

BAB 3

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Organisasi

Program magang dilaksanakan pada PT. Adicipta Inovasi Teknologi dengan menjalani posisi *Application Developer*, divisi Implementasi, yang secara terutama mengembangkan pada sisi web yakni, *front-end* dan *back-end*, terhadap produk Finance Mobility Suite milik Adins. Program ini juga dilaksanakan dibawah supervisi Bapak Adi Lie, yang merupakan Division Head dari divisi Implementasi. Bapak Adi Lie berperan dalam memberikan bimbingan, informasi, dan melakukan pengawasan.

3.2 Tugas yang Dilakukan

Program magang ini diawali oleh training untuk memperkenalkan *business flow* dan produk - produk Adins, terutama produk Finance Mobility Suite atau dipersingkat menjadi FIMS. Projek - projek yang dilaksanakan, merupakan pengembangan terhadap FIMS, yang di mana klien tersebut akan mengajukan permintaan perubahan atau *Change request* terhadap produk yang sudah diakuisisi sebelumnya. Aplikasi FIMS yang dikembangkan terdiri dari Mobile Order, Mobile Survey, dan Mobile Collection. Berikut adalah tugas beserta tanggung jawab dikelola:

1. PT. Finance XYZ

- Membuat fitur logging proses pemanggilan API milik vendor oleh API milik FIMS.

2. PT. Bank Alpha

- Pengembangan API untuk pengecekan submisi duplikat pada FIMS Mobile Order.
- Modifikasi front-end untuk laman web "Unassigned Task" pada FIMS Mobile Survey.
- Modifikasi API untuk proses submisi tugas pada FIMS Mobile Survey.

3. PT. Bank Beta

- Modifikasi fitur search dengan menambahkan filter pada laman web "Assign Task" FIMS Mobile Collection.
- Modifikasi fitur search dengan menambahkan filter pada laman web "Download Report" FIMS Mobile Collection.

3.3 Uraian Pelaksanaan Magang

Selama pelaksanaan program magang pada PT. Adicpta Inovasi Teknologi, terdapat beberapa tanggung jawab yang harus dilaksanakan. Tanggung jawab tersebut diuraikan dalam timeline tabel 3.1.



Tabel 3.1. Pekerjaan yang dilakukan tiap minggu selama pelaksanaan kerja magang

Minggu Ke -	Pekerjaan yang dilakukan
1	Melaksanakan program training.
2	Melaksanakan program training.
3	Melaksanakan Proyek Akhir.
4	Melaksanakan Proyek Akhir dan Sidang pada akhir minggu.
5	Mempelajari basis kode FIMS Mobile Collect PT. Finance XYZ
6	Mempelajari basis kode FIMS Mobile Collect PT. Finance XYZ
7	Mempelajari basis kode FIMS Mobile Collect PT. Bank Alpha
8	Pembuatan fitur logging proses pemanggilan API untuk PT. Finance XYZ
9	Pembuatan fitur pengecekan submisi formulir duplikat dan Modifikasi API pada proses submisi tugas pada FIMS Mobile Survey untuk PT. Bank Alpha
10	Modifikasi fitur search dengan menambahkan filter pada laman "Unassigned Task" pada web FIMS Mobile Survey untuk PT. Bank Beta.
11	Revisi filter search pada laman "Unassigned Task" pada web FIMS Mobile Survey untuk PT. Bank Beta.
12	Revisi filter search pada laman "Unassigned Task" pada web FIMS Mobile Survey untuk PT. Bank Beta.
13	Modifikasi front-end untuk laman web "Unassigned Task" pada FIMS Mobile Survey untuk PT. Bank Alpha.
14	Modifikasi fitur search dengan menambahkan filter pada laman "Download Report" pada web FIMS Mobile Survey untuk PT. Bank Beta.

