tersebut terurai dalam sub sub bab 3.5.1.

### 3.5.1 Fitur *Bulk Upload* Produk Shop



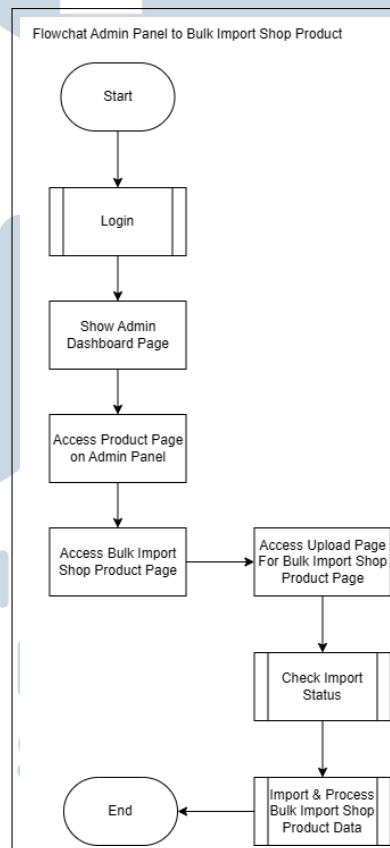
Gambar 3.1. *Flowchart Login Admin Page*

Gambar 3.1 merupakan *flow* autentikasi *user* dalam mengakses *admin panel*. Untuk mengakses *admin panel*, pengguna perlu melakukan *login* jika sudah memiliki akun dan *sign-up* bila pengguna belum memiliki akun. Apabila pengguna tidak memiliki akun dan memutuskan untuk tidak membuat akun, maka *admin panel* tidak dapat diakses.

