

BAB 3

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

Kegiatan magang di PT Cranium Royal Aditama pada batch 8 dilaksanakan dengan penempatan peserta sebagai *Full Stack Developer* dan *Quality Assurance Analyst*, yang diikuti oleh enam orang peserta magang. Selama pelaksanaan program, peserta memperoleh pembekalan materi teknis serta pendampingan yang disampaikan oleh Bapak Sugito selaku VP Engineering PT Cranium Royal Aditama. Selain bimbingan formal tersebut, peserta juga mendapatkan arahan dan dukungan tambahan dari mentor maupun karyawan senior yang berasal dari angkatan magang sebelumnya.

Dalam pelaksanaannya, peserta magang diberikan tanggung jawab untuk berkontribusi dalam pengembangan fitur pada sistem *Enterprise Resource Planning* (ERP) milik Cranium yang sedang berada dalam tahap pengembangan. Proses koordinasi pengembangan dilakukan melalui platform Discord dan WhatsApp sebagai media komunikasi utama. Pemantauan progres pekerjaan serta pendistribusian tugas didokumentasikan melalui Google Sheets dan Figma. Di samping keterlibatan dalam pengembangan sistem ERP, peserta magang juga memperoleh kesempatan untuk menjalankan peran sebagai *Quality Assurance Analyst* pada proyek *Learning Management System*.

3.2 Tugas yang Dilakukan

Tugas yang dijalankan selama kegiatan kerja magang sebagai Full Stack Developer di PT Cranium Royal Aditama meliputi pengembangan pada sisi back-end dan front-end. Bahasa pemrograman yang digunakan mencakup JavaScript, TypeScript, framework Java Spring Boot, HTML, dan CSS, dengan bantuan perangkat lunak seperti IntelliJ, GitHub Desktop, PGAdmin, dan Postman.

Pengembangan sistem ERP yang dilakukan berfokus pada penambahan fitur parameter pada modul Finance serta perbaikan bug. Proses pengembangannya mencakup tampilan front-end untuk mempermudah pengguna dalam sub-modul yang ada di modul Finance. Selain itu, selama masa magang juga diberikan tanggung jawab sebagai *Quality Assurance Analyst* pada proyek *Learning Management System*, dengan tujuan memastikan bahwa seluruh kebutuhan dan

spesifikasi yang diminta oleh klien dapat terpenuhi dengan baik.

3.3 Uraian Pelaksanaan Magang

Berikut merupakan uraian kegiatan magang selama berada di PT Cranium Royal Aditama. Pembahasan mencakup tabel pelaksanaan kerja magang, *sitemap*, *flowchart*, serta berbagai tugas yang dilakukan.

3.3.1 Pelaksanaan Kerja Magang

Pelaksanaan kerja magang diuraikan seperti pada Tabel 3.1.

Tabel 3.1. Pekerjaan yang dilakukan setiap minggu selama pelaksanaan magang

Minggu Ke -	Pekerjaan yang dilakukan
1	Mempelajari React dan Next.js serta template yang diberikan.
2	Mengerjakan front end fitur parameter pada modul Finance Cashier & Cash Adjustment.
3	Mengerjakan front end fitur parameter pada modul Finance Cash Consolidation & Voucher In.
4	Mengerjakan front end fitur parameter pada modul Finance Voucher Out & Voucher Transfer.
5	Mengerjakan front end fitur parameter pada modul Finance Cash Transfer & Cash Withdraw.
6	Mengerjakan front end fitur parameter pada modul Finance Invoice Cash Receipt & Planned Billing.
7	Melakukan update requirement & scenario test pada LMS Phase 1
8	Melakukan scenario test pada MyLearning (user)
9	Melakukan scenario test serta run-through defect list Phase 1 - Defect 1
10	Melakukan scenario test serta run-through defect list Phase 1 - Defect 1
11	Melakukan scenario test serta run-through defect list Phase 1 - Defect 2
Lanjut di halaman berikutnya	

Tabel 3.1 – Pekerjaan yang dilakukan setiap minggu selama pelaksanaan magang (lanjutan)

Minggu Ke -	Pekerjaan yang dilakukan
12	Melakukan scenario test serta run-through defect list Phase 1 - Defect 2
13	Melakukan scenario test serta run-through defect list Phase 1 Defect 3
14	Melakukan scenario test serta run-through defect list Phase 1 - Defect 3
15	Melakukan scenario test serta run-through defect list Phase 1 - Defect 4
16	Melakukan scenario test serta run-through defect list Phase 1 Defect 4
17	Pengecekan verifikasi kendala after implementation
18	Pengecekan verifikasi kendala after implementation
19	Enhancement Phase 1
20	Enhancement Phase 1

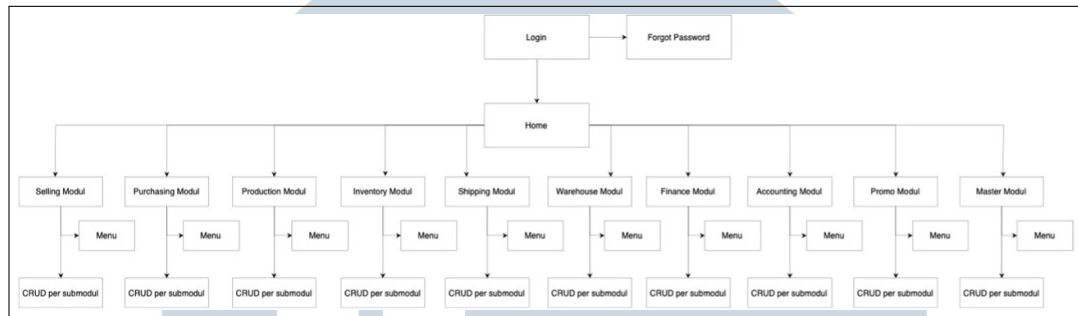
3.4 ERP Cranium

Pengembangan pada sistem ERP Cranium berfokus pada penyempurnaan fitur parameter preferences yang memungkinkan pengguna melakukan penyesuaian tampilan dan konfigurasi sesuai kebutuhan setiap submodul. Fitur ini dikembangkan untuk meningkatkan fleksibilitas dalam penggunaan ERP, terutama pada modul Finance yang memiliki banyak submodul seperti Cashier, Voucher In, Voucher Out, Cash Transfer, dan Invoice Cash Receipt. Melalui pengembangan ini, setiap halaman dan komponen frontend disesuaikan agar mampu menampilkan data secara dinamis berdasarkan preferensi pengguna, sehingga pengalaman penggunaan menjadi lebih efisien, adaptif, dan mudah untuk dikustomisasi.

3.4.1 Sitemap ERP

Sitemap berperan sebagai pedoman navigasi dalam proses pengembangan sebuah *website* guna memudahkan pengguna dalam melakukan akses serta membantu memperoleh informasi yang diperlukan secara cepat dan efisien. Pada pengembangan sistem ERP, sitemap disusun sebagaimana ditampilkan pada Gambar 3.1, di mana halaman *login* berfungsi sebagai titik awal akses bagi

pengguna. Setelah proses autentikasi berhasil dilakukan, sistem akan mengarahkan pengguna ke halaman yang relevan sesuai dengan kebutuhan dan hak akses masing-masing.