3.3.1 PT. Finance XYZ

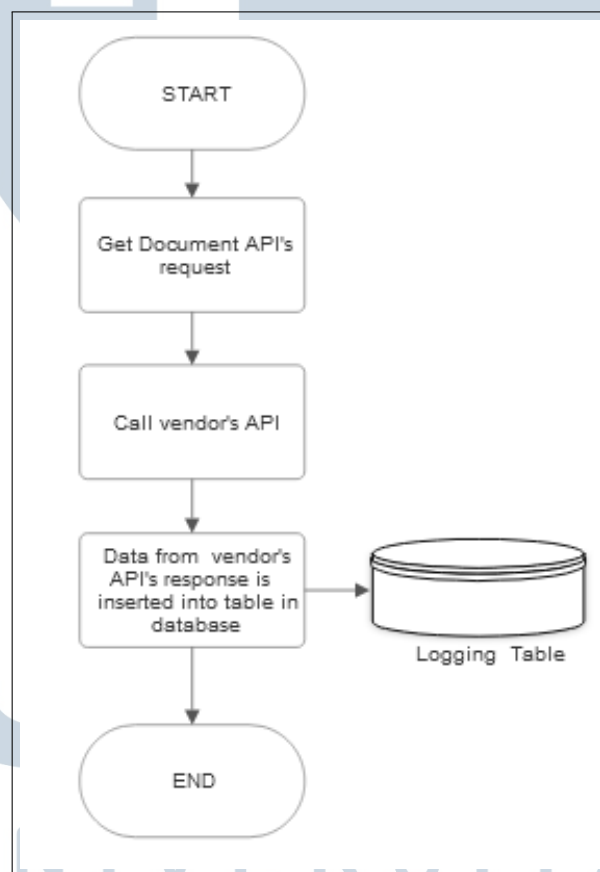
A. Fitur Logging Proses pemanggilan API Vendor

Pembuatan fitur ini bertujuan untuk menyimpan data sejarah atau untuk menyimpan *log* ketika API milik vendor dari PT. Finance XYZ, dipanggil oleh API dari FIMS. API milik FIMS akan dihit ketika user aplikasi mobile melakukan proses yang berhubungan dengan dokumen seperti melakukan penanda tangan digital

terhadap dokumen, cek status approval tanda tangan, dan melakukan download dokumen. Setelah API milik FIMS dipanggil, API tersebut akan memanggil API milik vendor yang akan melakukan, secara aktual, proses - proses yang telah disebutkan sebelumnya. Hasil *response* dari API milik vendor tersebut akan diolah dan disimpan pada tabel di database.

1. Flowchart

Gambar 3.4 menunjukkan flowchart yang memvisualisasikan alur dari proses logging proses pemanggilan API.



Gambar 3.1. Flowchart logging pemanggilan API milik Vendor

Flowchart ini diawali Aplikasi Mobile FIMS Mobile Collection yang akan melakukan *hit* terhadap API pada server. API tersebut akan melakukan pemanggilan API milik vendor dengan mengirimkan JSON *request*. Lalu data dari *response* proses pemanggilan API milik vendor tersebut akan diolah untuk di-*insert* ke dalam tabel dalam database. Tabel tersebut bertujuan untuk

menyimpan data semua proses pemanggilan API seperti tipe API yang di-pemanggilan , JSON *request* dan JSON *response* dari proses pemanggilan , kode status, dll.

2. Implementasi

Gambar 3.2 dan Gambar 3.3 merupakan tabel *logging* yang akan menyimpan hasil proses pemanggilan API milik vendor. *Property* JSON dari *response* pemanggilan API milik vendor akan diolah untuk di-*insert* ke dalam tabel tersebut pada database.

	UUID	API	REQUEST	RESPONSE
1				{\"status\":1,\"code\":1,\"message\":\"test\"}
2				
3				
4				
5				

Gambar 3.2. Tabel Response proses pemanggilan API Vendor Kolom 1-4

	STATUS		DTM_CRT
1	1		

Gambar 3.3. Tabel Response proses pemanggilan API Vendor Kolom Kolom 5-8

Pada tabel, terdapat beberapa kolom yang menyimpan data *logging*. Terdapat kolom "UUID" yang menjadi *primary key* dari setiap baris, kolom "API" yang menyimpan tipe API yang di-*hit*, kolom "REQUEST" menyimpan *string* dari *body request* JSON, kolom "RESPONSE" menyimpan *string body response* JSON, kolom "STATUS" menyimpan nilai kode status dari proses pemanggilan API, dua kolom yang tidak boleh dicantumkan, dan kolom "DTM_CRT" yang menyimpan tanggal dilakukannya proses pemanggilan API.