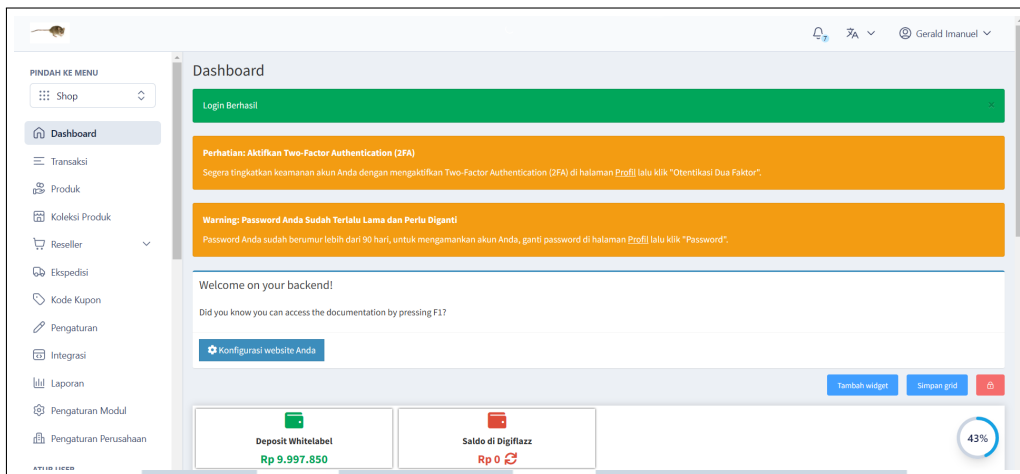


Gambar 3.2. Halaman *Login Admin Panel*

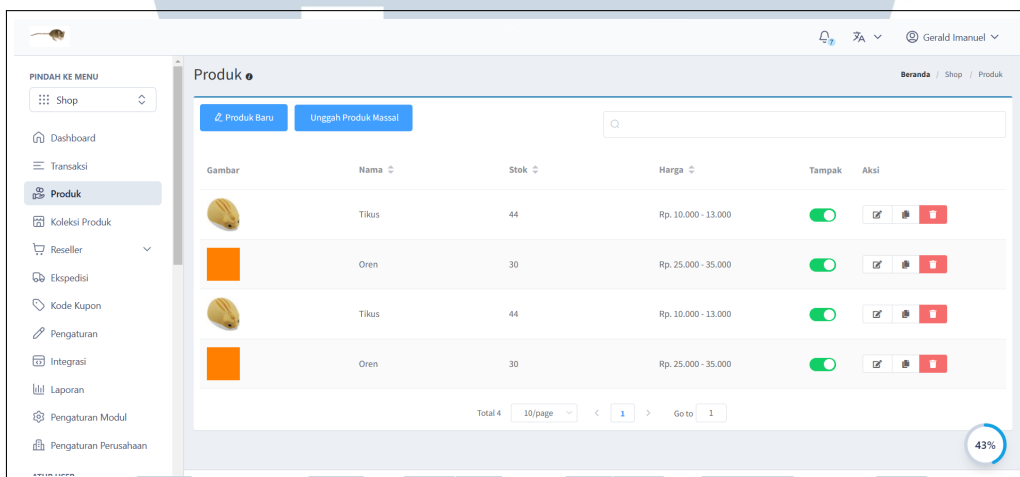
Gambar 3.2, merupakan tampilan awal saat *backend admin panel* diakses. Pengguna diharuskan melakukan *login* jika ingin mengakses fitur pada *admin panel*, sebaliknya jika pengguna tidak memiliki akun, maka *admin panel* tidak dapat diakses.



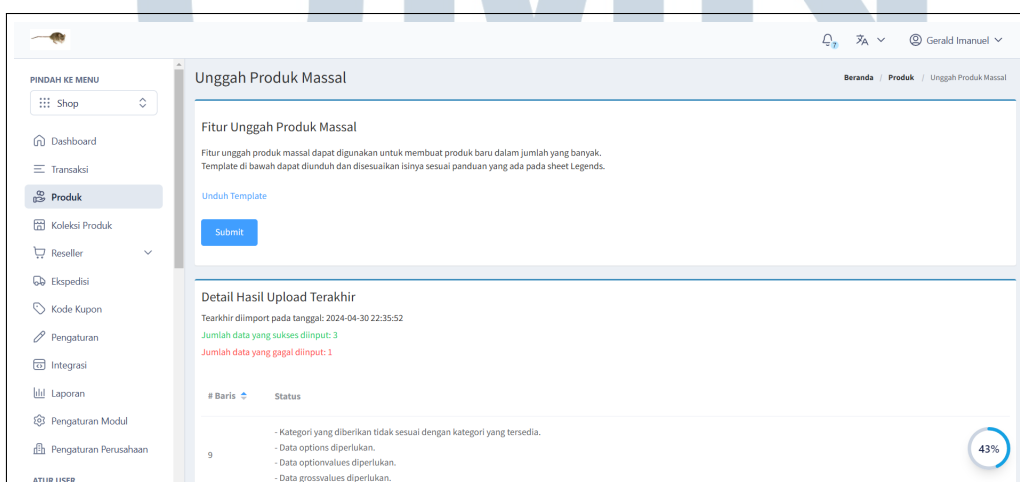
Gambar 3.3. *Flowchart Flow Admin Page to Bulk Import Page*