Gambar 3.1. Sitemap

Sumber: [11]

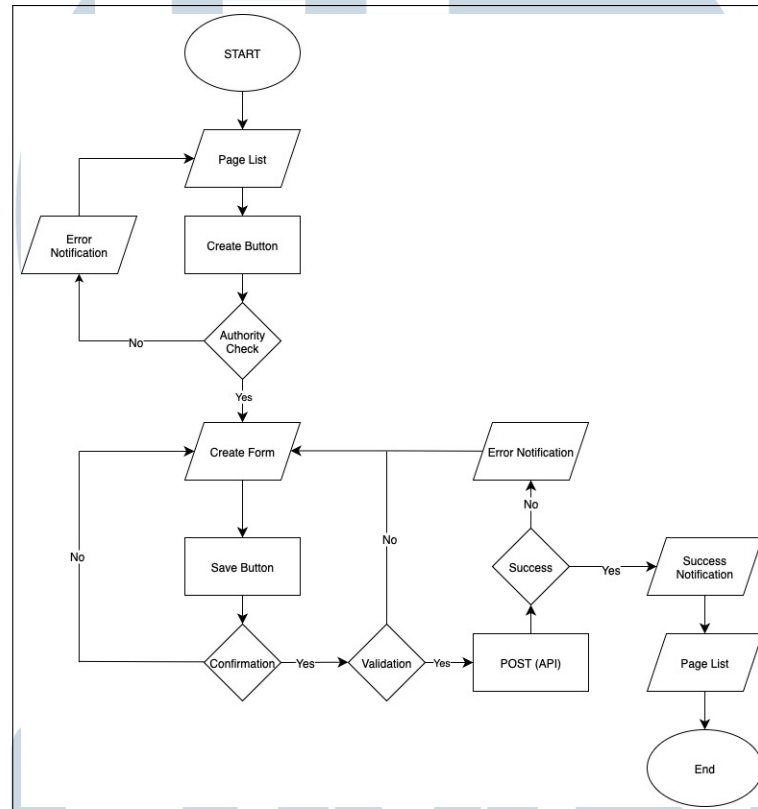
3.4.2 Flowchart Alur Kerja ERP

Flowchart digunakan sebagai acuan dalam memahami alur operasional pada sistem ERP yang sedang dikembangkan. Setiap halaman yang dirancang dan dioptimalkan mencerminkan implementasi fungsi utama sistem, yaitu *create*, *read*, *update*, dan *delete*. Seluruh tahapan interaksi pengguna dengan sistem divisualisasikan secara terstruktur melalui *flowchart* guna memberikan gambaran proses yang komprehensif.

A Create

Tahapan pembuatan data (*create*) diawali dari halaman *list* pada masing-masing submodul sebagaimana ditunjukkan pada flowchart 3.2. Pengguna dapat memulai proses dengan memilih tombol *create* yang akan mengarahkan ke halaman pembuatan data. Pada halaman tersebut, sistem menampilkan *form create* yang terdiri atas beberapa kolom (*fields*) sesuai dengan karakteristik submodul terkait. Sebelum halaman *create* diakses, sistem melakukan verifikasi terhadap *authority* pengguna. Apabila hak akses pengguna tidak dinyatakan *valid*, sistem akan menampilkan notifikasi *error*. Setelah pengguna memperoleh otorisasi yang sesuai, pengguna dapat mengisi *form create* dan menekan tombol *save*. Selanjutnya, sistem menampilkan *pop-up* konfirmasi sebagai tahap akhir sebelum proses validasi dilakukan. Jika seluruh data wajib telah terisi dengan benar, data akan dikirimkan melalui *API* menggunakan metode *POST*. Sistem kemudian melakukan proses

validasi untuk mendeteksi kemungkinan duplikasi maupun data yang tidak *valid*. Apabila proses tersebut gagal, sistem akan menampilkan notifikasi *error* dan mengarahkan pengguna kembali ke halaman *create*. Sebaliknya, jika proses berhasil, sistem menampilkan notifikasi keberhasilan dan mengarahkan pengguna kembali ke halaman *list*.



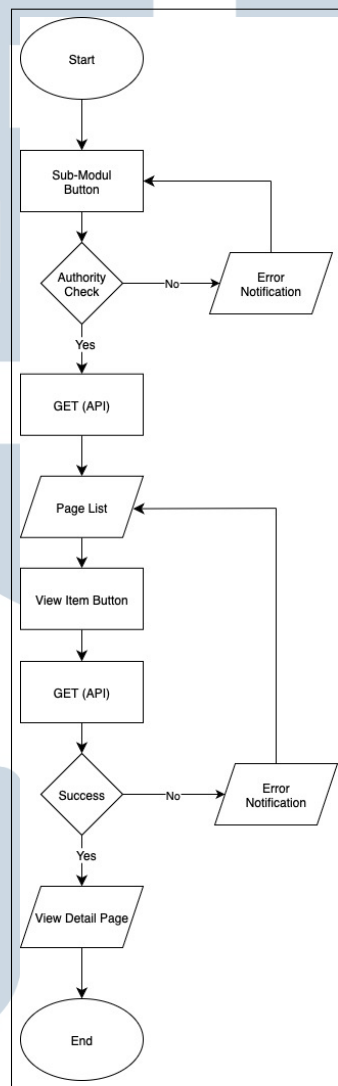
Gambar 3.2. Flowchart create

Gambar 3.2 menyajikan ilustrasi alur proses *create* pada sistem ERP yang dikembangkan oleh Cranium. Mekanisme *create* tersebut diterapkan secara konsisten dan digunakan pada seluruh submodul yang terdapat dalam sistem ERP.

B Read

Pada tahap penampilan data (*read*), pengguna diawali dengan menentukan *submodul* yang diinginkan melalui menu pada *sidebar*. Apabila pengguna memiliki hak akses (*authority*) yang sesuai, sistem akan mengeksekusi pemanggilan *API* dengan metode *READ* untuk menampilkan data dari *submodul* yang dipilih. Sebaliknya, jika hak akses pengguna tidak memenuhi ketentuan, sistem akan

menampilkan notifikasi *error* sebagaimana ditunjukkan pada flowchart 3.3. Setelah data berhasil ditampilkan dalam bentuk daftar (*list*), pengguna dapat menelusuri informasi lebih rinci dengan menekan tombol *view*. Tindakan tersebut akan membuka halaman *view* yang menyajikan detail data berdasarkan *id* yang dipilih.