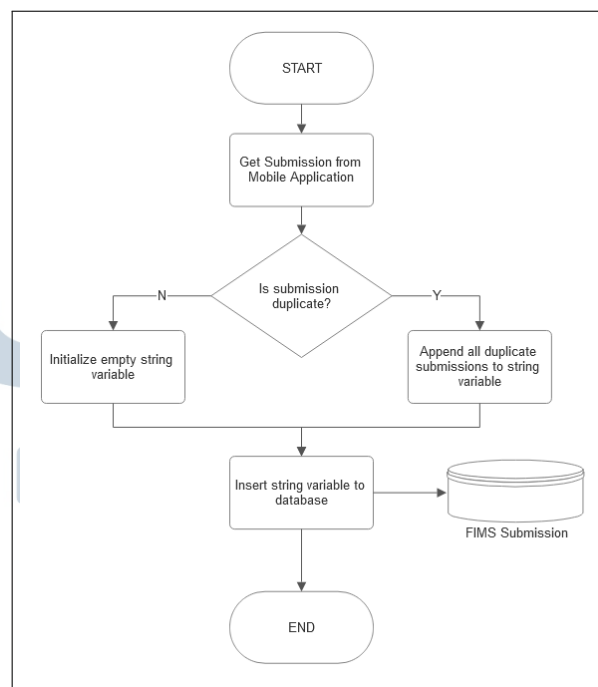
3.3.2 PT. Bank Alpha

A. Pengembangan API untuk pengecekan Submisi Formulir Duplikat pada FIMS Mobile Order

Pembuatan fitur ini bertujuan untuk melakukan pengecekan data formulir yang duplikat dengan melakukan pemanggilan API milik vendor PT. Bank Alpha. Proses pemanggilan tersebut diawali ketika user aplikasi Mobile Order melakukan submisi formulir data yang akan memanggil API milik FIMS. Setelah *request* diterima oleh API FIMS, maka API FIMS akan memanggil API milik vendor, di mana API milik vendor akan melakukan pengecekan terhadap keberadaannya submisi data formulir yang duplikat. Data yang diperoleh dari *response* API milik vendor, akan diolah untuk diubah menjadi 1 string, kemudian akan di-*insert* ke dalam kolom pada database.

1. Flowchart

Gambar 3.4 menunjukkan flowchart yang memvisualisasikan alur dari proses logging proses pemanggilan API.



Gambar 3.4. Flowchart API pengecekan submisi formulir duplikat

Flowchart ini diawali Aplikasi Mobile FIMS Mobile Collection yang akan

melakukan *hit* terhadap API pada server. API tersebut akan melakukan pemanggilan API milik vendor dengan mengirimkan JSON *request*. Lalu data dari *response* proses pemanggilan API tersebut akan diolah untuk di-*insert* ke dalam tabel dalam database. Tabel tersebut bertujuan untuk menyimpan data semua proses pemanggilan API seperti tipe API yang di-pemanggilan, JSON *request* dan JSON *response* dari proses pemanggilan, kode status, dll.