Gambar 3.4. Halaman *Dashboard Admin Panel*

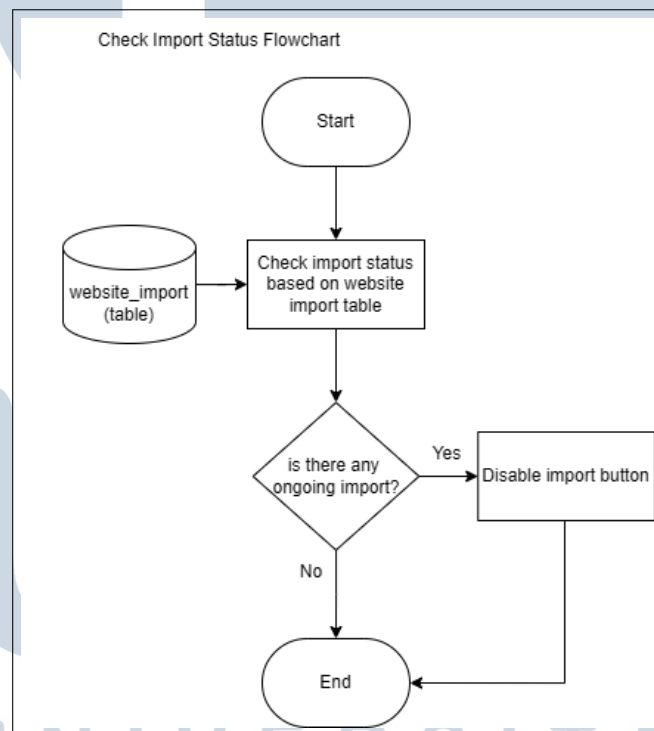


Gambar 3.5. Halaman *Produk*

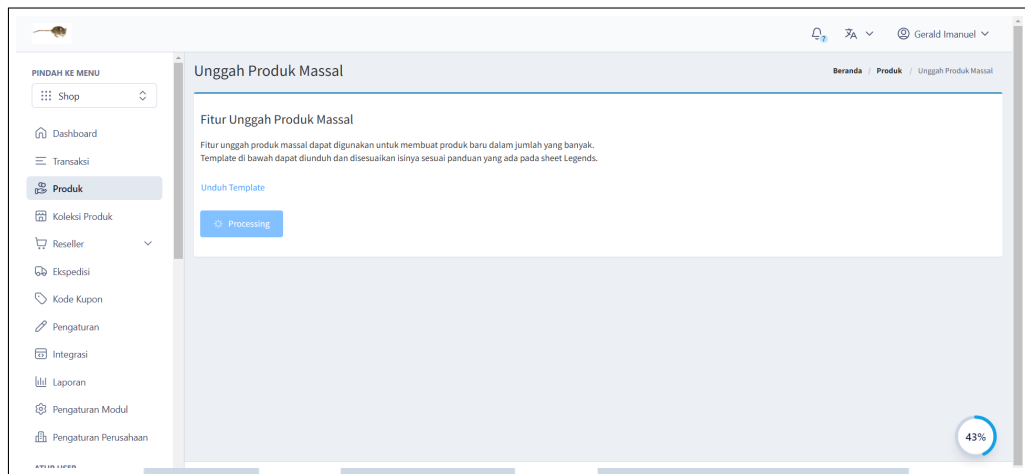


Gambar 3.6. Halaman *Bulk Upload*

Setelah pengguna melakukan *login* pada *admin panel*, pengguna akan diarahkan menuju halaman *dashboard* pada *admin panel*, seperti yang tertera pada gambar 3.4. Sesuai dengan *flowchart* pada gambar 3.3, pengguna dapat melakukan akses terhadap halaman produk. Halaman produk dapat digunakan oleh pengguna (pemilik toko) untuk melakukan input data produk, fitur yang dibuat adalah untuk mengunggah banyak produk dalam sekali waktu. Opsi tersebut tersedia pada halaman produk, seperti yang terlihat pada ilustrasi Gambar 3.5. Apabila pengguna memutuskan untuk melakukan *input* produk secara *bulk*, sesuai dengan alur *flowchart* pada gambar 3.3, pengguna akan diarahkan menuju halaman *bulk upload shop product*. Gambar 3.6 merupakan tampilan halaman *bulk upload shop product*. Pada halaman ini, pengguna dapat mulai melakukan *input* data produk secara *bulk*, menggunakan *template file excel* yang tersedia.