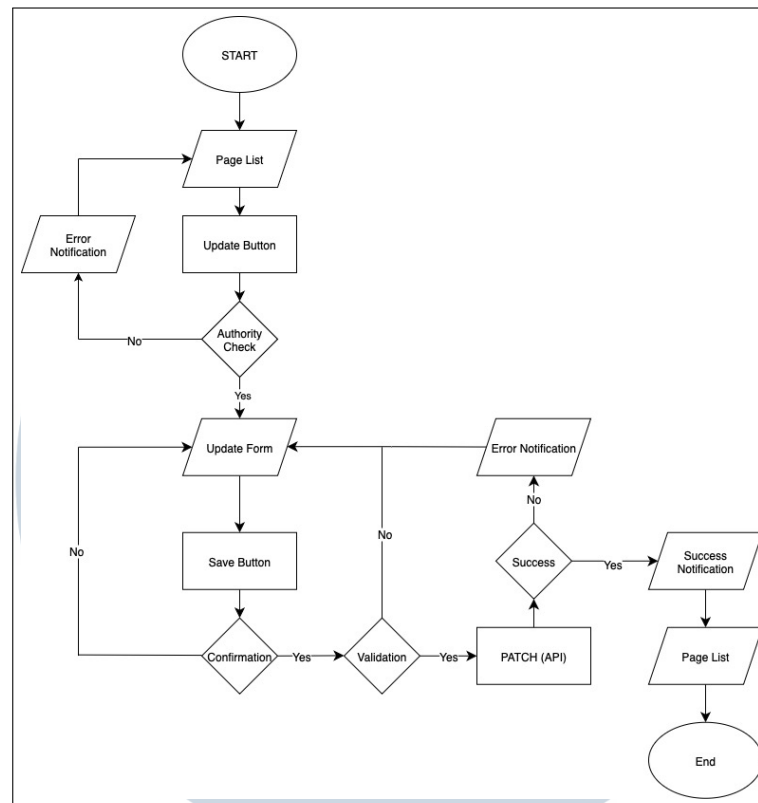
Gambar 3.3. Flowchart read

Gambar 3.3 menyajikan representasi alur proses *read* pada sistem ERP yang dikembangkan oleh Cranium. Mekanisme *read* tersebut diterapkan secara seragam dan digunakan pada seluruh submodul yang tersedia dalam sistem ERP.

C Update

Proses pembaruan data (*update*) dimulai dari halaman *list* pada masing-masing *submodul* sebagaimana ditunjukkan pada flowchart 3.4. Pengguna memilih tombol *update* untuk diarahkan ke halaman pembaruan data, di mana tersedia *form* dengan sejumlah kolom (*fields*) yang dapat digunakan untuk mengubah atau memperbarui data. Sebelum halaman tersebut diakses, sistem melakukan pemeriksaan hak akses (*authority checking*) terhadap pengguna. Apabila pengguna memiliki otorisasi yang sesuai, proses *update* dapat dilanjutkan; sebaliknya, jika otorisasi tidak terpenuhi, sistem akan menampilkan notifikasi *error*. Setelah *form* diisi, pengguna menekan tombol *save* yang akan memunculkan *pop-up modal* sebagai konfirmasi sebelum sistem memproses permintaan pembaruan. Jika data berhasil melewati tahap validasi dan tidak ditemukan duplikasi maupun data yang tidak valid, sistem akan mengirimkan permintaan *API* dengan metode *PATCH*. Namun, apabila validasi tidak terpenuhi, notifikasi *error* akan ditampilkan dan pengguna diarahkan kembali ke halaman *update form*. Jika proses pembaruan berhasil, sistem akan menampilkan notifikasi keberhasilan dan mengarahkan pengguna kembali ke halaman *list*.



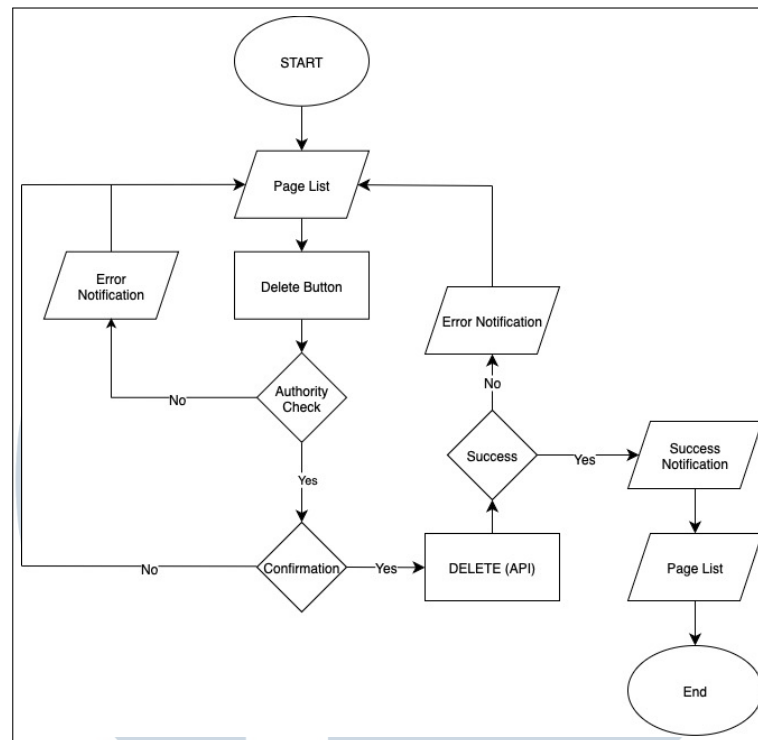


Gambar 3.4. *Flowchart update*

Gambar 3.4 menggambarkan alur kerja *update* pada sistem ERP yang dikembangkan oleh Cranium. Alur kerja *update* diterapkan secara konsisten dan berlaku pada seluruh submodul yang terdapat dalam sistem ERP.

D Delete

Pada proses penghapusan data (*delete*), alur kerja dimulai dari halaman *list* pada setiap *submodul* sebagaimana ditunjukkan pada flowchart 3.5. Pengguna menekan tombol *delete* pada data dengan *id* yang dipilih, kemudian sistem akan menampilkan *pop-up modal* konfirmasi. Apabila pengguna memilih “Yes”, sistem akan melakukan permintaan *API* dengan metode *DELETE* untuk menghapus data tersebut. Jika terjadi kesalahan, sistem akan menampilkan notifikasi *error* kepada pengguna. Namun, apabila proses berhasil, sistem akan menampilkan notifikasi sukses, data akan terhapus, dan pengguna akan diarahkan kembali ke halaman *list*.

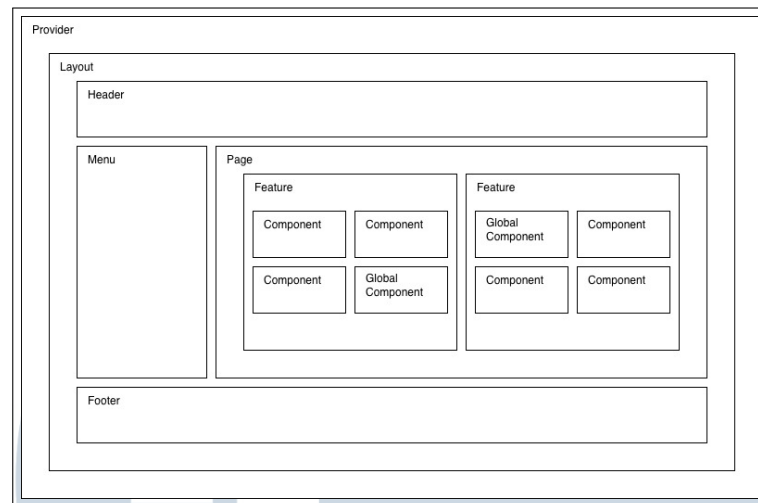


Gambar 3.5. Flowchart delete

Gambar 3.5 menggambarkan alur kerja *delete* pada sistem ERP yang dikembangkan oleh Cranium. Alur kerja *delete* diterapkan secara konsisten dan berlaku pada seluruh submodul yang tersedia.