2. Implementasi

Gambar 3.5 merupakan contoh JSON *response* dari hasil pemanggilan API. *Property* JSON dari *response* akan diolah untuk di-*insert* ke dalam tabel pada database.

```
1 {
2   "IsDuplicate": true,
3   "ListData": [
4     {
5       "LeadNo": " ",
6       "LeadStat": " ",
7       "BranchCode": " ",
8       "BranchName": " ",
9       "AppNo": " ",
10      "AppStat": " "
11    },
12    {
13      "LeadNo": " ",
14      "LeadStat": " ",
15      "BranchCode": " ",
16      "BranchName": " ",
17      "AppNo": " ",
18      "AppStat": " "
19    },
20    {
21      "LeadNo": " ",
22      "LeadStat": " ",
23      "BranchCode": " ",
24      "BranchName": " ",
25      "AppNo": " ",
26      "AppStat": " "
27    }
28  ],
29  "HeaderObj": {
30    "ResponseTime": "46 ms",
31    "StatusCode": "200",
32    "Message": "Success",
33    "ErrorMessages": null
34  },
35  "StatusCode": "200",
36  "Message": "Success",
37  "ErrorMessages": null,
38  "RowVersion": null
39 }
```

Gambar 3.5. Response proses pemanggilan API

Pada gambar 3.5, terdapat JSON *property* yakni, "IsDuplicate", "ListData", "HeaderObj", "StatusCode", "Message", "ErrorMessages", dan "RowVersion". JSON *Property* "IsDuplicate" membawa nilai *boolean*

yang mengindikasikan, apakah sebuah formulir aplikasi memiliki duplikat pada database atau tidak di mana true berarti terdapat duplikat, dan false berarti sebaliknya. JSON *property* "ListData" merupakan sebuah *list* yang dapat terpopulasi dengan detail dari setiap formulir duplikat. JSON *property* "HeaderObj", "StatusCode", "Message", "ErrorMessages", dan "RowVersion" memberikan informasi mengenai status hasil proses pemanggilan API. JSON *property* "ResponseTime" adalah durasi waktu proses pengecekan formulir aplikasi, "StatusCode" memberikan status kode *HTTP* di mana pada gambar 3.5, 200 memiliki arti bahwa proses pemanggilan berhasil. JSON *property* "Message" dan "ErrorMessages" memberikan pesan mengenai status proses pemanggilan .

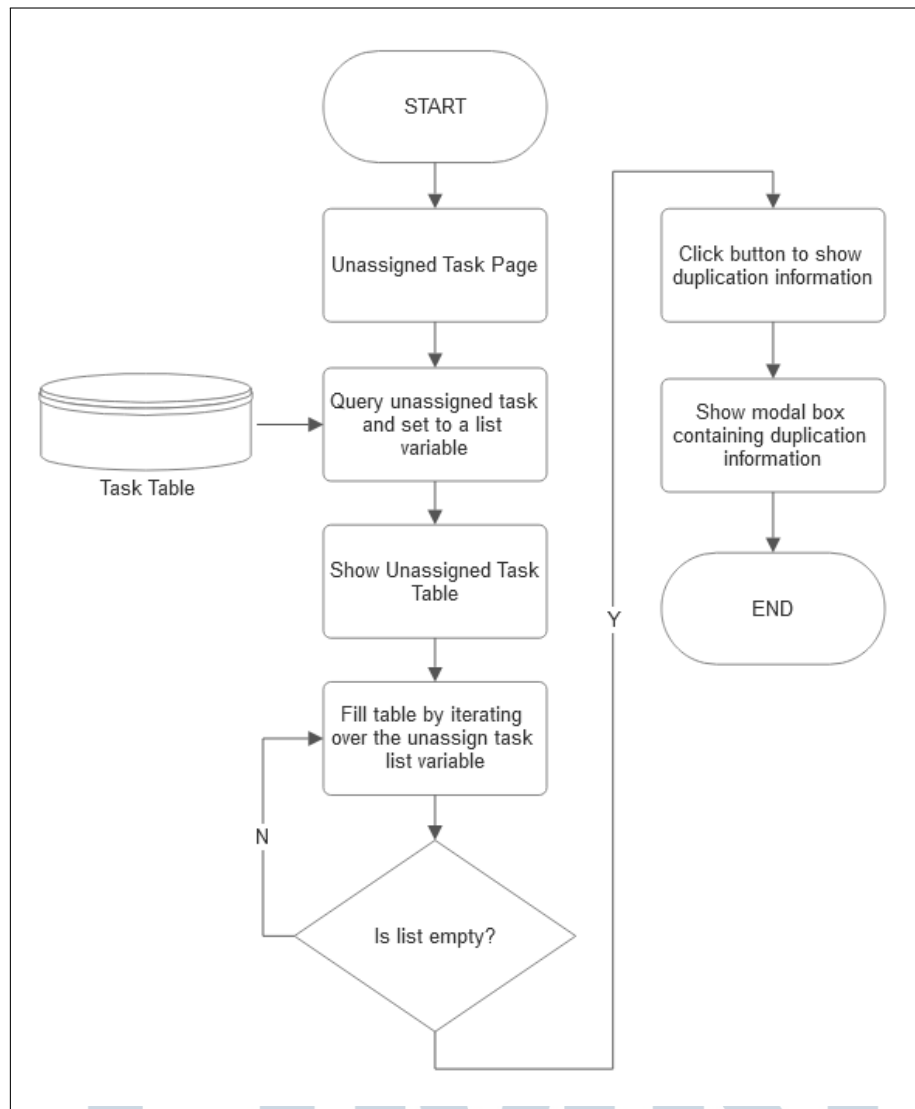
B. Modifikasi Front-End untuk Laman web "Unassigned Task" pada FIMS Mobile Survey