Gambar 3.7. Flowchart Check Import Status

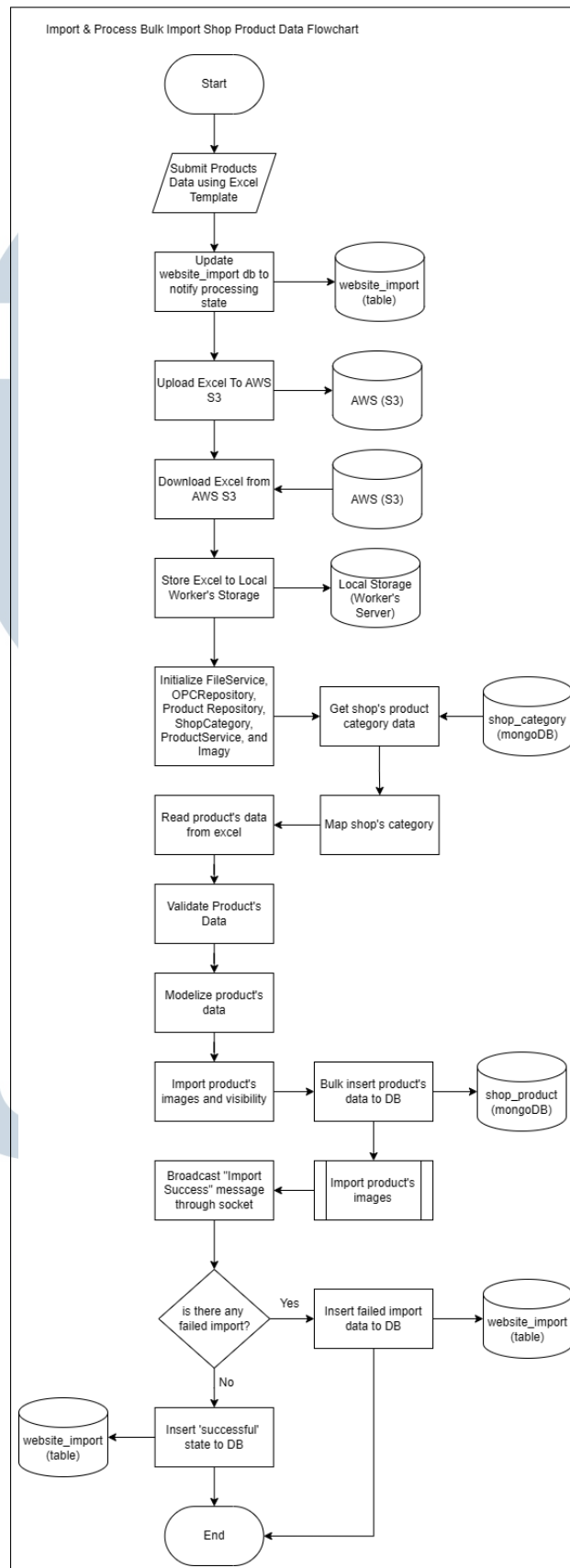


Gambar 3.8. *Processing State* pada Halaman *Bulk Upload*

Halaman *bulk upload shop product* dapat berubah sesuai dengan kondisi yang sedang terjadi. Sesuai dengan *flowchart* pada gambar 3.7, sistem akan melihat pada *database*, apakah sistem sedang mengalami proses *import* atau tidak. Apabila sedang terjadi proses *import* maka sistem akan menonaktifkan tombol untuk melakukan pengunggahan data seperti yang dapat dilihat pada gambar 3.8, sebaliknya apabila sedang tidak terjadi proses *import* tombol untuk melakukan pengunggahan akan aktif dan proses pengunggahan produk dapat dilakukan. Halaman pada gambar 3.6 menunjukkan bahwa tidak ada proses yang sedang berjalan dan proses pengunggahan data dapat dilanjutkan.