3.4.3 Arsitektur Frontend Sistem ERP

Dalam pengembangan sistem Enterprise Resource Planning (ERP) di PT Cranium Royal Aditama, perancangan arsitektur frontend dilakukan untuk memastikan struktur aplikasi memiliki keteraturan dan skalabilitas yang baik. Diagram pada Gambar 3.6 menggambarkan hierarki komponen pada sisi frontend yang dikembangkan selama kegiatan magang. Arsitektur ini terdiri atas beberapa lapisan utama, yaitu Provider, Layout, Header, Menu, Page, Feature, Component, Global Component, dan Footer.



Gambar 3.6. *Diagram Arsitektur Frontend Sistem ERP di PT Cranium Royal Aditama*

Lapisan Provider berfungsi sebagai penyedia konteks global yang memungkinkan berbagai komponen dalam aplikasi berbagi data atau konfigurasi yang sama, seperti tema, autentikasi, maupun pengaturan bahasa. Lapisan Layout bertanggung jawab untuk mengatur struktur utama tampilan halaman, mencakup penempatan elemen seperti Header, Menu, dan Footer agar tampilan antarhalaman konsisten.

Sementara itu, lapisan Page dan Feature digunakan untuk mengelompokkan komponen berdasarkan fungsi atau modul tertentu dalam sistem ERP, misalnya modul Master, Inventory, dan Finance Management. Di dalamnya terdapat Component sebagai elemen antarmuka yang bersifat spesifik untuk setiap halaman, serta Global Component yang bersifat reusable dan dapat digunakan di berbagai modul.

Dengan rancangan arsitektur seperti ini, pengembangan frontend menjadi lebih terstruktur, mudah dikelola, serta mendukung pengembangan berkelanjutan (scalable development). Selain itu, pemisahan tanggung jawab antar lapisan juga memudahkan kolaborasi antar pengembang dan proses pemeliharaan sistem ERP secara keseluruhan.

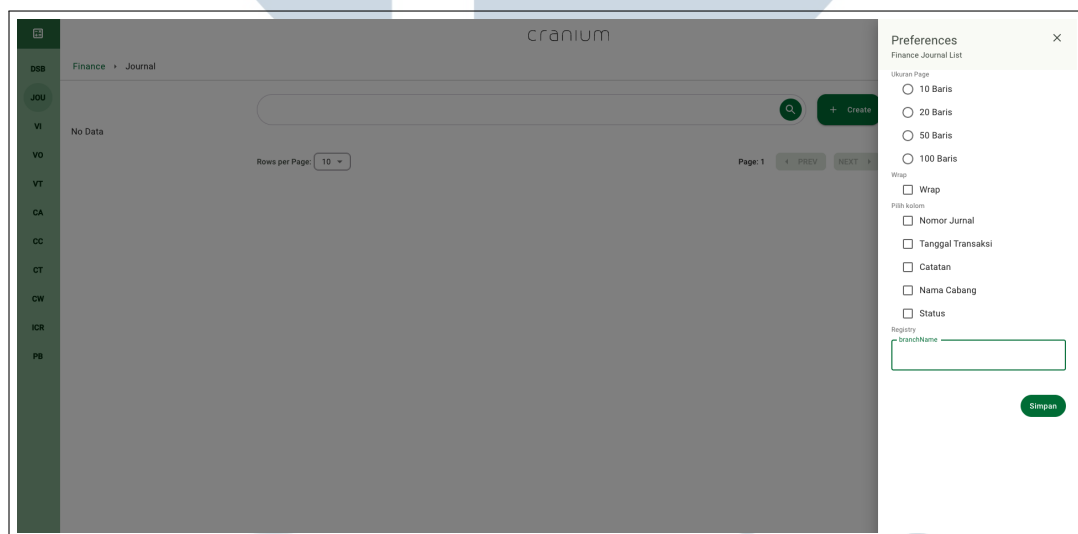
3.4.4 Pengembangan dan Konsep

Pengembangan dan juga perbaikan fitur yang dilakukan selama magang di PT Cranium Royal Aditama serta konsep yang digunakan untuk pengembangan.

A Pengembangan

Selama proses pengembangan proyek ERP Cranium, kegiatan utama difokuskan pada penyempurnaan fitur parameter preferences serta perbaikan terhadap berbagai bug yang ditemukan. Pengembangan fitur parameter preferences dilakukan agar pengguna ERP dapat memberikan kostumisasi terhadap konten yang ada.

Pengembangan fitur parameter preferences dilakukan pada submodul yang ada pada ERP terutama Modul Finance. Terdapat beberapa submodul yang dikerjakan seperti Cashier, Voucher In, Voucher Out, Invoice Cash Receipt, Cash Withdraw, Cash Consolidation, Voucher Transfer, Cash Transfer, Planned Billing, Cash Adjustment. Pengembangan ini dilakukan terhadap components yang ada pada masing-masing submodul agar dapat menampilkan hasil yang sesuai pada sisi Front End. Perubahan pada pages serta penambahan components dilakukan sebagai adaptasi dari fitur baru ini.



Gambar 3.7. Tampilan parameter preferences pada submodul Journal.

Gambar 3.7 merupakan tampilan dari parameter preferences yang ada pada ERP milik Cranium. Tampilan sidebar preferences dapat diakses dari panel masing-masing submodul dengan menekan *gambar settings*. Terdapat beberapa fitur yang dapat diakses sesuai dengan keinginan pengguna diantaranya ukuran page, wrap, kolom, dan registry (masih dikembangkan).

```
1 type JournalListPreferenceProps = {  
2   parameterName?: string;
```

```

3 };
4
5 export const JournalListPreference: React.FC<
    JournalListPreferenceProps> = ({ parameterName }) => {
6     const { showNotification } = useNotifications();
7     const { t } = useTranslation(['common', 'user', 'master']);
8
9     const journalListPreferenceData = useMyUserParameter({
        parameterName });
10    const journalListPreference = journalListPreferenceData.data;
11
12    const parameterColumn = journalListPreference?.parameterJson.
        column ?? {};
13    const parameterRegistry = journalListPreference?.parameterJson.
        registry ?? {};
14
15    const [isWrap, setIsWrap] = React.useState<boolean>(false);
16    const [pageSize, setPageSize] = React.useState<number | null>(
        null);
17    const [column, setColumn] = React.useState<Column | undefined>(
        parameterColumn);
18    const [registry, setRegistry] = React.useState<Registry |
        undefined>(parameterRegistry);

```

Kode 3.1: Deklarasi Props dan State Komponen pada JournalListPreference.tsx

Kode 3.1 mendefinisikan *props*, *state*, serta *hook* utama yang digunakan dalam komponen:

- *parameterName* adalah nama parameter yang digunakan untuk mengambil dan menyimpan pengaturan preferensi pengguna.
- *useMyUserParameter* mengambil data parameter dari server berdasarkan nama tersebut.
- *parameterColumn* dan *parameterRegistry* menyimpan konfigurasi kolom dan pengaturan daftar jurnal.
- *State* seperti *isWrap*, *pageSize*, *column*, dan *registry* mengatur tampilan dan pengaturan yang dipilih oleh pengguna.
- Dengan pendekatan *reactive*, setiap perubahan *state* akan otomatis memperbarui tampilan UI tanpa perlu memuat ulang halaman.

```

1   const handleChangeColumn = (event: React.ChangeEvent<
    HTMLInputElement>) => {
2   setColumn({
3     ...column,
4     [event.target.name as keyof JournalColumnPreference]: event.
      target.checked,
5   });
6 };
7
8   const handleChangeWrap = (event: React.ChangeEvent<
    HTMLInputElement>) => {
9   setIsWrap(event.target.checked);
10 };
11
12  const handleChangeRegistry = (event: React.ChangeEvent<
    HTMLInputElement>) => {
13  setRegistry({
14    ...registry,
15    [event.target.name as keyof JournalRegistryPreference]: event.
      target.value,
16  });
17 };

```

Kode 3.2: Fungsi `handleChange` pada `journal-list-preference.tsx`

Ketiga fungsi pada potongan kode 3.2 digunakan untuk menangani interaksi pengguna pada *form settings*.

