

BAB III

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

Kegiatan program kerja magang di PT Indobest Artha Kreasi dengan kedudukan *web developer intern* di departemen Internal sebagai *backend developer* yang dibawah bimbingan *Product Owner* yaitu Kak Stainley Suryadinata Winoto. Kedudukan *web developer intern* ditujukan untuk membantu mengerjakan pengembangan terhadap produk *Content Management System (CMS)* yang merupakan tanggung jawab dari departemen Internal untuk melakukan manajemen terkait transaksi-transaksi pembelian produk digital yang dilakukan melalui perusahaan Indobest Artha Kreasi.

Pengembangan *website* CMS di perusahaan IAK dilakukan dengan Bahasa pemrograman PHP dan menggunakan *framework* Laravel untuk membantu dalam pengembangan *website* dalam hal: sintaks yang ekspresif, *routing* yang fleksibel, manajemen basis data, migrasi data, keamanan, *template*, dan *testing*. *Website* CMS sendiri digunakan secara langsung oleh para karyawan yang bekerja di perusahaan IAK, contohnya divisi:

- 1) *Customer Service*: untuk memantau detail terkait transaksi pelanggan apabila ada keluhan.
- 2) *Trx Solution*: untuk mencari solusi terkait masalah dalam transaksi.
- 3) *Marketing*: untuk mendapatkan data terkait produk digital, harga produk, serta data lainnya yang diperlukan untuk akuisisi klien.
- 4) *Finance*: untuk menarik data transaksi dan membuat laporan keuangan

Ketika bekerja di divisi Internal, peserta magang beserta anggota tim lainnya melakukan koordinasi melalui aplikasi Telegram dan Basecamp. Telegram digunakan sebagai aplikasi/*software* untuk melakukan komunikasi baik secara personal maupun secara tim untuk mengirim pesan, berdiskusi dalam grup, berbagi file, dan kolaborasi lainnya. Sementara aplikasi/ *software* Basecamp digunakan

untuk memaksimalkan performa kolaborasi tim karena dapat membantu untuk mengatur dan mengelola proyek, berbagi *file/resources*, *sharing knowledge*, fitur kalender&reminder, serta dokumentasi dan pencatatan yang rapih.

3.2 Tugas dan Uraian Kerja Magang

Selama menjalankan praktik kerja magang di Divisi/ Team Internal dengan posisi sebagai *Web Developer Intern (backend)*, berikut merupakan deskripsi pekerjaan yang dikerjakan selama kurang lebih 5 bulan praktik kerja:

1. Melakukan analisa, pengembangan, *testing*, *improvement* dan evaluasi sistem produk digital IAK.
2. Bertanggung jawab dan melakukan penanganan terhadap *bug* atau *error* yang terjadi pada sistem IAK.
3. Memberikan laporan kepada manajemen terkait *progress* yang terjadi dari sistem IAK.

Berikut merupakan rincian pekerjaan dan tugas yang telah dilakukan peserta magang selama 5 bulan waktu bekerja beserta waktu mulai dan penyelesaiannya.

Tabel 3 Aktivitas Kerja Magang

No	Kegiatan	Tanggal Mulai	Tanggal Selesai
1	Onboarding Intern, Perkenalan environment & team, Set up tools, Research project & Laravel, Clone Project CMS	9 Januari 2023	12 Januari 2023
2	Mempelajari code CMS, Mengerjakan task fitur Lazada Email Table Prepaid & Lazada Email Table Postpaid, Code review fixing	11 Januari 2023	1 Februari 2023

	fitur Lazada Email Table Prepaid & Lazada Email Table Postpaid		
3	Mengerjakan tugas fixing bug validasi product_code ketika upload file excel di fitur Custom pricelist & Code review fixing bug Validasi product_code ketika upload file excel di fitur Custom pricelist.	2 Februari 2023	17 Februari 2023
4	Mengerjakan tugas fixing bug Number Required false value di fitur Add Operator	16 Februari 2023	17 Februari 2023
5	Persiapan Upcoming Cycle 2 dengan team finance dan team Internal (big batch project) & Mempelajari code di fitur Report & flow pengerjaan cycle 2 dan set up tools	18 Februari 2023	24 Februari 2023
6	Mengerjakan tugas Cycle 2 Fitur Reconciliation Report & Code review fixing Cycle 2: Finance Reconciliation Report	27 Februari 2023	10 April 2023
7	Mengerjakan tugas fixing bug productType empty & change etoll to emoney	10 April 2023	13 April 2023

8	Mengerjakan tugas Re-styling Maintenance mode menu	17 April 2023	26 April 2023
9	Persiapan Upcoming Cycle 3 (Small Batch Tasks)	26 April 2023	27 April 2023
10	Mengerjakan tugas adjustment Update menu CRUD untuk table mspulsa dan mspasca (product_category) & Code review fixing adjustment Update menu CRUD untuk table mspulsa dan mspasca (product_category)	2 Mei 2023	11 Mei 2023
11	Mengerjakan tugas Clear Cache di fitur RC Mapping & Code Review Fixing task Clear Cache di fitur RC Mapping	12 Mei 2023	23 Mei 2023
12	Mengerjakan tugas fixing bug apply_on di fitur Custom Pricelist Postpaid	23 Mei 2023	29 Mei 2023
13	Mengerjakan tugas menambahkan search user pada fitur Custom Pulsa Active Time	29 Mei 2023	31 Mei 2023
14	Mengerjakan tugas sensor kode voucher di fitur Update Report	30 Mei 2023	8 Juni 2023

15	Mengerjakan tugas Total Profit pada excel berbeda dengan Summary CMS	5 Juni 2023	13 Juni 2023
16	Mengerjakan tugas fixing bug untuk transaksi > H+7 pada fitur Update Report dan Set Status Transaction	15 Juni 2023	17 Juni 2023

Berikut merupakan uraian per penjelasan tugas yang dijelaskan dalam sub-bab:

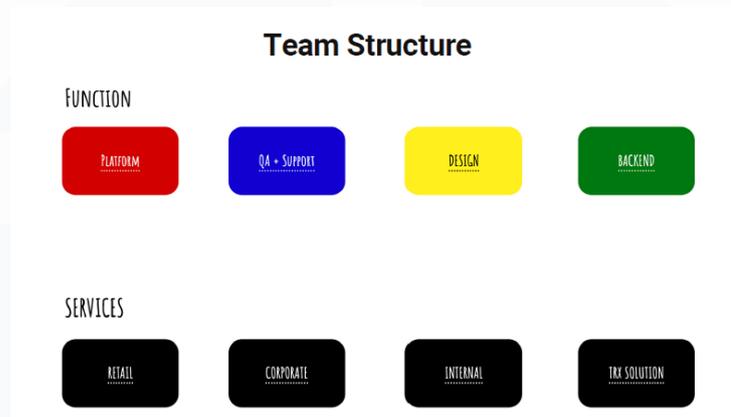
3.2.1 Onboarding Intern, pengenalan environment & team, set up tools, research project & Laravel.

3.2.1.1 Onboarding Intern

Pada tanggal 9 Januari 2023, peserta magang di PT IAK melakukan *Onboarding intern* di kantor IAK yang berlokasi di Alam Sutera, Tangerang. Kegiatan *Onboarding* menandakan hari pertama kegiatan magang dimulai dan bertujuan untuk memperkenalkan perusahaan IAK kepada para peserta magang dan membuat para peserta magang lebih cepat dalam beradaptasi dan berbaur di perusahaan, sehingga dapat mendukung produktivitas dan kesejahteraan karyawan. Sesi *Onboarding* dibawakan langsung oleh CEO PT IAK yaitu bapak Budiman Yungsen dan CTO PT IAK yaitu bapak Eddy Christiandy, serta diikuti oleh 3 peserta magang.

Bapak Budiman sebagai CEO menjelaskan banyak hal tentang apa itu perusahaan IAK, Visi dan Misi Perusahaan, sejarah dan perjalanan mendirikan perusahaan, serta culture dari perusahaan IAK. Sementara pak Eddy sebagai CTO banyak menjelaskan tentang teknologi yang digunakan perusahaan, *work-culture*, dan pembagian tim/ divisi peserta magang.

Sejak Januari 2023, pembagian divisi dan struktur karyawan IT dilakukan berdasarkan fungsi dan layanannya. Berikut merupakan pembagian divisi berdasarkan fungsi dan layanannya:



Gambar 3. 1 IT *Team Structure* PT IAK

- *Function:*
 - Platform: Team platform bertanggung jawab dalam memastikan layanan dan infrastruktur teknologi perusahaan berjalan dengan lancar, mendukung produktivitas *Engineer/* team lain, menjaga produk perusahaan dari resiko keamanan, menentukan parameter reliabilitas dan performance, menjaga dokumentasi, dan mendukung *Research & Development*.
 - QA + Support: Team QA + Support bertanggung jawab atas mengurangi faktor dan resiko dari *development*, melakukan *testing*, memastikan kualitas dari produk dan layanan layak bagi pelanggan, dan meningkatkan kepuasan dan keberhasilan produk untuk pelanggan dalam mengurangi kendala teknis.
 - Design: Team Design bertanggung jawab dalam merancang asset organisasi, sistem, dan desain

organisasi, Mengoptimasi aksesibilitas tampilan dan *client speed & performance*, menangani SEO, mengelola blog perusahaan, dan menangani *task & bug*.

- Backend: Team Backend bertanggung jawab dalam mengeksekusi code dalam bentuk *business logic* yang efektif dan dapat digunakan sebaik mungkin, mengimplementasi unit testing project, memastikan *Countinous Integration (CI)* dan *Continuous Deployment (CD)*, Mengoptimisasi *log, debugging, query, dan cache*, mengelola dokumentasi, dan mengerjakan *tasks & bug*.

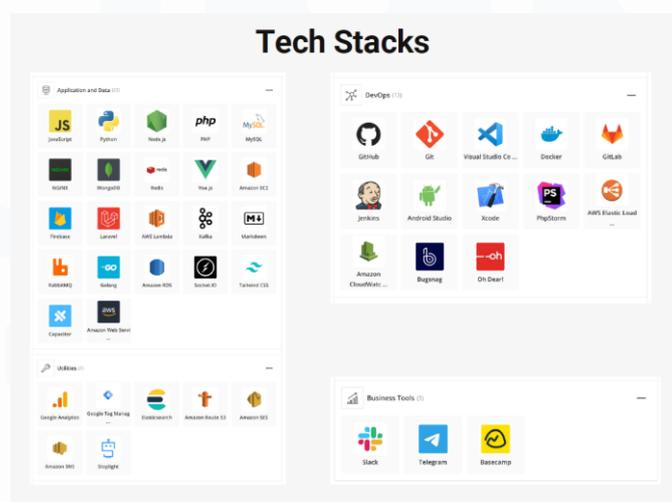
- *Services:*

- Retail: Team Retail bertanggung jawab dalam mengembangkan produk berupa layanan secara langsung pada *end user* (pelanggan) dan memastikan produk yang diciptakan sesuai dengan keinginan pelanggan. Kini team Retail di IAK bertanggung jawab atas produk IAK bernama Whitelabel.
- Corporate: Team Corporate bertanggung jawab dalam mengembangkan produk yang diciptakan untuk perusahaan yang melakukan kerjasama dengan perusahaan IAK (B2B).
- Internal: Team Internal bertanggung jawab dalam mengembangkan dan menjamin *tools & website* yang digunakan oleh karyawan IAK dalam menangani operasional bisnis berjalan dengan efisien dan efektif. Team Internal bertugas dalam

menangani *Content Management System* (CMS) perusahaan IAK dengan memfasilitasi *Finance*, *Customer Service*, *Marketing*, dan *Worker*.

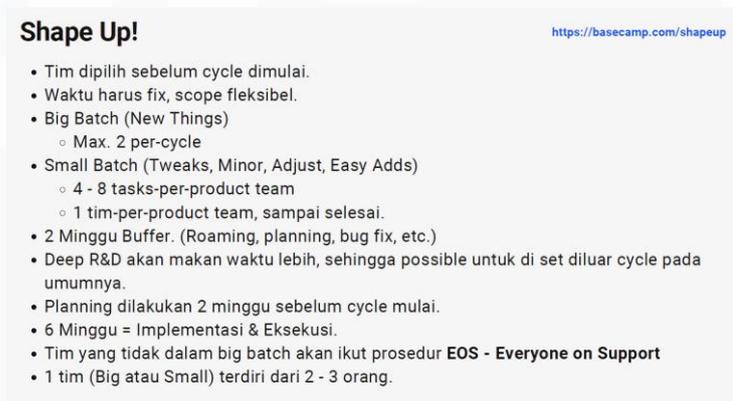
- Trx Solution: Team Trx Solution (*Transaction Solutions*) bertanggung jawab atas transaksi yang terjadi di produk mobilepulsa baik *prepaid* maupun *postpaid* dan memastikan transaksi berjalan secara valid, cepat, dan memberikan profitabilitas yang maksimal.