UMN  
UNIVERSITAS  
MULTIMEDIA  
NUSANTARA



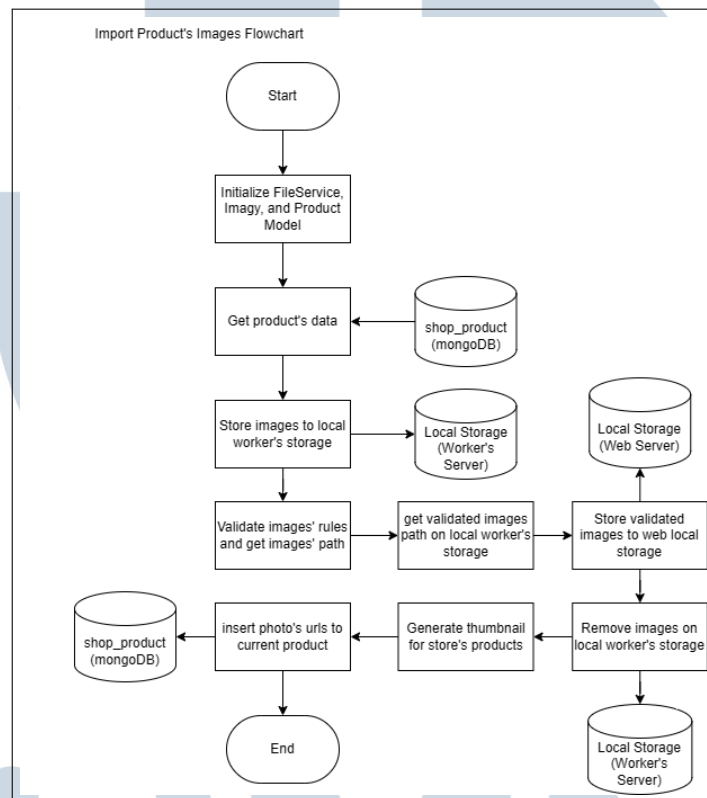


Gambar 3.9. Flowchart Bulk Import Process

Gambar 3.9 merupakan *flowchart* alur proses yang terjadi saat pengguna melakukan *import* data produk. Sesuai dengan *flowchart* pada gambar 3.9, setelah pengguna mengunggah data produk melalui *file template excel* yang diberikan, proses *bulk upload shop product* mulai dijalankan. Pada awal proses, sistem akan melakukan *update* terhadap *database* untuk memberitahu sistem bahwa ada proses yang sedang berjalan. Hal ini merupakan salah satu tindakan preventif akan terjadinya dua proses dalam satu waktu yang dapat menyebabkan *race condition*. Selanjutnya, sistem akan mengunggah *file excel* ke dalam salah satu *storage* yang dimiliki sistem Whitelabel, yaitu Amazon AWS S3. Setelah proses pengunggahan selesai dilakukan, *worker server* akan mengunduh *file excel* yang berada pada Amazon AWS S3. Sistem ini dirancang dengan tujuan untuk mengatasi kesenjangan penyimpanan yang muncul saat sistem beralih dari penggunaan *web server* ke *worker server*. Pada tahap awal pengunggahan, sistem akan menerima *file* pada *web server*. *File* tersebut kemudian akan dipindahkan ke *worker server* untuk diproses lebih lanjut. Perlu diketahui bahwa kedua *server* ini memiliki penyimpanan yang berbeda. Salah satu opsi untuk menangani kesenjangan penyimpanan ini adalah dengan menggunakan *media* penyimpanan sementara. Dalam hal ini, PT Indobest Artha Kreasi memilih Amazon AWS S3 sebagai salah satu penyimpanan yang terintegrasi dengan sistem Whitelabel.

Setelah *file* siap diproses, sistem akan melakukan inisialisasi terhadap *service*, *controller*, *repository*, *model*, dan *library* yang diperlukan. Selanjutnya, melalui model *Shop Category* yang telah diinisialisasi, sistem akan memperoleh dan melakukan *mapping* terhadap kategori yang terdapat pada *database*. Perlu didatakannya data kategori adalah agar sistem dapat melakukan validasi terhadap kategori produk yang diunggah. Setelah semua inisialisasi dilakukan, sistem akan mulai untuk membaca data *file excel* yang tersedia. Pada awal pemrosesan data, data yang didapat akan melalui proses validasi dan dilanjutkan proses modelisasi, fungsinya agar data yang akan dimasukkan ke dalam *database* sudah lebih terstruktur dan mengurangi terjadinya kesalahan karena perbedaan format data dengan data pada *eloquent repository*. Setelah data melalui proses modelisasi, sistem akan mulai membaca dan menyimpan kolom yang berisikan foto dan preferensi visibilitas pengguna akan produk tersebut. Selanjutnya setelah semua data sudah disiapkan, akan dilakukan proses *insert* data produk dan diikuti oleh proses *insert* foto produk. Proses *insert* data produk dan foto produk perlu dipisah untuk mengurangi waktu *query* dan meminimalisir terjadinya *timeout*. *Timeout* perlu diatur dalam sistem yang memisahkan *web server* dengan *worker server*. Proses pengunggahan foto

cenderung memiliki waktu yang relatif panjang, tergantung dengan ISP (*Internet Service Provider*) yang digunakan. Dengan demikian, perlu penggunaan *worker* dan *queue* perlu dipisahkan, agar keduanya memiliki ruang (waktu) yang lebih besar dalam melakukan pemrosesan.