- *handleChangeColumn* memperbarui status kolom yang dipilih (misalnya kolom “*Date*”, “*Amount*”, atau “*Branch*”).
- *handleChangeWrap* menentukan apakah tampilan tabel dibungkus atau tidak.
- *handleChangeRegistry* memperbarui nilai *registry*, seperti jumlah baris per halaman atau mode tampilan.

Semua perubahan disimpan di *state* lokal terlebih dahulu sebelum dikirim ke server melalui API.

```

1   const onSuccess = (userParameter: UserParameter) => {
2     showNotification({
3       type: 'success',
4       title: 'Success',
5       duration: NOTIFICATION_DURATION_SUCESS,
6       message: t('user.userParameter.updated', {

```

```

7      ns: 'user',
8      userParameter,
9    )),
10  });
11 };
12
13 const onError = (resError: ResponseError) => {
14   showNotification({
15     type: 'error',
16     title: 'Error',
17     duration: NOTIFICATION_DURATION_ERROR,
18     message: resError.myError.errors.join(', '),
19   });
20 };
21
22 const updateUserParameter = useUpdateUserParameter({
23   onSuccess,
24   onError,
25   parameterName,
26 });
27
28 const createUserParameter = useCreateUserParameter({
29   onSuccess,
30   onError,
31 });

```

Kode 3.3: Fungsi Update dan Create Parameter

Kode 3.3 merupakan potongan kode yang bertanggung jawab untuk komunikasi antara UI dan server API.

- *onSuccess* menampilkan notifikasi keberhasilan ketika parameter berhasil diperbarui atau dibuat.
- *onError* menampilkan pesan kesalahan yang diterima dari server, memudahkan pengguna mengetahui penyebab kegagalan.
- *useUpdateUserParameter* dan *useCreateUserParameter* memanfaatkan custom hook untuk mengirim data preferensi ke API.

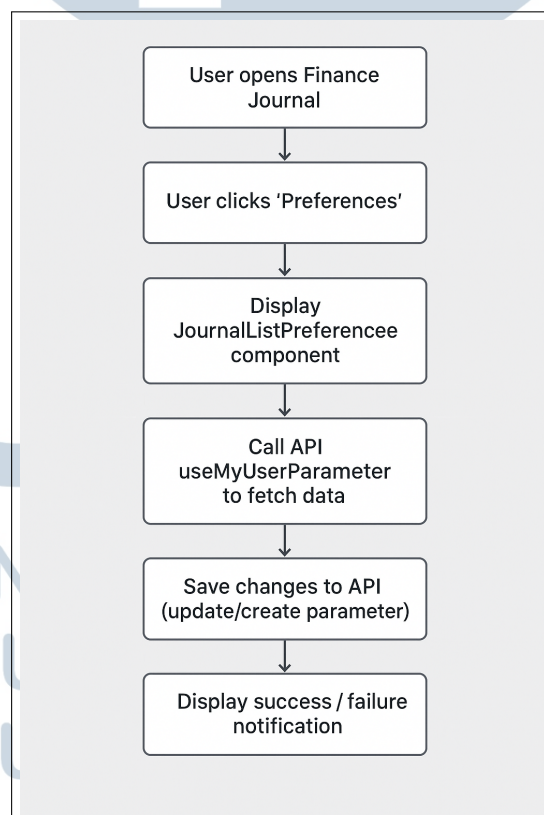
Dengan mekanisme ini, pengguna mendapat notifikasi langsung terhadap setiap *action* yang dilakukan, sehingga lebih interaktif dan transparan.

B Konsep

Fitur *Parameter Preference* merupakan fitur yang dirancang untuk memberikan fleksibilitas kepada pengguna dalam melakukan pengaturan tampilan atau konfigurasi data sesuai kebutuhan masing-masing pengguna. Fitur ini berfungsi untuk menyimpan preferensi pengguna (*user preferences*) terkait pengaturan kolom (*column visibility*), tata letak data, serta parameter tampilan lain agar dapat digunakan kembali pada saat pengguna membuka halaman submodul di kemudian hari.

Dalam implementasinya, sistem ini mengintegrasikan antara frontend *component* dengan backend API melalui proses pertukaran data menggunakan metode *fetch*, *update*, dan *create* parameter. Proses ini dilakukan agar setiap perubahan yang dilakukan oleh pengguna dapat tersimpan secara persisten pada basis data dan ditampilkan ulang sesuai dengan preferensi yang telah ditentukan.

Gambar 3.8 berikut menunjukkan alur kerja dari fitur Parameter Preference dengan contoh Finance - Journal:



Gambar 3.8. Alur Sistem Parameter Preference

(1) *User Opens Finance Journal*

Proses dimulai ketika pengguna membuka halaman Finance Journal pada sistem ERP. Halaman ini menampilkan daftar transaksi jurnal keuangan yang telah tercatat. Pada tahap ini, sistem mempersiapkan antarmuka pengguna dengan menampilkan data jurnal default yang diambil dari API tanpa konfigurasi khusus dari pengguna.

(2) *User Clicks 'Preferences'*

Setelah halaman jurnal terbuka, pengguna dapat mengakses tombol Preferences untuk melakukan pengaturan parameter tampilan. Aksi ini menunjukkan keinginan pengguna untuk memodifikasi pengaturan sesuai kebutuhan, misalnya menentukan kolom mana yang ingin ditampilkan atau menyembunyikan, mengubah ukuran tampilan, atau menentukan jumlah baris data yang akan dimuat per halaman.

(3) *Display JournalListPreference Component*

Setelah tombol *Preferences* ditekan, sistem menampilkan *JournalListPreference Component*. Komponen ini merupakan antarmuka interaktif berbasis *React* dan *Material UI* yang berfungsi sebagai wadah bagi pengguna untuk melakukan konfigurasi parameter. Komponen ini menampilkan berbagai opsi pengaturan dalam bentuk *checkbox*, *radio button*, maupun *text field* agar pengguna dapat dengan mudah menyesuaikan tampilan jurnal.