Pada sesi Onboarding pak Eddy memperkenalkan para peserta magang dengan team masing-masing dan memperkenalkan masing-masing *Product Owner* dari team tersebut. Peserta magang Katherine Allen Lius ditempatkan di team Internal sebagai *backend developer*. Selain itu, pak Eddy juga menjelaskan tentang Tech Stack dari perusahaan IAK yaitu kombinasi dari bahasa pemrograman, teknologi, *software*, *framework* dan *tools* yang digunakan dalam mendukung proses pembangunan *software* dan produk layanan IAK. Berikut merupakan *tech stacks* dari perusahaan IAK:



Gambar 3. 2 *Tech Stacks* IAK

Tidak hanya itu, pak Eddy juga menjelaskan tentang *work-culture* dari perusahaan IAK yang tersedia dalam dokumen *Shape Up* milik perusahaan IAK.



Gambar 3. 3 *Shape Up*

Proses *development* di perusahaan IAK dijalankan *per-cycle* (6 minggu kerja). Dalam satu *cycle*, anggota team dibagi menjadi 2 berdasarkan *scope/ target* pekerjaan pada *cycle* tersebut yaitu (*Big Batch dan Small Batch*). Anggota yang ditugaskan sebagai bagian dari *Big Batch* bertanggung jawab atas membangun pembaruan dalam produk, misalnya fitur baru yang ditargetkan untuk selesai dalam 1 *cycle* (6 minggu). Sementara anggota yang ditugaskan sebagai *Small Batch*, bertugas untuk mengerjakan tugas-tugas yang memakan waktu lebih singkat namun jumlahnya lebih banyak (4-8 *task per cycle*). Anggota yang dijelaskan sebagai *Small Batch* juga berperan sebagai EOS – *Everyone on Support* yaitu menangani *bug* atau kendala pada produk yang ditemukan selama berjalannya *cycle* dan bersifat membutuhkan tindakan dan perhatian yang segera. Berikut merupakan pembagian *target per cycle* pada peserta magang Katherine Allen Lius:

- *Cycle 1: Small Batch*
- *Cycle 2: Big Batch*

- *Cycle 3: Small Batch*

3.2.1.2 Perkenalan *environment & team*

Perkenalan *environment & team* di IAK dilakukan ketika peserta magang bekerja secara WFO di kantor. Hal ini dikarenakan jadwal tiap team yang berbeda hari WFO & WFH. Untuk team Internal, peserta magang melakukan perkenalan ketika team melakukan *weekly meeting* dan *update task*. Selain itu, perkenalan peserta magang terhadap seluruh anggota karyawan juga dilakukan ketika *Feast Meeting* perusahaan dilakukan yaitu tanggal 13 Januari 2023.

3.2.1.3 Set up *tools*

Di minggu pertama kegiatan magang, peserta magang diharuskan untuk melakukan instalasi dan persiapan tools yang akan digunakan selama proses pekerjaan. Berikut merupakan beberapa tools yang diinstalasi oleh peserta magang pada minggu pertama bekerja: Aktivasi e-mail perusahaan, Git, Gitlab, Github, instalasi PHP Storm sebagai *code editor*, *set-up* basecamp dan telegram, instalasi plugin & VPN perusahaan. Selain itu, peserta magang juga melakukan instalasi database SQL dan mongoDB milik perusahaan. Tool TablePlus juga digunakan untuk membantu *developer* dalam mengelola dan berinteraksi dengan *database*. Selain itu *software* Fork, digunakan untuk membantu proses kontrol terhadap versi proyek karena tersambung dengan akun Git.

3.2.1.4 *Research project & Laravel*

Sebelum diberikan tugas pertama oleh *Product Owner*, peserta magang diharuskan untuk melakukan *research* atau mempelajari sendiri terkait dengan proses bisnis perusahaan, project yang akan

dijalankan, *programming language*, *tools* dan *tech stacks* yang akan digunakan, serta *Framework* yang digunakan pada *project CMS* yaitu Laravel. Untuk mendukung proses pembelajaran tersebut, perusahaan juga menyediakan *resource* pembelajaran terkait dengan bahasa pemrograman PHP, Javascript, Laravel, dan berbagai *tech stacks* lainnya melalui kelas online berbayar (*courses*), yaitu Laracast.com, Wes Bos, Spatie, dan berbagai sumber belajar lainnya.

3.2.1.5 Clone project CMS

Sebagai salah satu *developer* yang akan bekerja dengan CMS, peserta magang diharuskan untuk melakukan *clone project* atau membuat salinan identik dari proyek yang sudah ada dalam sistem pengendalian versi yaitu Gitlab dan Github. Tindakan ini dilakukan untuk memastikan bahwa *developer* dapat memperoleh Salinan lengkap dari proyek yang sudah ada, meliputi: file, folder, riwayat revisi, dan segala informasi yang tersedia di proyek agar *developer* dapat mengakses, melakukan modifikasi, dan berkontribusi dalam perubahan proyek tanpa merubah/mengganggu proyek yang sudah ada sebelumnya.

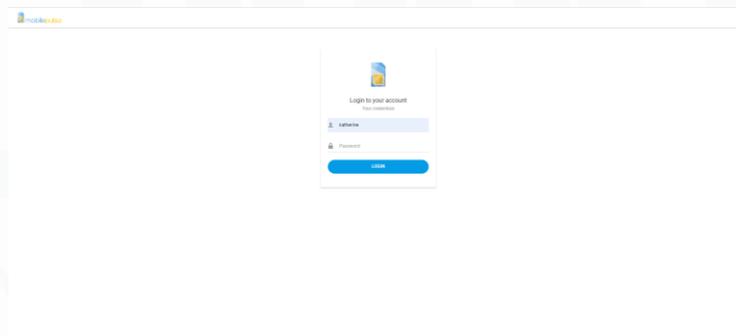
Proses ini dilakukan dengan perintah '*git clone*' di terminal (Git CMD), melakukan penyesuaian file *.env*, melakukan run '*composer install*', run '*php artisan tinker*', dan run '*php artisan serve*' untuk memulai server.

3.2.2 Mempelajari code CMS, Mengerjakan task fitur Lazada Email Table Prepaid & Lazada Email Table Postpaid, Code review fixing fitur Lazada Email Table Prepaid & Lazada Email Table Postpaid

3.2.2.1 Mempelajari code CMS

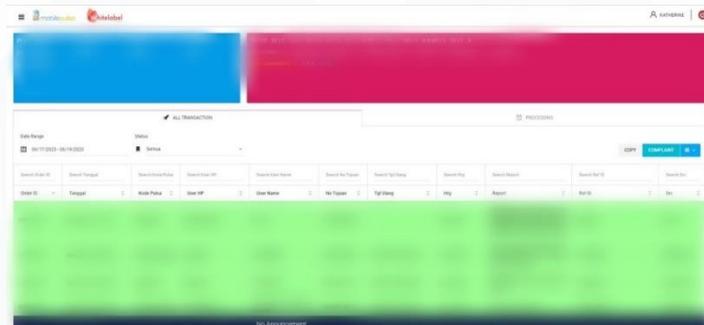
CMS adalah singkatan dari *Content Management System*, yang merupakan sebuah *website* yang digunakan oleh IAK untuk keperluan internal. Tujuan utama CMS adalah untuk membantu pekerjaan dan kolaborasi antar karyawan dalam perusahaan terkait dengan transaksi yang sedang berlangsung. Di IAK, terdapat beberapa pengguna CMS, seperti *Customer Service (CS)*, *Accounting*, *Marketing*, *IT*, dan pimpinan perusahaan. Penerapan CMS di IAK memiliki beberapa manfaat yang penting. Misalnya, CMS membantu CS dalam mengakses informasi pelanggan dengan mudah saat mereka menghadapi kendala. Selain itu, CMS memungkinkan CS untuk melakukan perubahan informasi secara *real-time* dan melacak riwayat transaksi pelanggan. CMS juga berperan dalam pembuatan laporan keuangan, perhitungan pajak, analisis data transaksi, serta menawarkan berbagai fungsi lainnya yang terus berkembang untuk mendukung operasional bisnis perusahaan secara keseluruhan.

Berikut merupakan tampilan dari halaman login dari *website* CMS yang digunakan untuk melakukan verifikasi identitas pengguna, pengelolaan akses berdasarkan *role user*, dan menjaga keamanan sistem *website* CMS



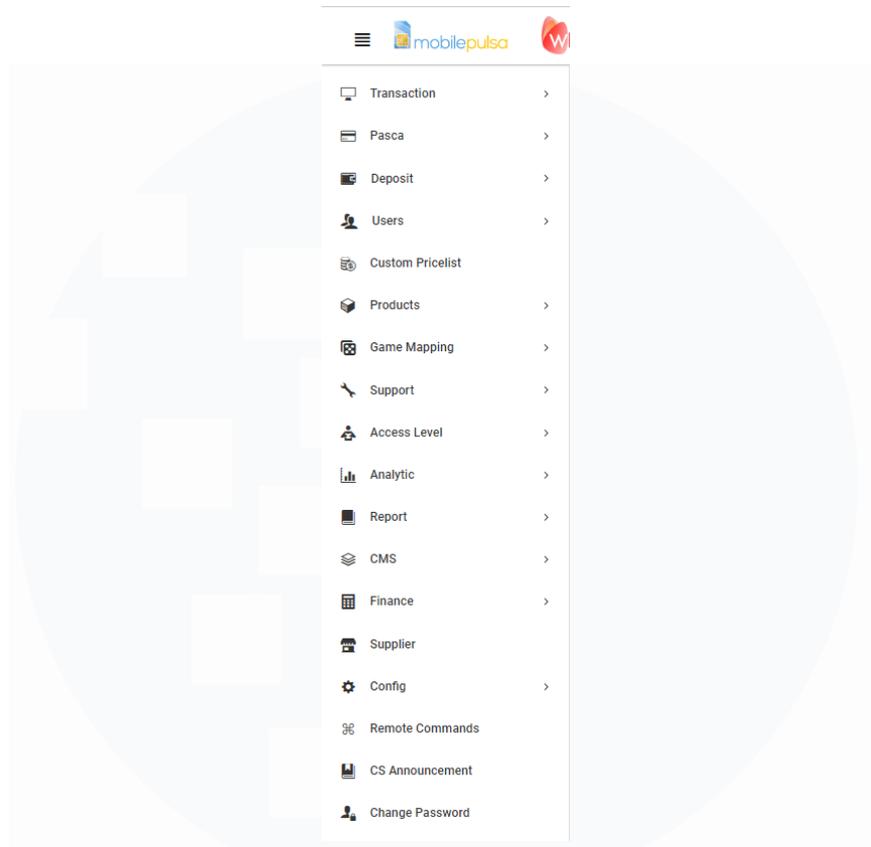
Gambar 3. 4 *Login Page* CMS

Berikut merupakan tampilan *homepage* dari *website* CMS, halaman ini berguna untuk menampilkan *dashboard*, transaksi yang masuk secara *real-time*, dan menampilkan *summary* dari transaksi.