Laman "Unassigned Task" adalah salah satu fitur pada laman web FIMS Mobile Survey yang berfungsi untuk menampilkan tabel yang berisi daftar tugas yang belum ditugaskan kepada pegawai. Modifikasi yang dilakukan bertujuan untuk menambahkan kolom baru pada tabel bernama "Duplicate Information" yang berisi tombol pada setiap baris. Apabila ditekan, tombol tersebut akan menampilkan informasi mengenai apakah terdapat duplikasi tugas. Duplikasi ini berarti tugas tersebut memiliki beberapa data yang sama dengan tugas yang lain.

1. Flowchart

Gambar 3.6 menunjukkan flowchart yang memvisualisasikan alur dari inialisasi laman "Unassigned Task" sampai ditekannya tombol "Duplicate Information".

UNIVERSITAS
MULTIMEDIA
NUSANTARA




Gambar 3.6. Flowchart inialisasi laman "Unassigned Task" sampai pemunculan *Modal Box* untuk "Duplicate Info"

Flowchart ini diawali dengan user yang membuka laman "Unassigned Task". Pada laman ini, terdapat tabel yang akan berisi daftar tugas yang belum ditugaskan kepada pegawai. Pengisian data pada tabel akan diawali dengan mengisi variabel *list* dengan *query* terhadap database untuk mendapatkan daftar tugas yang *unassigned*. Kemudian, setiap elemen pada variabel *list* tersebut akan di-iterasi untuk ditampilkan sebagai baris pada tabel sampai variabel *list* sudah kosong. Ketika tabel sudah terpopulasi, maka pada setiap baris akan memiliki tombol "View Info" pada kolom "Duplicate Info". Apabila tombol tersebut ditekan, maka akan muncul *modal box* yang berisi

tentang informasi duplikasi tugas dari baris task yang dipilih.