(4) *Call API useMyUserParameter to Fetch Data*

Ketika komponen *JournalListPreference* aktif, sistem secara otomatis memanggil API *useMyUserParameter* untuk mengambil data parameter yang sebelumnya telah disimpan oleh pengguna. Jika parameter sudah pernah dibuat, maka sistem akan menampilkan konfigurasi tersebut agar pengguna dapat melihat dan memperbarui pengaturannya. Jika belum ada data, maka sistem akan menampilkan pengaturan default. Langkah ini menjamin personalisasi pengalaman pengguna serta konsistensi tampilan antarsesi.

(5) *Save Changes to API (Update/Create Parameter)*

Setelah pengguna melakukan perubahan dan menekan tombol *Save*, sistem akan memproses hasil input tersebut dan mengirimkannya kembali ke API. Jika parameter pengguna sudah ada sebelumnya, maka proses update dilakukan; sedangkan jika belum ada, sistem akan membuat parameter

baru melalui proses *create*. Tahap ini memastikan bahwa setiap preferensi pengguna tersimpan secara permanen di *server* dan dapat digunakan kembali pada sesi berikutnya.

(6) *Display Success/Failure Notification*

Setelah proses penyimpanan selesai, sistem akan memberikan umpan balik berupa notifikasi kepada pengguna. Jika proses berhasil, sistem menampilkan pesan *success notification* yang menandakan bahwa parameter berhasil diperbarui. Sebaliknya, jika terjadi kegagalan akibat masalah jaringan atau validasi data, sistem akan menampilkan *error notification* agar pengguna mengetahui bahwa penyimpanan parameter tidak berhasil.

3.5 Quality Assurance Analyst

Selama kegiatan magang di Cranium, saya berkesempatan terlibat dalam berbagai project yang sedang berjalan, yang jumlahnya cukup banyak dan memiliki karakteristik yang beragam. Keragaman ini memberikan tantangan sekaligus pengalaman berharga dalam memahami alur kerja pengembangan sistem di lingkungan perusahaan. Setiap project memiliki kebutuhan, kompleksitas, serta prioritas yang berbeda, sehingga saya perlu melakukan penyesuaian analisis secara cepat dan akurat. Peran sebagai Assurance Analyst menuntut saya untuk memastikan bahwa setiap fitur yang dikembangkan memenuhi standar kualitas, berjalan sesuai requirement, serta siap digunakan oleh pengguna akhir tanpa mengurangi stabilitas maupun performa sistem secara keseluruhan. Melalui proses ini, saya mendapatkan pemahaman yang lebih mendalam mengenai pentingnya quality assurance dalam keberhasilan implementasi sebuah project teknologi.

3.5.1 LMS Quality Assurance

Proyek LMS merupakan kegiatan pengembangan sistem yang difokuskan pada perancangan dan implementasi *Learning Management System* (LMS). Sistem ini dikembangkan untuk menunjang pelaksanaan pembelajaran daring secara lebih sistematis, efisien, serta dapat diakses dengan mudah oleh seluruh pengguna yang terlibat.

Dalam tahap implementasi, tim pengembang LMS berupaya membangun sebuah platform yang mampu mengelola materi pembelajaran, mengatur jadwal pelatihan, memantau perkembangan pengguna, serta menyediakan fitur evaluasi

yang saling terintegrasi. Adapun tujuan utama dari proyek ini adalah membentuk ekosistem pembelajaran digital yang efektif guna meningkatkan kualitas dan efisiensi pelatihan bagi pengguna internal.

Guna memastikan sistem LMS berjalan sesuai dengan kebutuhan pengguna serta memenuhi standar kualitas yang telah ditetapkan, proses *Quality Assurance* (QA) dilaksanakan secara komprehensif. Proses QA pada proyek LMS dilakukan melalui serangkaian tahapan pengujian yang terstruktur sebagai berikut:

(1) *Update Requirement* dari User

Proses QA dimulai setelah tim pengembang menerima pembaruan *requirement* dari pengguna. *Requirement* ini menjadi acuan dasar bagi tim QA dalam menyusun skenario pengujian agar sesuai dengan fungsi yang dikembangkan. Requirement dari *user* diberikan kepada pengembang dalam bentuk tabel *spreadsheet* seperti ditunjukkan pada gambar 3.9.

A. Classes Information Detail				
Section	Field	Type	Default	Description
Course Information	Item Title	Text	-	Items Title
	Item ID	Text	-	Items ID
	Item Type	Text	-	Item Type
	Duration	Text	-	Get data Duration (days/hours) Format: x Day / x Hours
	Target Participant	Text	-	Get data Target Participants from item details field Target Participants
	Button - Start/Continue Course - Assign to Team - Review Activity	Button	Start/Continue Course Assign to Team: Active (blue)	* Start Course - Tampil pada user yang memiliki hak untuk start course *Continue Course - Tampil menggantikan button Start Course - Hanya tampil jika kondisi course sedang berjalan/sudah dimulai * Assign to Team Untuk mengassign suatu item (baik terdapat class atau tidak) ke karyawan - Hanya tampil pada user Manager/atasan yang dapat meng-assign team member * Review Activity Untuk mereview activity yang sudah dikerjakan pada E-Learning. Logic review activity = get activity berdasarkan item e-learning yang terupdate. Misal peserta mengerjakan item e-learning A1 dengan activity A,B,C dan setelah dikerjakan ternyata e-learning A1 tadi ada activity tambahan sehingga activitynya menjadi A,B,C,D. Maka ketika peserta review activity yang tampil adalah activity A,B,C,D (Activity Existing + Activity Tambahan terupdate, dengan completion status belum completed, dan ketika karyawan membuka activity D tersebut statusnya menjadi completed / checklist namun tidak mempengaruhi training yang sudah tercatat di Learning History karyawan tersebut)

Gambar 3.9. *Requirement library*

(2) *Scenario Test Cases*

Sambil menunggu proses pengembangan yang dilakukan oleh tim developer, tim QA mulai menyusun skenario pengujian (*test scenario*) serta *test case* dengan mengacu pada *requirement* yang telah diperbarui. Penyusunan skenario seperti ditunjukkan pada gambar 3.10 dikerjakan secara bersamaan

dengan proses pengembangan untuk memastikan bahwa seluruh kebutuhan pengujian telah siap ketika fitur selesai dibuat.