Gambar 3. 5 *Homepage* CMS

Berikut merupakan tampilan *side bar* dari *website* CMS yang terletak di sisi kiri layar yang bermanfaat untuk pengguna ketika melakukan navigasi menu/fungsi yang tersedia di CMS. Menu/fungsi di CMS juga dikelompokan sesuai dengan kategorinya masing-masing, seperti: *Transaction, Pasca, Deposit, Users, Custom Pricelist, Products, Game Mapping, Support, Access Level, Analytic, Report, CMS, Finance, Supplier, Config, Remote Commands, CS Announcement, dan Change Password.*



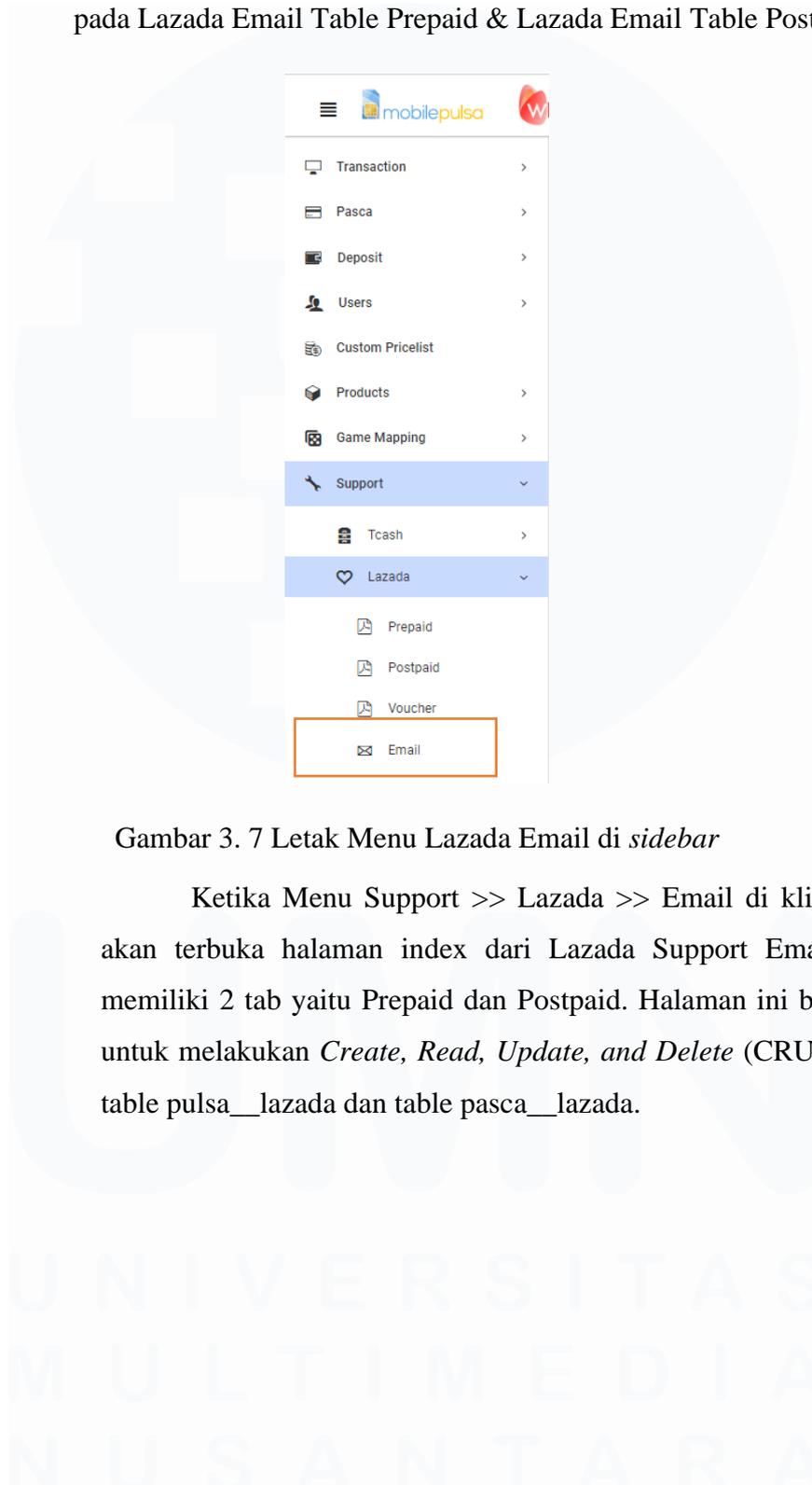
Gambar 3. 6 *Sidebar Menu CMS*

Pada saat minggu pertama bekerja, *supervisor* peserta magang memberi perintah untuk mempelajari *website CMS* secara umum dan menyeluruh untuk membiasakan diri terhadap CMS dan memahami alur dari fungsi *website* dan code CMS yang sudah ada sebagai persiapan sebelum mendapatkan tugas pertama yang berkaitan dengan CMS.

3.2.2.2 Mengerjakan *task* fitur Lazada Email Table Prepaid & Lazada Email Table Postpaid

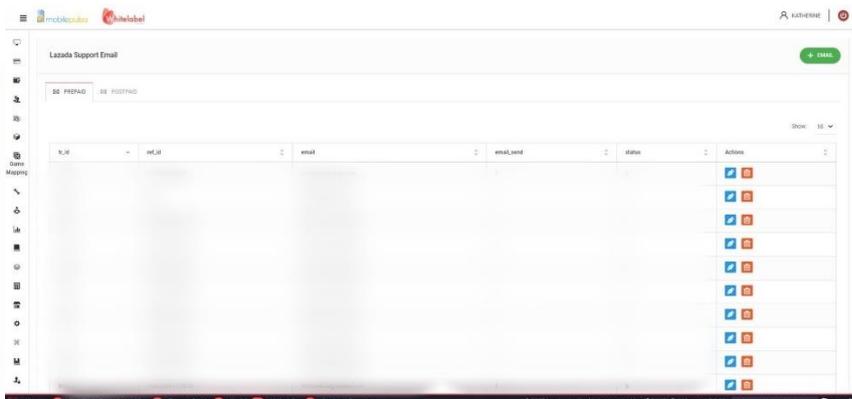
Tugas pertama yang diberikan oleh supervisor kepada peserta magang adalah membuat menu baru/ fitur untuk melakukan CRUD pada Lazada Email Table Prepaid & Lazada Email Table Postpaid. Menu ini terletak pada menu Support >> Lazada >> Email. Berikut

merupakan hasil kerja peserta magang dalam membangun fitur CRUD pada Lazada Email Table Prepaid & Lazada Email Table Postpaid.



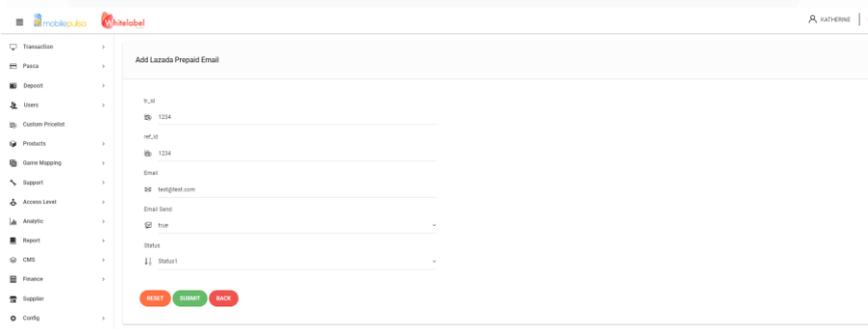
Gambar 3. 7 Letak Menu Lazada Email di *sidebar*

Ketika Menu Support >> Lazada >> Email di klik, maka akan terbuka halaman index dari Lazada Support Email yang memiliki 2 tab yaitu Prepaid dan Postpaid. Halaman ini berfungsi untuk melakukan *Create, Read, Update, and Delete* (CRUD) pada table pulsa__lazada dan table pasca__lazada.



Gambar 3. 8 Halaman *Index* Lazada Email

Pada tab prepaid, Ketika tombol ‘ + EMAIL’ pada kanan atas layar di klik, maka halaman akan berpindah ke halaman untuk menambahkan *row* baru pada Lazada Email Table prepaid.



Gambar 3. 9 Halaman *Add* Lazada Email Table Prepaid

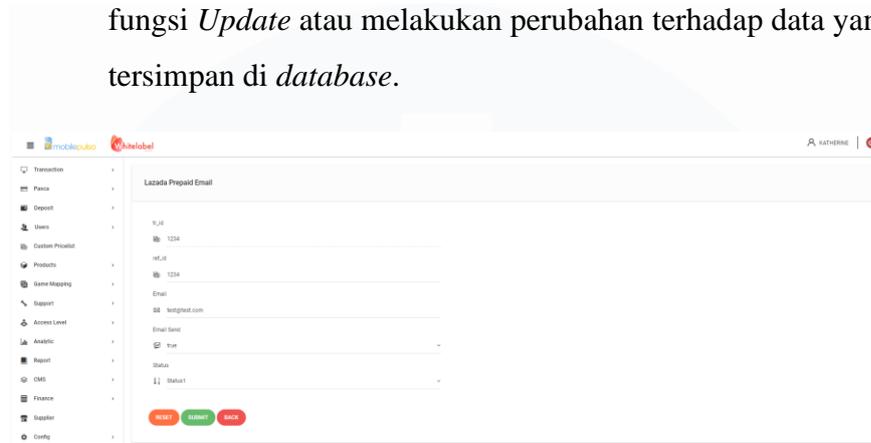
Ketika tombol ‘SUMBIT’ di klik, maka data yang telah dimasukan user akan tersimpan dalam data base dan ditampilkan pada table. Selain itu, website akan menampilkan handle notifikasi yang menandakan data berhasil tersimpan dalam database.



Gambar 3. 10 *Handle Success Add* Lazada Email Table Prepaid

Ketika tombol berwarna biru ‘*edit*’ dengan logo pena yang terletak pada kolom ‘*action*’ pada halaman *index* Lazada *Support*

Email (terletak di tiap *row* data) di klik, maka *user* dapat melakukan fungsi *Update* atau melakukan perubahan terhadap data yang sudah tersimpan di *database*.



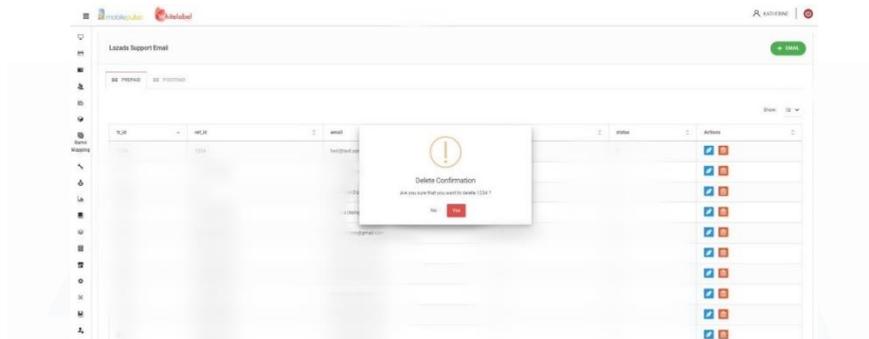
Gambar 3. 11 Halaman *Edit* Lazada Email Table Prepaid

Ketika tombol '*SUBMIT*' di klik, maka perubahan terhadap data akan tersimpan pada *database*. Selain itu, *website* akan menampilkan *handle* notifikasi yang menandakan data berhasil diperbarui dalam *database*.



Gambar 3. 12 *Handle Success Update* Lazada Email Table Prepaid

Ketika tombol berwarna merah '*delete*' dengan logo tempat sampah yang terletak pada kolom '*action*' pada halaman *index* Lazada Support Email (terletak di tiap *row* data) di klik, maka user dapat melakukan fungsi *Delete* atau menghapus data yang sudah tersimpan di *database*. Ketika tombol '*delete*' di klik, maka akan muncul *pop-up confirmation box* untuk mengkonfirmasi tindakan user terkait penghapusan data.



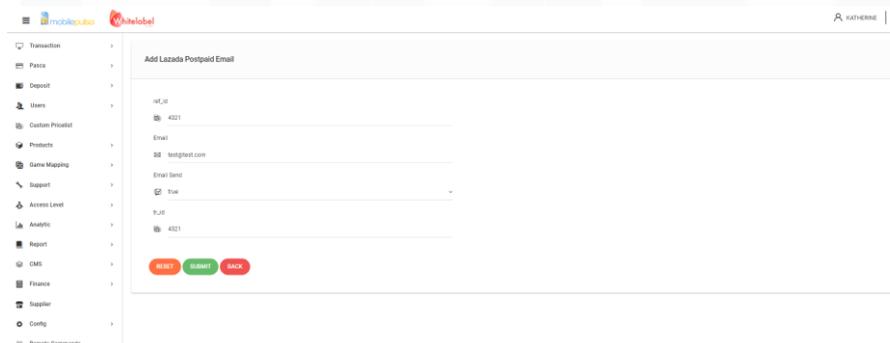
Gambar 3. 13 *Pop-up Confirmation Box* ketika *delete* Lazada Email Prepaid

Ketika tombol ‘YES’ di klik, maka penghapusan data akan tersimpan pada *database*. Selain itu, *website* akan menampilkan *handle* notifikasi yang menandakan data berhasil dihapus dari dalam *database*.