2. Implementasi

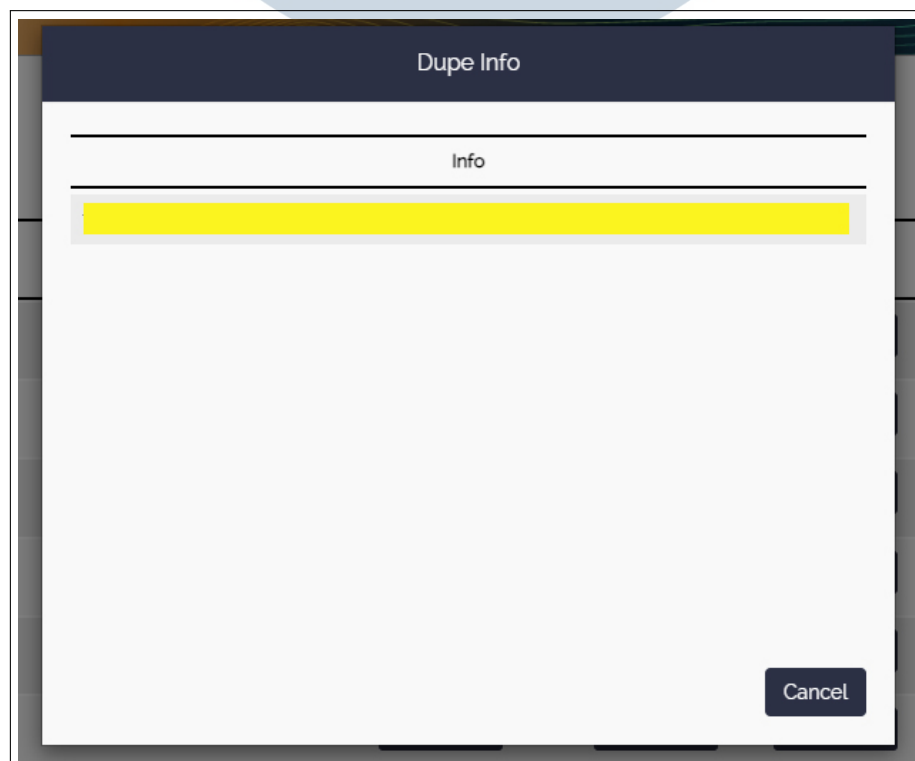
Gambar 3.7 dan gambar 3.8 merupakan gambar hasil dari implementasi modifikasi terhadap tabel "Unassigned Task".



Application Number	Customer Name	Address	Phone	Distance from Nearest Branch	Duplicate Info
					View Info
					View Info
					View Info
					View Info
					View Info

Gambar 3.7. Tabel pada laman "Unassigned Task"

Tabel yang terdapat pada Gambar 3.7 menampilkan daftar *unassigned task*, di mana kolom yang berada di sisi sebelah kanan adalah kolom "Duplicate Info" yang berisikan tombol "View Info".



Gambar 3.8. *Modal Box* untuk menampilkan informasi duplikasi tugas

Gambar 3.8 menunjukkan *modal box* yang ditampilkan ketika tombol

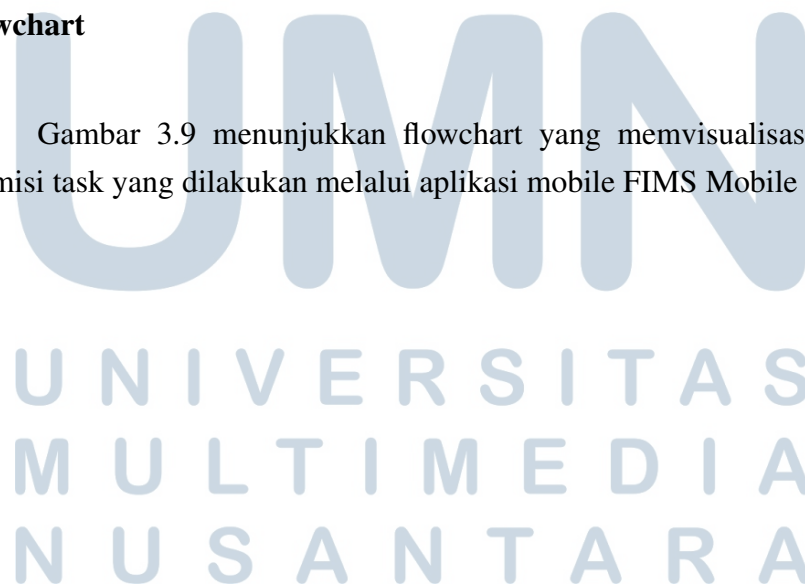
”View Info” ditekan. Di dalamnya, terdapat informasi mengenai detail tugas duplikat dari baris tugas yang dipilih. Yang dimaksud adalah, setiap baris dalam tabel mewakili satu tugas, dan mungkin ada beberapa tugas yang memiliki duplikat. *Modal box* tersebut dimaksudkan untuk menampilkan informasi tentang tugas-tugas yang merupakan duplikat dari tugas yang dipilih. Contoh dari informasi yang dimunculkan adalah ”Terdapat duplikat tugas pada Cabang : X dengan nomor formulir : Y”.