Module	Test Cases	Type	Test Type	Priority	Pre-conditions	Steps Description	Expected Result	Actual Result	Test Status
Menampilkan menu Banners	Via menu tab MyDevelopment	Functional	Positive	High	Login sebagai admin ke LMS	1. Pilih menu Learning Administration 2. Pilih sub menu Application Administration 3. Pilih sub menu Banners	1. Membuka beberapa sub menu, salah satunya Application Administration 2. Membuka beberapa sub menu, salah satunya Banners 3. Redirect user ke halaman menu Banners	1. Membuka beberapa sub menu, salah satunya Application Administration 2. Membuka beberapa sub menu, salah satunya Banners 3. Redirect user ke halaman menu Banners	Passed
	Via link menu Banners	Functional	Positive	High	Login sebagai admin ke LMS	User akses menu Banners menggunakan link -> ...	Redirect user ke halaman menu Banners	Redirect user ke halaman menu Banners	Passed
	Cek komponen halaman menu Banners	UI	Positive	High	User berada di halaman Banners	Perhatikan setiap komponen (section) pada halaman menu Banners	1. Breadcrumb: Home > Learning Administration > Application Administration > Banners 2. Title: Banners 3. Text: Slide show banners every [count] seconds 4. Sections: - Banners Preview - Banners Collection	1. Breadcrumb: Home > Learning Administration > Application Administration > Banners 2. Title: Banners 3. Text: Slide show banners every [count] seconds 4. Sections: - Banners Preview - Banners Collection	Passed
	Cek komponen section Banners Preview	UI	Positive	High	1. User berada di halaman Banners 2. User sudah menambahkan Banner	Perhatikan setiap komponen pada section Banners Preview	1. Title: Banners Preview 2. Slide show image size 1080 x 342px 3. Arrow kiri dan kanan di setiap slide show 4. Banners title di bawah image	1. Title: Banners Preview 2. Slide show image size 1080 x 342px 3. Arrow kiri dan kanan di setiap slide show 4. Banners title di bawah image	Passed
	Cek komponen section Banners Collection (collapsed)	UI	Positive	High	User berada di halaman Banners	Perhatikan setiap komponen pada section Banners Collection	1. Title: Banners Collection 2. Button: + Add Banners 3. Section per Banner: - icon (+) untuk expand collapse section - icon drag untuk memindahkan posisi Banner - icon kebab menu untuk menampilkan opsi delete Banner 4. Buttons: Cancel, Save	1. Title: Banners Collection 2. Button: + Add Banners 3. Section per Banner: - icon (+) untuk expand collapse section - icon drag untuk memindahkan posisi Banner - icon kebab menu untuk menampilkan opsi delete Banner 4. Buttons: Cancel, Save	Passed
	Cek komponen section Banners Collection (expanded)	UI	Positive	High	User berada di halaman Banners	1. Expand salah satu section per Banner 2. Perhatikan setiap komponen pada section Banners Collection	1. Membuka section detail per Banner 2. Fields: - Image (mandatory): upload box (Choose file or drag it here) - URL - Title	1. Membuka section detail per Banner 2. Fields: - Image (mandatory): upload box (Choose file or drag it here) - URL - Title	Passed

Gambar 3.10. Scenario Test Cases

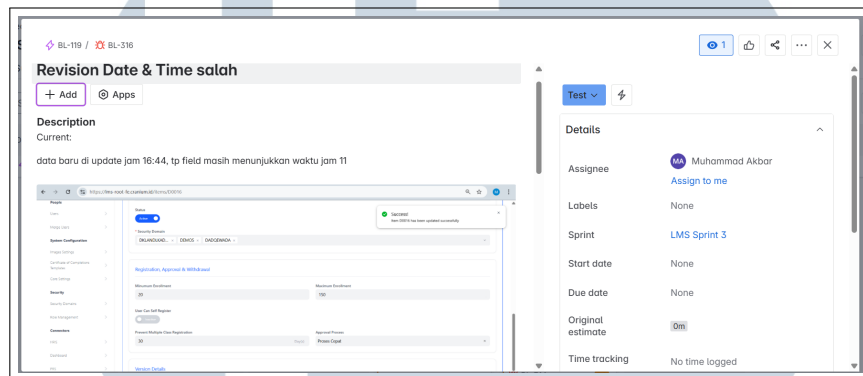
(3) Testing

Setelah pengembangan fitur selesai, tim QA melaksanakan pengujian dengan mengacu pada skenario yang telah disusun sebelumnya. Proses pengujian dapat dilakukan secara manual maupun otomatis, bergantung pada kebutuhan dan kompleksitas fitur. Pada tahap ini, QA memastikan bahwa setiap fungsi bekerja sesuai dengan spesifikasi dan kebutuhan pengguna. Seluruh proses pengujian dilakukan pada lingkungan staging sebagaimana ditunjukkan pada gambar 3.11.

Gambar 3.11. Tampilan Staging LMS

(4) Bug Report

Jika ditemukan ketidaksesuaian atau error, tim QA akan mencatat dan melaporkan temuan tersebut melalui sistem *ticketing* sebagaimana ditunjukkan pada gambar 3.12. Setiap laporan bug (*bug report*) disusun secara rinci, mencakup langkah reproduksi, penjelasan mengenai permasalahan, serta bukti pendukung seperti tangkapan layar.



Gambar 3.12. Jira ticket

(5) Retest Bug

Setelah perbaikan dilakukan oleh tim *developer*, QA kembali melakukan pengujian ulang (*retest*) untuk memastikan bahwa perbaikan telah berjalan dengan benar dan tidak memunculkan *error* baru. Apabila ditemukan *bug tambahan*, QA akan kembali membuat laporan melalui Jira, dan proses ini diulangi hingga seluruh temuan berhasil diselesaikan.

(6) Deliver to User

Setelah seluruh fitur berhasil diuji dan dinyatakan memenuhi kriteria, sistem kemudian diserahkan kepada pengguna untuk menjalani *user acceptance testing* (UAT). Apabila tahap tersebut selesai dengan baik, sistem dapat dilanjutkan ke proses produksi atau *deployment*. Dengan adanya alur QA yang terencana dan terdokumentasi secara menyeluruh, proyek LMS diharapkan mampu menghasilkan sistem yang stabil, sesuai dengan kebutuhan serta harapan pengguna, dan memiliki performa yang optimal saat digunakan.

3.5.2 User Acceptance Testing (UAT) & Defect List

Setelah seluruh rangkaian pengujian internal selesai dilakukan oleh tim QA dan sistem dinyatakan memenuhi standar kualitas, tahap selanjutnya adalah *User Acceptance Testing* (UAT). UAT merupakan proses pengujian yang dilakukan langsung oleh pengguna (user) untuk memverifikasi apakah fitur yang telah dikembangkan benar-benar sesuai dengan kebutuhan bisnis, alur operasional, serta ekspektasi yang telah ditentukan sebelumnya. Pada proyek LMS, tahap UAT menjadi komponen penting karena memastikan bahwa setiap fungsi yang dirancang mampu mendukung proses pembelajaran, pengelolaan materi, serta aktivitas administrasi secara efektif dan tanpa gangguan. Selama pelaksanaan UAT oleh pihak klien, pengguna menjalankan seluruh skenario penggunaan yang telah disiapkan, meliputi proses login, navigasi modul, pengelolaan kelas, unggah materi, serta evaluasi pembelajaran. Hasil dari pengujian ini menjadi dasar utama untuk menilai kelayakan sistem sebelum diimplementasikan pada lingkungan produksi.

Dalam proses UAT, pengguna dapat menemukan berbagai ketidaksesuaian atau isu fungsional yang tidak muncul pada saat pengujian internal. Temuan-temuan tersebut kemudian dicatat dalam *Defect List* seperti pada gambar 3.13, yaitu daftar yang berisi seluruh bug, error, maupun masalah tampilan yang muncul selama pelaksanaan UAT. Pada proyek LMS, defect yang ditemukan biasanya berkaitan dengan alur kerja tertentu yang tidak berjalan sebagaimana mestinya, validasi data yang belum sepenuhnya tepat, tampilan antarmuka yang tidak konsisten, atau respons sistem yang tidak sesuai kebutuhan operasional pengguna. Setiap defect yang tercatat kemudian memiliki status, tingkat prioritas, dan penanggung jawab, sehingga proses perbaikannya dapat dipantau secara terstruktur. *Defect List LMS Phase 1* dibagi menjadi 4 Defect List, yang bertotalkan 1101 bugs. Tahap ini memakan waktu cukup lama karena banyaknya dokumen yang kurang jelas sehingga menyebabkan mispersepsi.