Gambar 3. 14 *Handle Success Delete* Lazada Email Table Prepaid

Sementara pada tab *postpaid* di halaman indeks Lazada Support Email, Ketika tombol ‘+ EMAIL’ pada kanan atas layar di klik, maka halaman akan berpindah ke halaman untuk menambahkan *row* baru pada Lazada Email Table Postpaid.



Gambar 3. 15 Halaman *Add* Lazada Email Table Postpaid

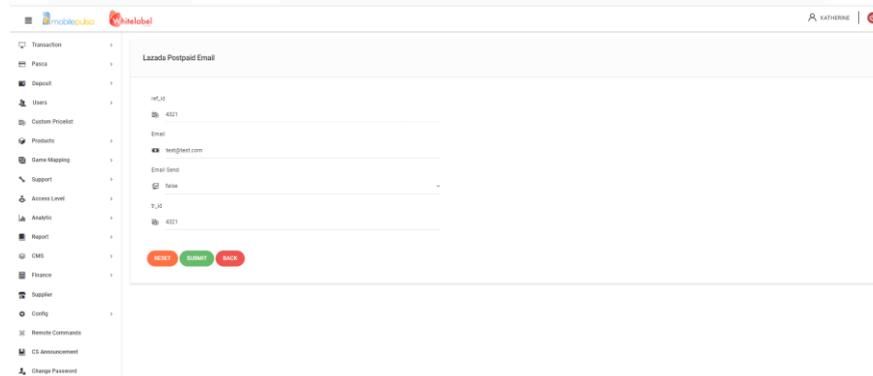
Ketika tombol ‘SUMBIT’ di klik, maka data yang telah dimasukan user akan tersimpan dalam data base dan ditampilkan

pada table. Selain itu, *website* akan menampilkan *handle* notifikasi yang menandakan data berhasil tersimpan dalam *database*.



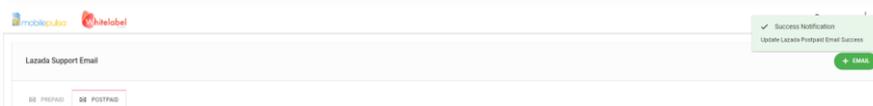
Gambar 3. 16 *Handle Success Add Lazada Email Table Postpaid*

Ketika tombol berwarna biru '*edit*' dengan logo pena yang terletak pada kolom '*action*' pada halaman *index* Lazada Support Email (terletak di tiap *row* data) di klik, maka *user* dapat melakukan fungsi *Update* atau melakukan perubahan terhadap data yang sudah tersimpan di *database*.



Gambar 3. 17 Halaman *Edit Lazada Email Table Postpaid*

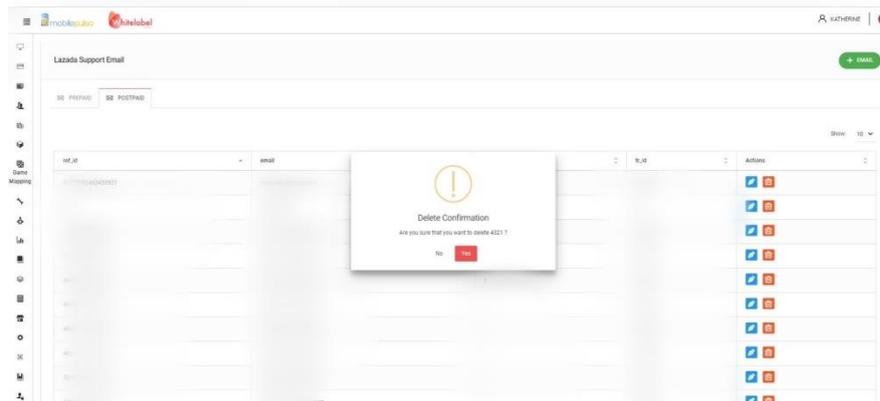
Ketika tombol '*SUBMIT*' di klik, maka perubahan terhadap data akan tersimpan pada *database*. Selain itu, *website* akan menampilkan *handle* notifikasi yang menandakan data berhasil diperbarui dalam *database*.



Gambar 3. 18 *Handle Success Update Lazada Email Table Postpaid*

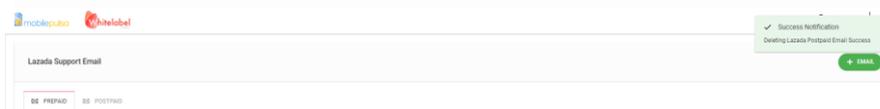
Ketika tombol berwarna merah '*delete*' dengan logo tempat sampah yang terletak pada kolom '*action*' pada halaman *index* Lazada

Support Email (terletak di tiap *row* data) di klik, maka user dapat melakukan fungsi *Delete* atau menghapus data yang sudah tersimpan di database. Ketika tombol '*delete*' di klik, maka akan muncul *pop-up confirmation box* untuk mengkonfirmasi tindakan *user* terkait penghapusan data.



Gambar 3. 19 *Pop-up Confirmation Box* ketika *delete* Lazada Email Postpaid

Ketika tombol '*YES*' di klik, maka penghapusan data akan tersimpan pada database. Selain itu, *website* akan menampilkan *handle* notifikasi yang menandakan data berhasil dihapus dari dalam *database*.



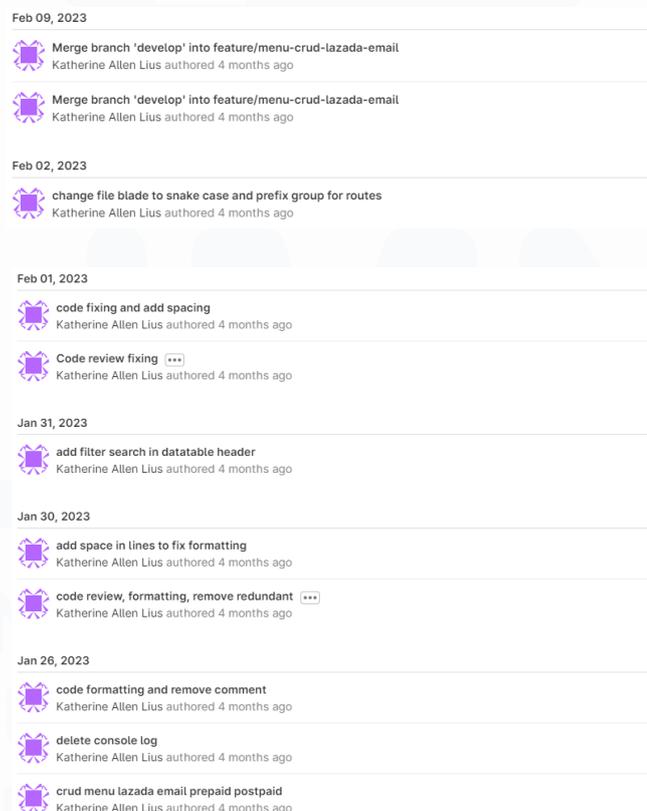
Gambar 3. 20 *Handle Success Delete* Lazada Email Table Postpaid

Ketika semua fungsi CRUD pada Lazada Email Table Prepaid & Lazada Email Table Postpaid serta tampilan halaman-halaman yang berkaitan sudah berjalan sesuai dengan semestinya, peserta magang pun melakukan *Merge Request* melalui Gitlab untuk meminta peninjauan kode kepada *supervisor (Product Owner)*.

3.2.2.3 *Code review fixing* fitur Lazada Email Table Prepaid & Lazada Email Table Postpaid

Setelah melakukan *Merge Request*, *Supervisor* akan melakukan pengecekan terkait code yang sudah dibuat dan memastikan bahwa code yang dibuat berkualitas, konsisten, dan berjalan sesuai dengan keinginan perusahaan.

Setelah *supervisor* melakukan *code review*, maka *supervisor* akan menandai line code yang masih salah atau memerlukan revisi dengan fitur *comment* pada Gitlab. Kemudian, peserta magang akan memperbaiki code sesuai dengan permintaan *supervisor* dan melakukan '*commit*' untuk mengumpulkan code revisi dan meminta evaluasi Kembali terkait code yang sudah di revisi. Proses tersebut dilakukan terus menerus hingga semua code sudah berkualitas, konsisten, dan sesuai. Berikut merupakan beberapa *commit* yang dilakukan oleh peserta magang ketika mengerjakan tugas CRUD pada Lazada Email Table Prepaid & Lazada Email Table Postpaid.



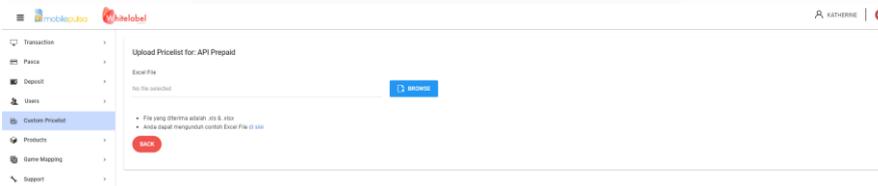
Gambar 3. 21 Hasil *Commit* Fitur Lazada Email Prepaid & Postpaid

3.2.3 Mengerjakan tugas *fixing bug* validasi `product_code` ketika *upload file excel* di fitur *Custom pricelist & Code review fixing bug* Validasi `product_code` ketika *upload file excel* di fitur *Custom pricelist*

3.2.3.1 Mengerjakan tugas *fixing bug* validasi `product_code` ketika *upload file excel* di fitur *Custom pricelist*

Tugas kedua yang diberikan kepada peserta magang adalah memperbaiki *bug* yang muncul ketika user ingin menggunakan fitur *Custom Pricelist* pada API Prepaid melalui fungsi *upload custom pricelist* menggunakan file excel, namun `product_code` yang tersimpan di *database* tidak sesuai dengan `product_code` yang di upload melalui excel.

Berikut merupakan tampilan halaman '*Upload Custom Pricelist API Prepaid*' melalui Excel.



Gambar 3. 22 Halaman *Upload Custom Pricelist API Prepaid*

Berikut merupakan tampilan file *template excel* yang sudah disediakan melalui *website CMS*. Untuk menggunakan fitur ini, User dapat melakukan download *file template custom pricelist excel*, kemudian mengisi data sesuai dengan kolom yang sudah ada, lalu melakukan *upload* melalui CMS untuk menyimpan data *custom pricelist* ke dalam *database*.

No	Pulsa Code	User HP	H2H	Fix Price	Pulsa Status
1	test	1234		1000	1
2	test	1234		1000	1
3	test	1234		1000	1
4	test	1234		1.000	1

Gambar 3. 23 Template File Excel untuk *Upload Custom Pricelist API Prepaid*

Untuk mengerjakan *task* ini, pertama-tama peserta magang melakukan *debugging* untuk mengetahui alasan dari terjadinya *bug*. Setelah dilakukan *debugging*, peserta magang menemukan sumber dari masalah yaitu karena perbedaan kapital pada *pulsa_code* yang menyebabkan sistem tidak dapat membaca/mendeteksi *product_code* yang di *upload* melalui excel, contohnya apabila *product_code* pada file excel berisi test123GB sementara *product_code* pada database merupakan test123gb sistem tidak akan berhasil membaca *product_code* tersebut dan menganggap *product_code* tidak ada dalam database.

Untuk mengatasi masalah ini, peserta magang memperbaiki code yang sudah ada terutama pada bagian *query* yang membaca data dari file excel yang diunggah *user* agar menjadi “*case insensitive*” yaitu agar sistem tidak mampu membedakan antara huruf kapital dan huruf non-kapital ketika membaca data (dianggap sama). Hal ini bertujuan agar ketika user CMS melakukan *upload custom pricelist* dengan *product_code* yang berbeda kapital dikemudian hari, maka akan tetap terbaca oleh sistem dan dapat tersimpan pada *database*.