Gambar 3.10. *Flowchart Import Products' Images*

Data produk yang sudah berhasil dimasukan, selanjutnya akan melalui proses *import* foto yang sudah dipersiapkan pada fase sebelumnya. *Flowchart* pada gambar 3.10 merupakan alur kerja sistem, dalam melakukan *update* data produk untuk memasukkan foto produk. Proses *import images* diawali dengan inialisasi *service*, *model*, dan *library* yang diperlukan. Selanjutnya sistem akan memanggil data produk yang sudah berhasil masuk ke dalam *database* untuk memulai proses *update* data. Foto yang diunggah pengguna adalah dalam bentuk URL, sehingga perlu dilakukan pengunduhan terhadap foto yang akan diunggah, sesuai dengan alur flowchart pada gambar 3.10, gambar yang telah diunduh akan diunggah sementara ke dalam penyimpanan *worker server* untuk melalui proses validasi. Setelah validasi terhadap ukuran foto, jenis ekstensi, dan ukuran *file* foto dilakukan, sistem akan menyimpan *path* foto yang telah tervalidasi pada penyimpanan *worker server*. Selanjutnya foto tersebut akan diunduh dan diunggah ke dalam *local*

*storage* milik *web server*. Setelah pengunggahan foto pada penyimpanan *web server* berhasil dilakukan, foto yang berada pada penyimpanan *worker server* akan dihapus, mengingat foto sudah tidak digunakan, sehingga perlu dihapus untuk mengurangi beban penyimpanan *server*. Pada fase ini, sistem sudah menyimpan data foto produk dalam penyimpanan *web server* dan sudah siap ditampilkan pada halaman produk. Untuk finalisasi dalam proses pengunggahan foto, sistem akan menghasilkan *thumbnail* produk secara otomatis, selanjutnya semua URL foto yang telah diproses akan dimasukkan ke dalam database produk yang sebelumnya sudah disiapkan. Sistem akan melakukan proses *update* terhadap data produk yang berada di penyimpanan *database MongoDB Whitelabel*.

Setelah proses *update* berhasil dilakukan, sesuai dengan alur *flowchart* pada gambar 3.9, sistem akan melakukan *brodacast* notifikasi yang menandakan proses *import* data telah selsai dan sukses dilaksanakan. Selanjutnya sistem juga akan melihat apakah ada data yang gagal diunggah akibat tidak tervalidasi. Jika tersedia data yang gagal diunggah, maka data akan diunggah ke dalam *database MySQL*, setelah itu seluruh proses *bulk import* selsai dilaksanakan. Sebaliknya, apabila tidak ada data yang gagal diunggah selama proses *bulk import*, maka sistem akan menunggah state *successful* ke dalam database *MYSQL*, setelah itu seluruh proses *bulk import* telah selsai dan sukses dilaksanakan. Produk sudah dapat dilihat pada halaman *shop* pengguna.

### 3.6 Kendala yang Ditemukan

Selama praktik *internship* dilaksanakan, terdapat beberapa kendala yang dialami. Berikut adalah beberapa kendala yang dihadapi selama proses *internship* berlangsung.

- a. Proses persiapan *environment* proyek, menyesuaikan versi PHP, Composer, Node, Openresty, MySQL, MongoDB, dan berbagai *library* penunjang lainnya yang sudah *deprecated* serta berbeda pada setiap *repository*, mempersulit proses instalasi.
- b. Penggunaan *design principles* dan *design patterns* seperti SOLID (*Single Responsibility Principle, Open-Closed Principle, Liskov Substitution Principle, Interface Segregation Principle, Dependency Inversion Principle*) pada kaidah penulisan program, seperti *Controller, Router, Model, Repository, Service, Trait*, dan *Eloquent* yang menerapkan sistem *multi-*

*tenancy* dengan banyak modul, membuat proses pengembangan terhambat karena harus mempelajari alur, struktur, serta kaidahnya terlebih dahulu.

### 3.7 Solusi terhadap Kendala yang Ditemukan

Berdasarkan kendala serta masalah yang dialami selama proses kerja *internship*, beberapa solusi telah ditemukan untuk mengatasi kendala tersebut.

- a. Mencari dokumentasi terkait *library* yang digunakan, mempelajari dokumentasi serta panduan instalasi yang dimiliki PT Indobest Artha Kreasi, serta menggunakan perangkat *version manager* seperti NVM, untuk mempersingkat proses pengaturan.
- b. Mendengarkan saran senior, mempelajari *design principles* yang digunakan, serta mempelajari alur program yang sudah dibuat sebelumnya.