No. Sken (KLA)	No	Created Date	Feature	Description	Type	Priority	Status	Fixed Date (Ready to test)	PIC Cranium
	1043	28-Jul-25	EAP	EAP otomatis hanya launch ke kelas induk saja, kelas yang merefer tidak ter launch EAP nya hasil test, EAP tidak jalan otomatis kedua duanya baik di kelas induk dan kelas refer	Defect	High	Done		Athallah
D.7.3(7) - PSH	1044	28-Jul-25	Classes - Registration	Registration status seharusnya Canceled bukan Canceled (tolong crosscheck ulk wording lainnya)	Defect	Low	Cancel		PRS
	1045	28-Jul-25	Library User	For Me - Search "Anggy" Kemudian pindah ke All Catalog Search apapun ga muncul - nyangkut di pencarian "Anggy"	Defect	High	Done	29/07/2025	Wildan
	1046	28-Jul-25	Item	Ketika item di inactive kemudian save muncul system under maintenance	Defect	High	Done	01/08/2025	Akbar
	1047	29-Jul-25	Notifikasi - EAP	Jumlah count notif tidak sama dengan yang ada di list dan atau tidak terupdate	Defect	Medium	Done		Athallah
	1048	29-Jul-25	Library User	Beberapa behavior aneh 1. Show in all catalog aktif, tapi tidak muncul item : 476 2. Show in all catalog diaktifkan, muncul di library all catalog, lalu dinonaktifkan, tetap muncul item : 476 3. Search harsya, tapi item yang keluar tidak mengandung harsya Sudah normal lagi	Defect	High	Cancel		
	1049	29-Jul-25	Notify Me	Peserta status registered namun peserta ybs masih bisa klik notify me	Defect	High	Done		Theo
	1050	29-Jul-25	Agenda - Attendance (Classes)	*Delete Confirmation Please make sure that related data has been deleted accordingly Are you sure want to delete this data ? masih belum sesuai, di mana kontributor belum memenuhi semua syarat. Di mana baru attendance status aja yang success, HRSI statusnya masih kosong. retest 1 agus : masih belum sesuai ketika kontributor sudah memenuhi syarat, delete confirmationnya masih keluar. retest 4 agus : masih belum sesuai, posisi masih sama.	Defect	Medium	Done		Imam

Gambar 3.13. Defect List Phase 1

Setelah defect tercatat, tim developer akan melakukan perbaikan sesuai prioritas yang ditetapkan. Selanjutnya, tim QA melakukan pengujian ulang untuk memastikan bahwa perbaikan berjalan dengan benar dan tidak menimbulkan error baru. Siklus ini dapat berlangsung beberapa kali hingga seluruh defect dinyatakan terselesaikan dan pengguna memberikan persetujuan final bahwa sistem layak digunakan. Dengan adanya proses UAT dan pengelolaan defect yang terdokumentasi secara menyeluruh, proyek LMS dapat menghasilkan sistem yang stabil, akurat, serta benar-benar sesuai dengan kebutuhan pengguna ketika nanti diterapkan pada tahap produksi.

3.5.3 Rekap Issue Kendala After Implementation

Setelah sistem berhasil diimplementasikan pada lingkungan produksi, tim QA dan developer terus memantau performa sistem untuk memastikan stabilitas dan kehandalan fitur yang telah dikembangkan. Pada tahap ini, beberapa isu baru muncul yang tidak terdeteksi selama fase UAT atau pengujian internal, antara lain:

- Bug terkait integrasi dengan yang hanya terlihat ketika data nyata digunakan oleh pengguna.
- Ketidaksesuaian validasi data pada kondisi tertentu yang jarang terjadi.
- Respons sistem yang lambat saat beban pengguna meningkat.
- Kesalahan tampilan antarmuka pada browser atau perangkat tertentu.

Setiap bug yang muncul di tahap produksi dicatat secara rinci dalam *Production Defect List* dengan informasi status, prioritas, dan penanggung jawab perbaikan. Tim developer melakukan perbaikan sesuai prioritas, kemudian QA melakukan pengujian ulang untuk memastikan bug tersebut telah diperbaiki dan tidak menimbulkan masalah baru. Siklus ini berulang hingga seluruh isu di produksi terselesaikan dan sistem dapat beroperasi secara optimal.

Proses dokumentasi dan perbaikan yang sistematis ini memastikan bahwa proyek LMS tetap stabil, performa sistem terjaga, serta kebutuhan pengguna terpenuhi setelah implementasi.

3.5.4 Enhancement Phase

Fase enhancement merupakan tahap lanjutan setelah implementasi awal sistem LMS, yang bertujuan untuk meningkatkan performa, kestabilan, dan kesesuaian sistem dengan kebutuhan pengguna. Pada *LMS Enhancement Phase I*, beberapa perubahan atau penyesuaian ulang requirement dilakukan berdasarkan masukan pengguna dari fase sebelumnya. Penyesuaian ini meliputi pengoptimalan fitur agar lebih efisien dan sesuai dengan alur kerja yang diharapkan, termasuk memperbaiki bug minor, memperbarui validasi data, dan menyesuaikan antarmuka agar lebih intuitif.

Selain itu, fase ini juga mencakup *system tuning* untuk meningkatkan kinerja pemrosesan, terutama ketika sistem menangani jumlah data yang besar. Optimalisasi dilakukan pada alur pengambilan dan penyimpanan data, serta perbaikan performa pada modul yang sering diakses pengguna. Dengan adanya enhancement ini, sistem diharapkan dapat berjalan lebih cepat, responsif, dan lebih stabil saat digunakan dalam lingkungan produksi.

3.6 Kendala dan Solusi yang Ditemukan

Kendala yang dialami selama masa magang:

- (1) Komponen yang memiliki tingkat kompleksitas tinggi memerlukan waktu pengerjaan yang lebih lama, dan belum tersedia *best practice* yang tepat, sehingga prosesnya menjadi kurang efisien.
- (2) Penggunaan resource yang tinggi menyebabkan *laptop/device* bekerja di luar kapasitasnya sehingga kesulitan menjalankan proyek. Kondisi ini menghambat proses kerja dan menimbulkan kerugian waktu.

- (3) Banyak requirement yang berubah secara mendadak ketika sudah berjalan untuk memperbaiki alur/proses sistem LMS.
- (4) Diskusi dengan klien yang berbeda divisi membuat informasi saling bertabrakan.

Solusi yang ditemukan:

- (1) Memperbaiki komunikasi antar tim agar dapat mempercepat proses pengerjaan.
- (2) Membagi tugas dengan lebih rata agar *laptop/device* yang digunakan tidak bekerja terlalu keras dan membuat *crash*.
- (3) Mempelajari relasi antar dokumen lebih dalam sehingga dapat melihat jika ada dokumen yang saling bertabrakan.
- (4) Memastikan requirement yang diberikan oleh user merupakan dokumen versi ter-mutakhir.