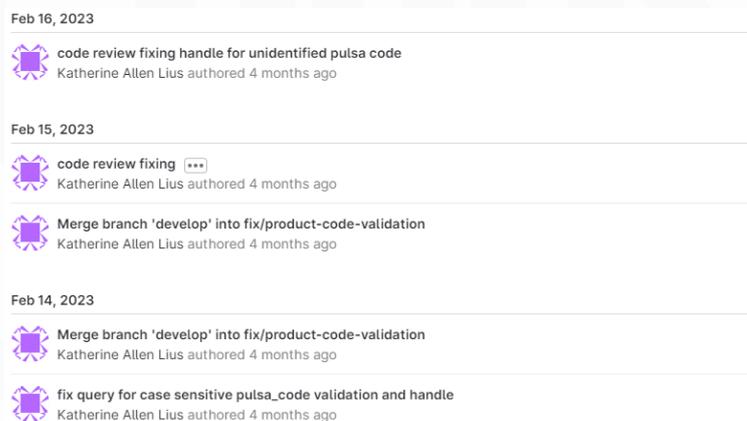
Ketika semua fungsi dan *bug* validasi *product_code* berhasil diperbaiki dan sudah berjalan sesuai dengan semestinya, peserta

magang pun melakukan *Merge Request* melalui Gitlab untuk meminta peninjauan kode kepada *supervisor (Product Owner)*.

3.2.3.2 *Code review fixing bug Validasi product_code ketika upload file excel di fitur Custom pricelist*

Setelah melakukan *Merge Request*, *Supervisor* akan melakukan pengecekan terkait *code* yang sudah dibuat dan memastikan bahwa *code* yang dibuat berkualitas, konsisten, dan berjalan sesuai dengan keinginan perusahaan.

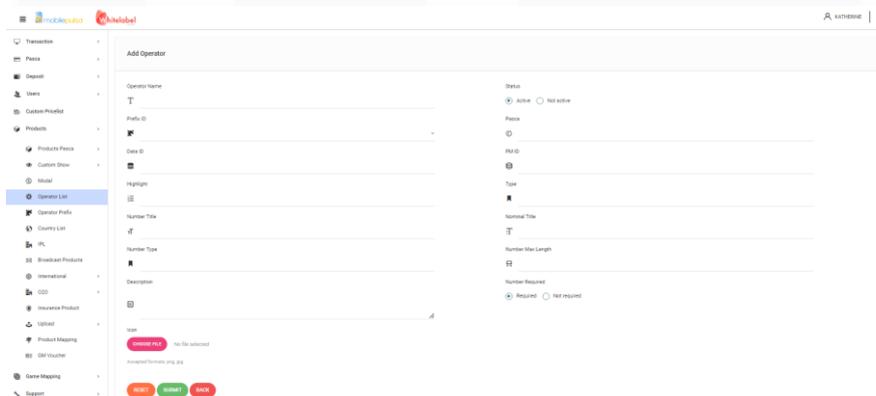
Setelah supervisor melakukan *code review*, maka *supervisor* akan menandai *line code* yang masih salah atau memerlukan revisi dengan fitur *comment* pada Gitlab. Kemudian, peserta magang akan memperbaiki *code* sesuai dengan permintaan *supervisor* dan melakukan '*commit*' untuk mengumpulkan *code* revisi dan meminta evaluasi kembali terkait *code* yang sudah di revisi. Proses tersebut dilakukan terus menerus hingga semua *code* sudah berkualitas, konsisten, dan sesuai. Berikut merupakan beberapa *commit* yang dilakukan oleh peserta magang ketika mengerjakan tugas bug validasi *product_code* ketika *upload file excel* di fitur *Custom pricelist*.



Gambar 3. 24 Hasil *Commit fixing bug* validasi *product_code*

3.2.4 Mengerjakan tugas *fixing bug Number Required false value* di fitur Add Operator

Tugas ketiga yang dikerjakan oleh peserta magang adalah bug yang muncul pada Menu Products >> Operator List, yaitu ketika user ingin menambahkan operator pada CMS dan mengisi data `number_required`(YES) kemudian klik tombol 'SUBMIT', data yang tersimpan pada atribut `number_required` berubah menjadi 'NO'.



Gambar 3. 25 Halaman *Add Operator*

Untuk mengerjakan tugas ini, peserta magang melakukan *debugging* terkait nilai yang disimpan oleh sistem dalam fungsi CREATE dan ADD Operator. Setelah dilakukan *debugging*, ditemukan bahwa terdapat kesalahan penyimpanan *value* pada variabel `number_required`.

Ketika semua fungsi pada *Add Operator* sudah berjalan sesuai dengan semestinya, peserta magang pun melakukan *Merge Request* melalui Gitlab untuk meminta peninjauan kode kepada *supervisor (Product Owner)*.

Setelah melakukan *Merge Request*, *Supervisor* akan melakukan pengecekan terkait code yang sudah dibuat dan memastikan bahwa code yang dibuat berkualitas, konsisten, dan berjalan sesuai dengan keinginan perusahaan.

3.2.5 Persiapan *Upcoming Cycle 2* dengan *team finance* dan *team Internal (big batch project)* & Mempelajari code di fitur *Report & flow* pengerjaan *cycle 2* dan *set up tools*

3.2.5.1 Persiapan *Upcoming Cycle 2* dengan *team finance* dan *team Internal (big batch project)*

Setelah Cycle 1 selesai, *Supervisor* peserta magang mulai merencanakan tugas-tugas dan pembagian *role* anggota team Internal untuk Cycle 2. Berdasarkan keputusan atasan, peserta magang diberikan tugas sebagai *Big Batch* yang akan membantu membuat fitur yang akan membantu karyawan Finance dalam menjalankan pekerjaannya di CMS agar lebih cepat dalam membuat laporan rekonsiliasi yang dibutuhkan setiap akhir bulan untuk menyelaraskan data transaksi dari sumber yang berbeda (baik untuk klien dan untuk pihak internal).

Tahap persiapan ini dilakukan dengan *meeting team Internal Big Batch* dengan *team Finance*, serta CEO dan CTO. Pada *meeting* tersebut, dibahas mengenai pembaruan CMS apa yang dibutuhkan oleh *team Finance* beserta detail dan eksekusinya. Setelah itu, pembahasan mengenai alur pengerjaan secara teknikal dan pembagian *scope* juga dibahas untuk mematangkan rencana terkait dengan eksekusi Cycle 2.

Team Finance mengungkapkan kendala mereka ketika melakukan *generate report* dengan fitur CMS yang sudah ada, kemudian dimodifikasi secara manual setiap akhir bulannya. Kegiatan *repetitive* ini memakan waktu yang cukup banyak dan membuat pekerjaan *team finance* di akhir bulan menumpuk. Sehingga dengan adanya fitur ini, diharapkan dapat membantu membuat pekerjaan mereka menjadi lebih efektif dan efisien. Tidak hanya itu, *team finance* juga meminta tambahan fungsi ketika fungsi *Generate Reconciliation Report*, dijalankan maka akan terhitung rangkuman (*summary*)

perhitungan COGS (*Cost of Goods Sold*), DPP (Dasar Pengenaan Pajak), PPn (Pajak Pertambahan Nilai), dan PPh22 (Pajak Penghasilan Pasal 22).

Peserta magang Katherine Allen Lius pada Cycle ke-2 ditugaskan untuk mengerjakan fitur *Generate Reconciliation Report* yang harus diselesaikan dalam 6 minggu (*develop code*) dan 2 minggu (*buffer week/ fixing* dan *testing*). Tugas ini dikerjakan secara individu namun peserta magang boleh melakukan konsultasi kepada *supervisor* maupun rekan kerja lainnya apabila menghadapi hambatan.

3.2.5.2 Mempelajari code di fitur *Report & flow* pengerjaan cycle 2 dan set up tools

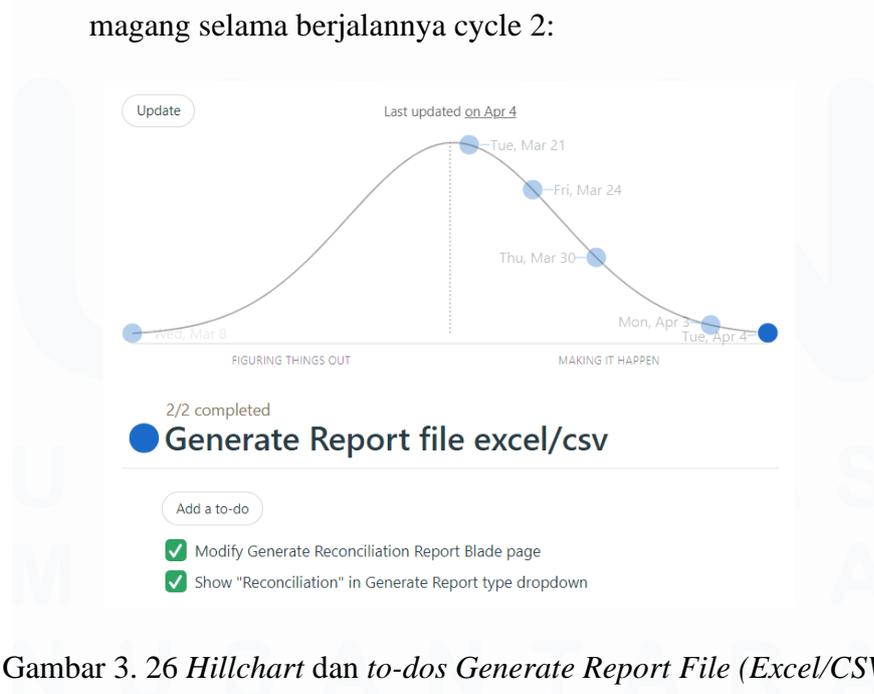
Setelah meeting dengan team Finance dilakukan, team Internal mulai menyusun dan mempersiapkan diri dengan melakukan pembelajaran terhadap code yang nanti akan dikembangkan untuk membuat fitur *Generate Reconciliation Report*. Di CMS sendiri, sudah ada fitur generate berbagai report seperti: General dan report lainnya yang berkaitan dengan klien. Untuk itu, dalam membangun code untuk fitur *Generate Reconciliation Report* peserta magang dapat melakukan pengembangan dan modifikasi dari code yang sudah ada namun diiringi dengan penyesuaian yang cukup banyak, mengingat dibutuhkannya *Summary Reconciliation Report* yang dapat melakukan perhitungan secara otomatis.

Selain itu, pada tahap persiapan ini peserta magang juga banyak melakukan diskusi dengan team *Finance* secara personal dan menyiapkan *tools* yang dibutuhkan seperti *set up database production*, dan lain-lain.

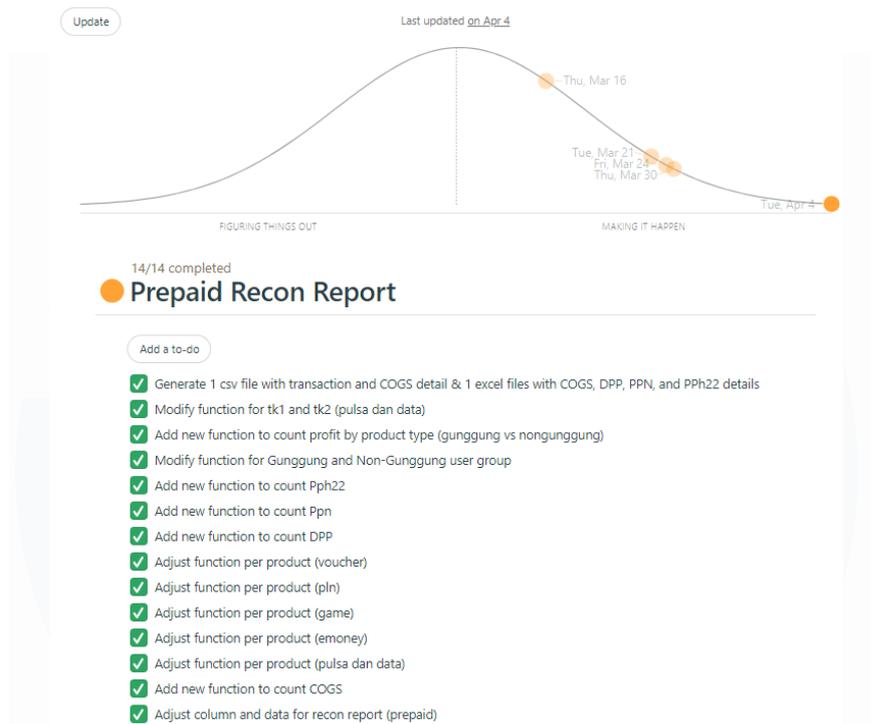
3.2.6 Mengerjakan tugas Cycle 2 Fitur Reconciliation Report & Code review fixing Cycle 2: Finance Reconciliation Report

Berdasarkan hasil meeting bersama finance, kendala dari proses pengerjaan *Cycle 2 Generate Reconciliation Report* terletak pada fitur rangkuman (*summary*) dari *Reconciliation Report* tersebut, karena perhitungan COGS, DPP, PPn, dan PPh22 untuk tiap produk berbeda-beda (pulsa dan data, voucher game, e-money, pln, dan voucher), serta tipe transaksi juga mempengaruhi perbedaan perhitungan (prabayar atau pascabayar). Selain itu, peran IAK untuk tiap operator dalam kategori produk pulsa dan data juga berbeda untuk tiap operatornya, sehingga ketika proses *coding* perlu dilakukan banyak penyesuaian untuk tiap operator.

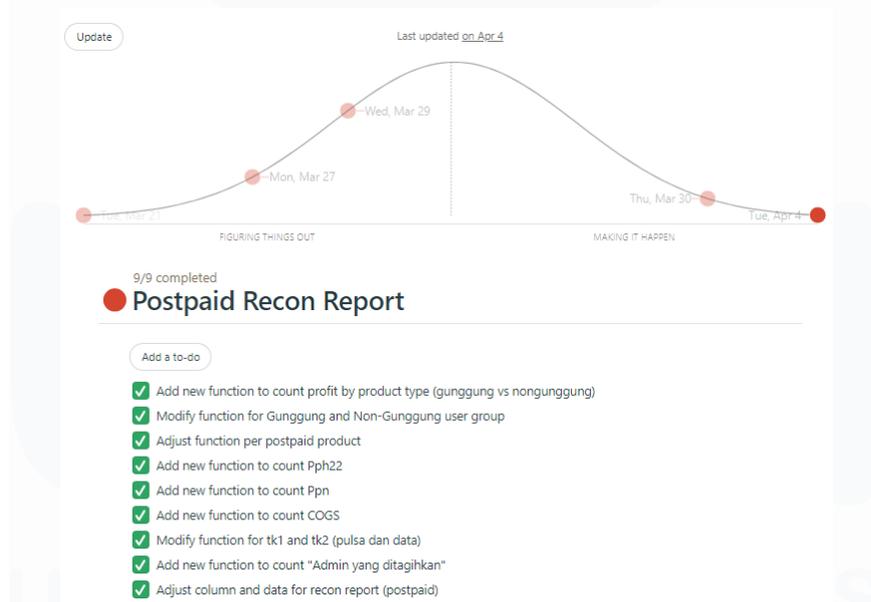
Sebelum memulai proses *developing*, peserta magang diminta melakukan *breakdown to-dos* yang dibutuhkan selama proses *developing*, dan melakukan *update hillchart* melalui aplikasi basecamp untuk menunjukkan *progress* dari project yang dijalankan. Berikut merupakan *Hill Chart* dan *to-dos* yang dijabarkan oleh peserta magang selama berjalannya cycle 2:



Gambar 3. 26 Hillchart dan to-dos Generate Report File (Excel/CSV)



Gambar 3. 27 Hillchart dan to-dos Prepaid Recon Report

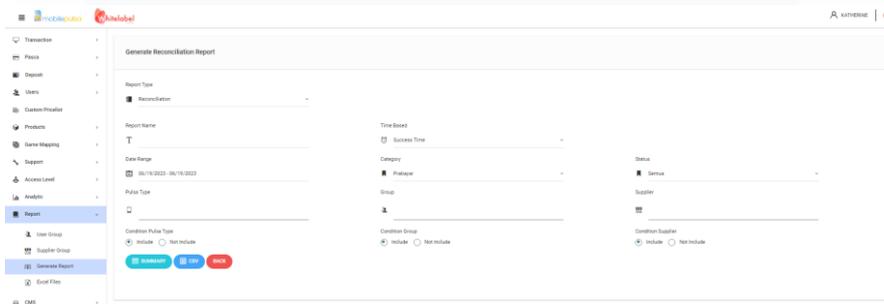


Gambar 3. 28 Hillchart dan to-dos Postpaid Recon Report

Hillchart digunakan untuk memvisualisasikan *progress* dalam mencapai target yang sudah ditentukan, sementara *to-dos* berguna untuk membantu peserta magang dalam mengetahui apa

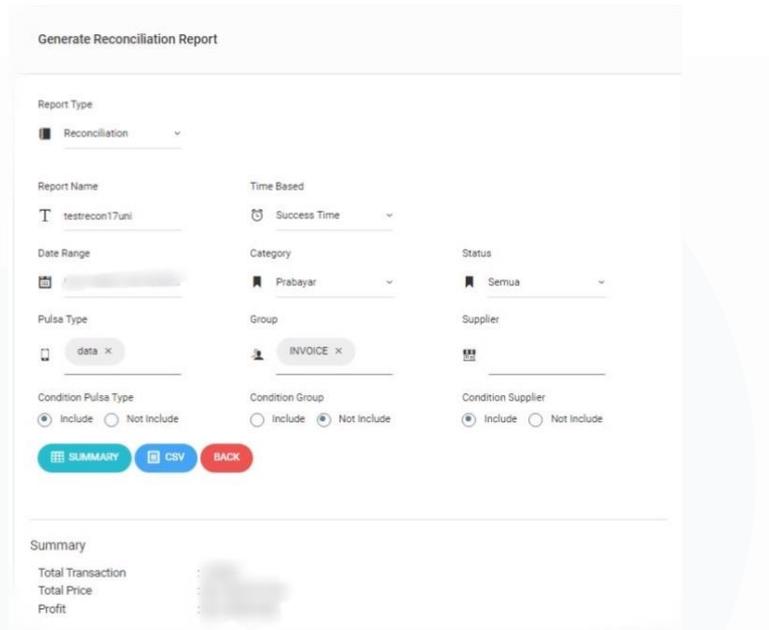
saja yang perlu dibuat dalam proses mencapai target, memudahkan manajemen pembagian waktu dalam menjalankan tugas, dan memastikan bahwa tidak ada detail yang terlewat ketika proses *developing code*.

Ketika peserta magang sudah selesai melakukan *breakdown to-do* tentang apa saja yang akan dikerjakan dan sudah merasa siap untuk melakukan coding terkait fitur *Generate Reconciliation Report*. Maka proses pembangunan code dimulai. Fitur ini terletak di Menu Report >> Generate Report (Report Type = Reconciliation). Berikut merupakan tampilan dan fitur *Generate Reconciliation Report* yang dibuat oleh peserta magang sesuai dengan ketentuan yang diinginkan oleh team finance.



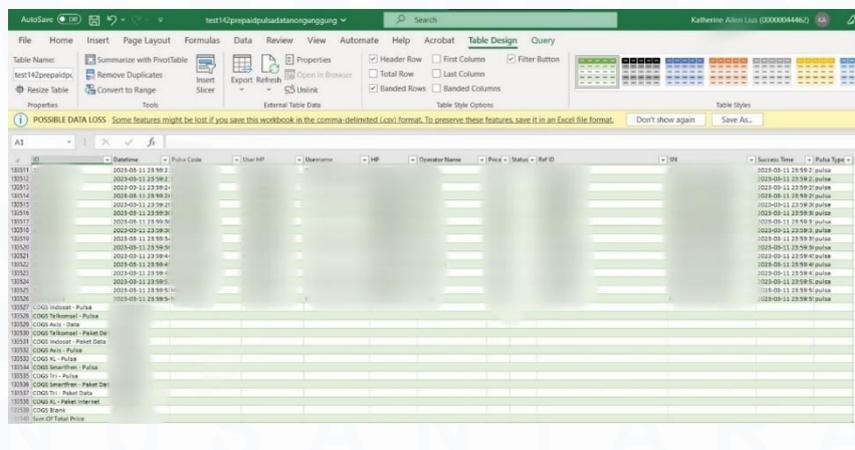
Gambar 3. 29 Halaman *Generate Reconciliation Report*

Ketika semua *field* sudah terisi, dan *user* melakukan klik pada tombol 'SUMMARY', maka akan muncul informasi terkait rangkuman dari *Reconciliation Report* yang akan di *generate*. Informasi tersebut terkait dengan *total transaction*, *total price*, dan *profit*.



Gambar 3. 30 Contoh *Testing Generate Reconciliation Report*

Sementara, ketika *user* melakukan klik tombol ‘CSV’ maka proses *Generate Reconciliation Report* akan dijalankan. Ketika proses *generate* sudah berhasil dilakukan, maka *computer user* akan melakukan *download* 2 file hasil *generate* dari fungsi tersebut, yaitu: file CSV berisi seluruh data transaksi yang tercatat beserta rangkuman perhitungan COGS dan *total price*, sementara file Excel berisi rangkuman dari transaksi yaitu COGS, DPP, PPN, PPH22, dan *total price*.



Gambar 3. 31 Hasil *Generate Reconciliation Report File CSV*

1	03/11/2023 - 03/12/2023						
2	Partner Nai Provider/Product	Partner as Dist Tk	PPH 22	COGS	DPP	PPN	PPH 22
3	PT Indobest Telkomsel - Pulsa						
4	Axis - Data						
5	Telkomsel - Paket Data						
6	Indosat - Paket Data						
7	Axis - Pulsa						
8	XL - Pulsa						
9	XL - Paket Internet						
10	Total						
11	Indosat - Pulsa						
12	Smartfren - Pulsa						
13	Tri - Pulsa						
14	Smartfren - Paket Data						
15	Tri - Paket Data						
16	Total						
17	Grand Total						
18							
19							

Gambar 3. 32 Hasil *Generate Reconciliation Report File Excel*

Selain itu, ketika file berhasil di *generate* maka akan muncul *handle* yang menandakan keberhasilan *Generate Reconciliation Report*.



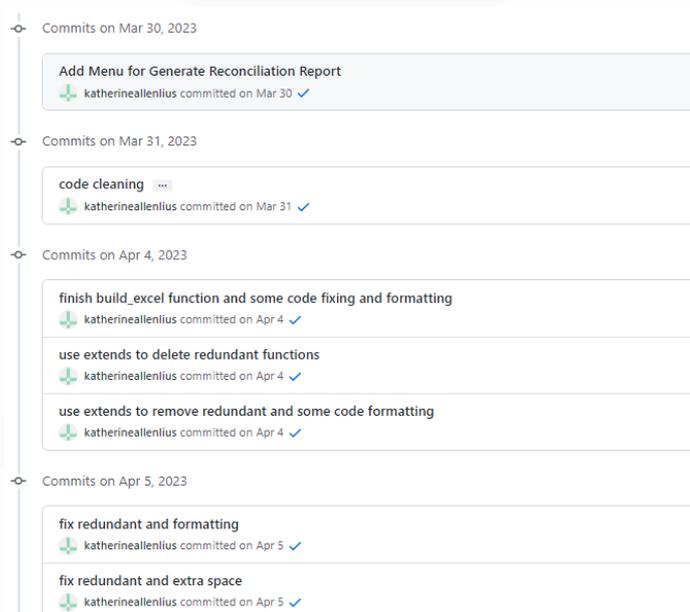
Gambar 3. 33 *Handle Generating Reconciliation Report Excel & CSV*

Ketika semua *to-do* sudah terpenuhi, dan *hillchart* telah mencapai tahap penyelesaian. Peserta magang melakukan *testing* untuk memastikan bahwa fitur sudah berjalan sesuai dengan permintaan. Ketika semua fungsi *Generate Reconciliation Report* sudah berjalan sesuai dengan semestinya, peserta magang pun melakukan *Pull Request* melalui Github untuk meminta peninjauan kode kepada *supervisor (Product Owner)*.

3.2.7 Code review fixing Cycle 2: Finance Reconciliation Report

Setelah melakukan *Pull Request*, *Supervisor* akan melakukan pengecekan terkait code yang sudah dibuat dan memastikan bahwa code yang dibuat berkualitas, konsisten, dan berjalan sesuai dengan keinginan perusahaan.

Setelah *supervisor* melakukan *code review*, maka *supervisor* akan menandai *line code* yang masih salah atau memerlukan revisi dengan fitur *comment* pada Github. Kemudian, peserta magang akan memperbaiki *code* sesuai dengan permintaan *supervisor* dan melakukan '*commit*' untuk mengumpulkan *code* revisi dan meminta evaluasi. Kembali terkait code yang sudah di revisi. Proses tersebut dilakukan terus menerus hingga semua code sudah berkualitas, konsisten, dan sesuai. Berikut merupakan beberapa commit yang dilakukan oleh peserta magang ketika mengerjakan tugas *Generate Reconciliation Report*.



Gambar 3. 34 Hasil Commit Fitur *Finance Reconciliation Report*

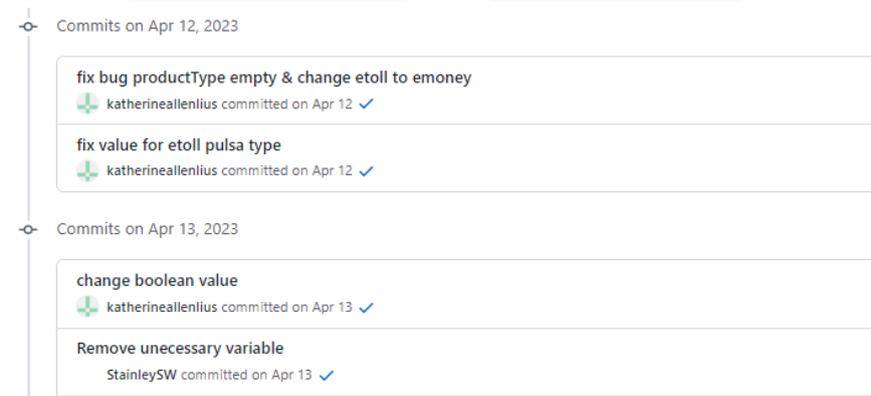
Ketika kualitas code dan hasil eksekusi dinilai sudah sesuai dengan permintaan perusahaan, maka tugas mengerjakan *Generate Reconciliation Report* dinyatakan selesai dengan tepat waktu.

3.2.8 Mengerjakan tugas *fixing bug productType empty & change etoll to emoney*

Ketika fitur *Generate Reconciliation Report* mulai digunakan oleh team Finance, ditemukan *bug* bahwa apabila *user* tidak mengisi *field* *productType* maka *file* rekonsiliasi gagal melakukan *generate*. Maka untuk mengatasi *bug* ini, peserta magang melakukan *debugging*, dan setelah menemukan sumber masalah dari *bug* maka peserta magang mulai melakukan penyesuaian code agar *file* rekonsiliasi tetap berhasil di *generate* ketika *productType* kosong. Setelah itu berdasarkan permintaan *Finance*, peserta magang juga melakukan perubahan pada tampilan keterangan produk ‘*etoll*’ menjadi ‘*emoney*’. Setelah memastikan *bug* berhasil diperbaiki maka, peserta magang pun melakukan *Pull Request* melalui Github untuk meminta peninjauan kode kepada *supervisor* (*Product Owner*).

Setelah melakukan *Pull Request*, *Supervisor* akan melakukan pengecekan terkait code yang sudah dibuat dan memastikan bahwa code yang dibuat berkualitas, konsisten, dan berjalan sesuai dengan keinginan perusahaan. Setelah *supervisor* melakukan *code review*, maka *supervisor* akan menandai *line code* yang masih salah atau memerlukan revisi dengan fitur *comment* pada Github. Kemudian, peserta magang akan memperbaiki *code* sesuai dengan permintaan *supervisor* dan melakukan ‘*commit*’ untuk mengumpulkan code revisi dan meminta evaluasi Kembali terkait code yang sudah di revisi. Proses tersebut dilakukan terus menerus hingga semua code sudah berkualitas, konsisten, dan

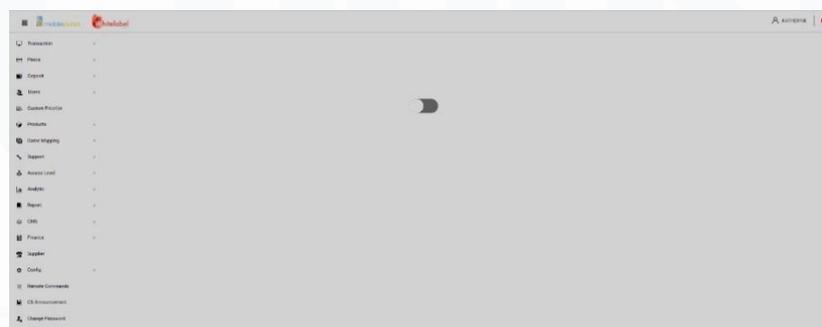
sesuai. Berikut merupakan beberapa *commit* yang dilakukan oleh peserta magang ketika mengerjakan tugas memperbaiki *bug productType empty & change etoll to emoney*.



Gambar 3. 35 Hasil *commit fixing bug productType empty & change etoll to emoney*

3.2.9 Mengerjakan tugas *Re-styling Maintenance mode menu*

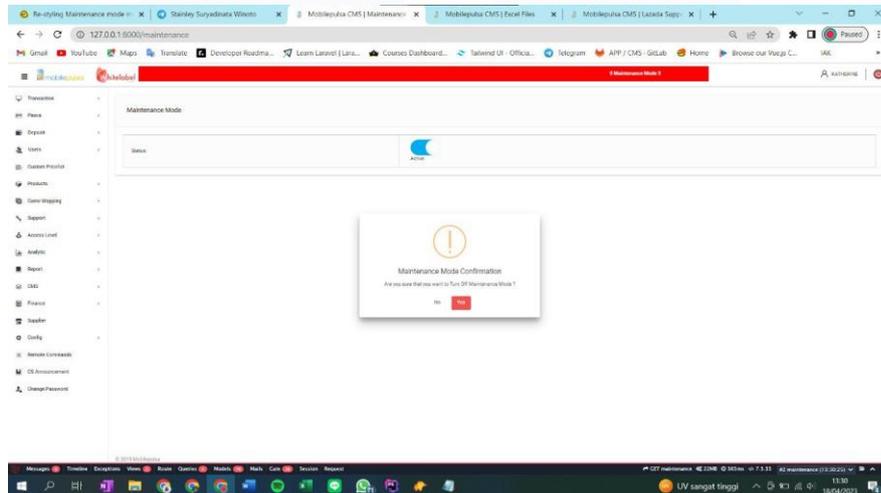
Pada task ini, peserta magang diminta untuk membuat tampilan front end yang mendukung fungsi UI/UX dengan baik pada *Menu Maintenance Mode*. Menu ini terletak pada menu Config >> Maintenance. Berikut merupakan tampilan *Menu Maintenance Mode* sebelum di *re-styling*.



Gambar 3. 36 Halaman *Maintenance Mode* sebelum di *Re-Styling*

Peserta magang diminta untuk menambahkan keterangan terkait *toggle on/off button maintenance mode*, membuat *baseline*

tampilan menu, dan menambahkan *confirmation box* ketika user melakukan aksi pada tombol *on/off*. Berikut merupakan *hasil re-styling menu maintenance mode* yang dilakukan.



Gambar 3. 37 *Pop-up Confirmation Box* di halaman *Maintenance Mode*

Setelah memastikan bahwa tampilan halaman *Maintenance Mode* sudah sesuai dengan keinginan *supervisor*, maka peserta magang pun melakukan *Pull Request* melalui Github untuk meminta peninjauan kode kepada *supervisor (Product Owner)*.

3.2.10 **Persiapan *Upcoming Cycle 3 (Small Batch Tasks)***

Untuk mempersiapkan *Cycle 3* yang akan datang, team Internal melakukan *meeting* untuk membahas mengenai task-task yang akan dijalankan di *cycle 3*. Pada *cycle* ini, peserta magang mendapat peran sebagai *Small Batch Team* Internal yang difokuskan pada perbaikan *bugs* dan *quick wins* yang sudah ditemukan di *cycle-cycle* sebelumnya, Berikut merupakan *task* yang sudah ditemukan dan dijadikan target untuk diselesaikan selama *cycle 3*.

SMALL Batch

Untuk small batch di cycle 3 akan berfokus pada perbaikan bugs dan quick wins yang ditemukan selama cycle" sebelumnya.

PIC: Katherine

Task List:

1. Restyling menu maintenance mode - DONE
2. Update menu CRUD untuk menambahkan kolom "product_category" pada table mspulsa dan mspasca (2 minggu)
3. [Clear cache RC mapping - Internal: CMS](#) (1 minggu)
4. [Custom PL Postpaid upload excel apply on empty - Internal: CMS](#) (1 minggu)
5. Sensor kode voucher pada menu update report (1 minggu)
6. Search user pada menu custom pulsa active time (1 minggu)

Gambar 3. 38 List Task Small Batch Cycle 3

Namun, tidak menutup kemungkinan bahwa seiring berjalannya waktu task dapat bertambah untuk *Small Batch*. Hal ini karena *task* akan disesuaikan sesuai dengan urgensinya.

3.2.11 Mengerjakan tugas *adjustment Update menu CRUD untuk table mspulsa dan mspasca (product_category) & Code review fixing adjustment Update menu CRUD untuk table mspulsa dan mspasca (product_category)*

Pada task ini, peserta magang diminta untuk melakukan penyesuaian pada code yang sudah ada di CMS, hal ini dikarenakan terdapat penambahan kolom pada table dalam database perusahaan, yaitu kolom `product_category`. Oleh karena itu, perlu dilakukan penyesuaian terhadap code yang terpengaruhi terhadap perubahan database tersebut.

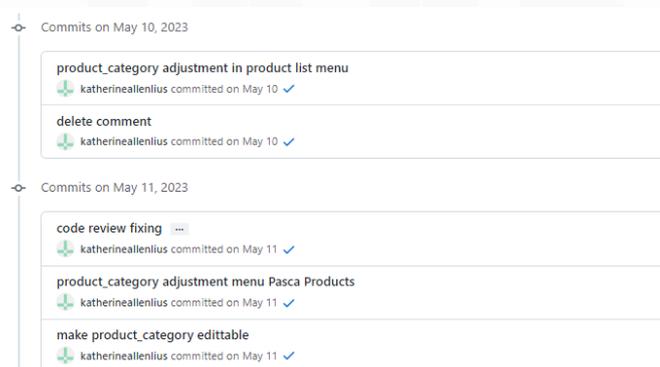
Setelah dilakukan pengecekan, peserta magang menyimpulkan bahwa penyesuaian utama perlu dilakukan pada 2 menu yang terdampak dari perubahan kolom ini, yaitu: Menu Product List dan Menu Pasca List. Penyesuaian banyak dilakukan pada fungsi di backend terutama pada fungsi CRUD kedua menu tersebut. Setelah memastikan semua penyesuaian dan fungsi pada kedua menu berjalan normal, maka peserta magang pun melakukan

Pull Request melalui Github untuk meminta peninjauan kode kepada *supervisor (Product Owner)*.

3.2.12 *Code review fixing adjustment Update menu CRUD untuk table mspulsa dan mspasca (product_category)*

Setelah melakukan *Pull Request*, *Supervisor* akan melakukan pengecekan terkait code yang sudah dibuat dan memastikan bahwa code yang dibuat berkualitas, konsisten, dan berjalan sesuai dengan keinginan perusahaan.

Setelah *supervisor* melakukan *code review*, maka supervisor akan menandai *line code* yang masih salah atau memerlukan revisi dengan fitur comment pada Gitlab. Kemudian, peserta magang akan memperbaiki code sesuai dengan permintaan *supervisor* dan melakukan '*commit*' untuk mengumpulkan code revisi dan meminta evaluasi Kembali terkait code yang sudah di revisi. Proses tersebut dilakukan terus menerus hingga semua code sudah berkualitas, konsisten, dan sesuai. Berikut merupakan beberapa commit yang dilakukan oleh peserta magang ketika mengerjakan tugas *Update menu CRUD* untuk table mspulsa dan mspasca (*product_category*).



Gambar 3. 39 Hasil *Commit adjustment Update menu CRUD* untuk *table mspulsa* dan *mspasca (product_category)*

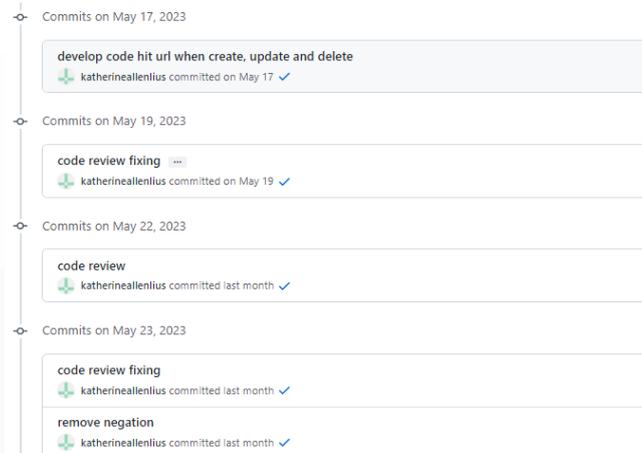
3.2.13 Mengerjakan tugas *Clear Cache* di fitur RC Mapping & Code Review Fixing task *Clear Cache* di fitur RC Mapping

Pada task ini, peserta magang ditugaskan untuk melakukan *develop code* untuk melakukan *hit url* yang diminta *supervisor* untuk melakukan *clear cache* ketika user melakukan *create*, *update*, dan *delete* pada fitur RC Mapping. Peserta magang mengerjakan tugas ini dengan menambahkan fungsi untuk *hit url* pada file *controller* ketika melakukan fungsi *create*, *update*, dan *delete*.

Setelah berhasil *develop code*, peserta magang dapat mengecek *code* yang telah dibuat apakah berhasil *hit url* atau tidak melalui *tools Insomnia* yang berfungsi untuk menguji API yang sudah dibuat. Peserta magang berhasil mengerjakan task ketika mendapatkan *response* 'SUCCESS' di *Insomnia*.

Setelah memastikan bahwa *url* berhasil di hit dalam fungsi CUD, maka peserta magang pun melakukan *Pull Request* melalui Github untuk meminta peninjauan kode kepada *supervisor (Product Owner)*.

Setelah *supervisor* melakukan *code review*, maka *supervisor* akan menandai *line code* yang masih salah atau memerlukan revisi dengan fitur *comment* pada Github. Kemudian, peserta magang akan memperbaiki *code* sesuai dengan permintaan *supervisor* dan melakukan 'commit' untuk mengumpulkan *code* revisi dan meminta evaluasi Kembali terkait *code* yang sudah di revisi. Proses tersebut dilakukan terus menerus hingga semua *code* sudah berkualitas, konsisten, dan sesuai. Berikut merupakan beberapa *commit* yang dilakukan oleh peserta magang ketika mengerjakan tugas *Clear cache* pada menu RC Mapping.

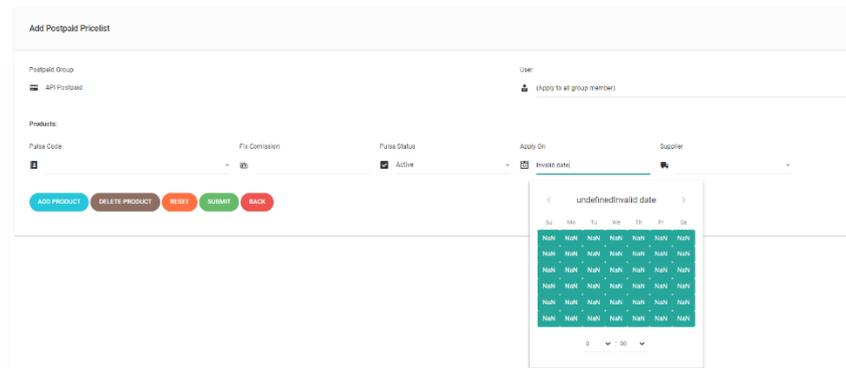


Gambar 3. 40 Hasil *Commit Task Clear Cache RC Mapping*

3.2.14 Mengerjakan tugas *fixing bug apply_on* di fitur *Custom Pricelist Postpaid*

Pada *task* ini, peserta magang ditugaskan untuk memperbaiki bug yang ada pada fitur *Custom Pricelist Postpaid*, yang muncul ketika *user* melakukan *upload file excel* berisi *custom pricelist* namun gagal membaca kolom *apply_on* sehingga menjadi '*empty*'. Setelah dilakukan debugging, peserta magang menemukan sumber masalah yaitu pada fungsi membaca hasil *upload file*, *header* untuk '*apply_on*' hilang sehingga tidak dapat disimpan oleh sistem.

Setelah memperbaiki code, peserta magang melakukan testing dan memastikan bahwa bug telah diperbaiki. Selain itu, ketika mengerjakan *task* peserta magang juga menyelesaikan dan memperbaiki *bug* pada halaman *Add Postpaid Pricelist* yang menunjukan *invalid date* pada bagian *Apply On*.

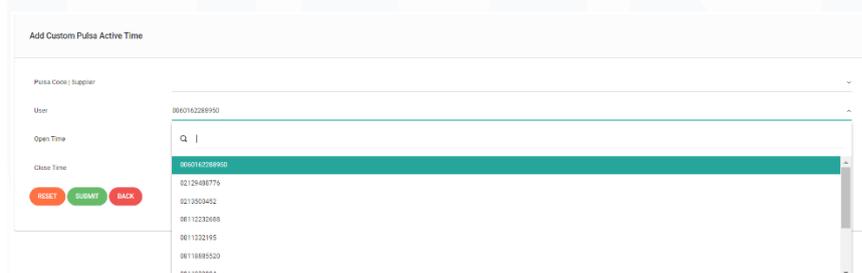


Gambar 3. 41 *Bug Invalid Date* pada *Input Apply On*

Setelah memastikan bahwa, tidak ada lagi masalah peserta magang melakukan *Pull Request* melalui Github untuk meminta peninjauan kode kepada *supervisor (Product Owner)*.

3.2.15 Mengerjakan tugas menambahkan *search user* pada fitur *Custom Pulsa Active Time*

Pada task ini, peserta magang ditugaskan untuk menambahkan fungsi *search user* pada fitur *Custom Pulsa Active Time*. Berikut tampilan dari fungsi *search user* yang sudah dibuat oleh peserta magang.

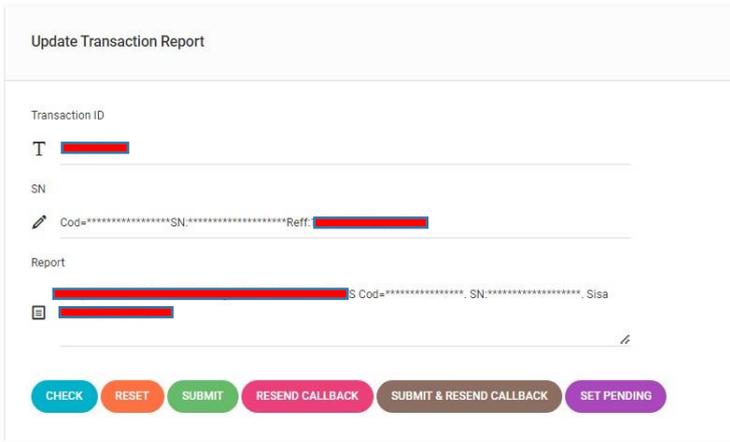


Gambar 3. 42 Fungsi *Search user* pada *Custom Pulsa Active Time*

Setelah memastikan bahwa tampilan dan fungsi sudah sesuai dengan keinginan *supervisor*, maka peserta magang pun membuat *Pull Request* untuk mengajukan *Code Review* dari *supervisor*.

3.2.16 Mengerjakan tugas sensor kode voucher di fitur *Update Report*

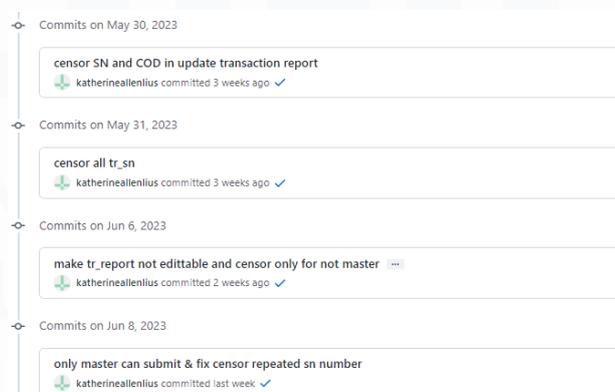
Pada *task* ini, peserta magang ditugaskan untuk melakukan sensor kode COD dan SN pada fitur *Update Transaction Report*. Tugas ini dibutuhkan untuk menjaga keamanan pengguna dan penyalahgunaan kode transaksi yang sudah ada. Peserta magang memanfaatkan fitur *Regex (Regular Expression)* untuk melakukan sensor di menu *update transaction* tersebut. Berikut merupakan tampilan *Update report* setelah dilakukan sensor.



The screenshot shows a web form titled "Update Transaction Report". It contains several input fields: "Transaction ID" with a redacted value, "SN" with a redacted value, and a "Report" field with a redacted value. Below the form, there are six buttons: "CHECK" (blue), "RESET" (orange), "SUBMIT" (green), "RESEND CALLBACK" (pink), "SUBMIT & RESEND CALLBACK" (brown), and "SET PENDING" (purple).

Gambar 3. 43 Hasil sensor kode voucher di *Update Report*

Setelah kode COD dan SN berhasil di sensor, peserta magang melakukan *Pull Request*, untuk meminta *code review* kepada *supervisor*. Berikut beberapa *commit* yang dikumpulkan oleh peserta magang selama mengerjakan *task* ini.



Gambar 3. 44 Hasil *Commit* sensor kode voucher di *Update Report*

3.2.17 Mengerjakan tugas *Total Profit* pada excel berbeda dengan *Summary CMS*

Pada task ini, peserta magang ditugaskan untuk mengerjakan bug pada fitur *Generate General Report* yaitu *Summary CMS* yang menunjukkan jumlah *total profit* yang berbeda dengan hasil yang sebenarnya pada transaksi excel. Untuk mengerjakan task ini, peserta magang melakukan *debugging* pada fungsi yang melakukan perhitungan *summary CMS*.

Setelah dilakukan *debugging*, ditemukan bahwa terdapat selisih nominal *total profit* < Rp 100, 00 rupiah antara *Summary CMS* dengan transaksi aslinya. Setelah berkonsultasi dengan supervisor, diduga perbedaan nominal tersebut disebabkan karena adanya pembulatan dari perhitungan *summary*. Setelah dilakukan *develop code* untuk membandingkan total profit secara manual dan *summary CMS*, ditemukan bahwa masalah ini disebabkan karena ada fungsi *aggregations/ query database* yang tidak sesuai dan membutuhkan penyesuaian lagi pada *database*.

Mengingat bahwa perubahan *database* beresiko dapat mempengaruhi fungsi *CMS* yang lainnya maka *supervisor* memutuskan untuk menunda perbaikan bug task ini, dan peserta magang tidak perlu melakukan *Pull Request* karena akan ditinjau ulang oleh *supervisor*.

3.2.18 Mengerjakan tugas *fixing bug* untuk transaksi > H+7 pada fitur *Update Report* dan *Set Status Transaction*

Pada task ini, peserta magang ditugaskan untuk memperbaiki bug pada fungsi *Update Report* dan *Set Status* di halaman *Transaction* yang masih bisa dilakukan oleh *role* selain master untuk

transaksi diatas 7 hari. Peserta magang berhasil mengerjakan task ini dengan cara menambahkan fungsi untuk validasi *access role user* ketika mengakses fungsi *update report* dan set status transaksi.

3.3 Kendala yang Ditemukan

Berikut adalah kendala yang ditemui oleh peserta magang selama menjalankan praktik kerja magang di PT Indobest Artha Kreasi:

1. Sulit untuk memahami konsep penggunaan framework Laravel karena tidak mendapatkan materi tersebut di masa perkuliahan.
2. Sulit untuk memahami alur proses bisnis dalam website CMS IAK, mengingat banyaknya fitur yang tersedia dan kompleksnya code yang sudah ada.

3.4 Solusi atas Kendala yang Ditemukan

Untuk mengatasi kendala-kendala yang ditemukan selama praktik kerja magang, solusi yang dijalankan adalah sebagai berikut:

1. Mempelajari dokumentasi framework Laravel, implementasi library, dan cara penggunaan tools yang belum pernah digunakan sebelumnya secara mandiri. Selain itu, peserta magang juga banyak melakukan diskusi langsung dan konsultasi dengan supervisor dan rekan kerja ketika kurang memahami tech stacks.
2. Memberi saran kepada supervisor dan membantu proses pembuatan SOP Training untuk memperkenalkan website CMS IAK dan alur bisnis bagi karyawan baru, serta panduan pemeliharaan kode.