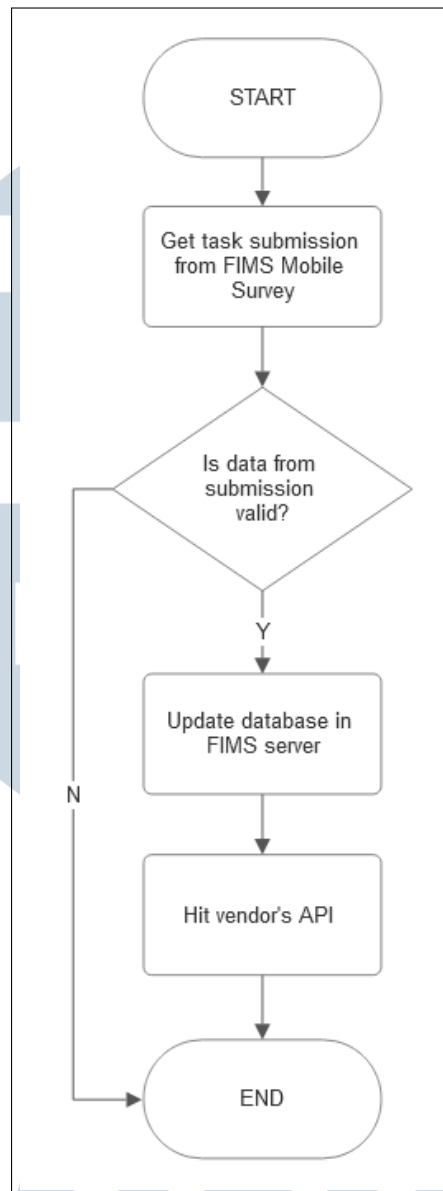
C. Modifikasi API untuk proses submisi tugas pada FIMS Mobile Survey

Pada aplikasi FIMS Mobile Survey, terdapat aktivitas yang dapat dilakukan oleh pengguna, yaitu melakukan submisi tugas. Ketika suatu tugas disubmisi, aplikasi mobile akan memanggil API submisi tugas pada server milik FIMS. Dalam API tersebut, terdapat mekanisme untuk memvalidasi data yang diterima dari tugas yang disubmisi. Setelah semua data tervalidasi, data tugas pada database akan diperbarui dengan data baru, serta dilakukan pemanggilan API milik vendor untuk melakukan pembaruan data pada database di server mereka. Modifikasi API yang dilakukan bertujuan untuk menambahkan properti pada JSON *request* yang dikirimkan ke API milik vendor. Properti tersebut merupakan informasi duplikasi yang telah dibahas pada poin - poin sebelumnya.

1. Flowchart

Gambar 3.9 menunjukkan flowchart yang memvisualisasikan alur submisi task yang dilakukan melalui aplikasi mobile FIMS Mobile Survey.





Gambar 3.9. Flowchart alur submisi tugas dari FIMS Mobile Survey

Alur flowchart diawali dengan user yang melakukan submisi tugas melalui aplikasi FIMS Mobile Survey, lalu aplikasi akan memanggil API submisi tugas pada server milik FIMS. Data yang diterima oleh API tersebut akan divalidasi. Apabila data tidak valid, respons akan diberikan kepada aplikasi FIMS Mobile survey yang berisi pesan error. Sebaliknya, apabila data valid, maka data pada tabel pada server FIMS akan diperbarui. Lalu, setelah data telah diperbarui pada database, maka data submisi tugas dikirim ke API milik vendor. Pemanggilan API milik vendor ini bertujuan untuk

mengsinkronasi data pada database FIMS, dan database milik vendor.

2. Implementasi

Gambar 3.10 merupakan gambar hasil implementasi modifikasi API submisi tugas FIMS Mobile Survey

```
91 "listBouwheer": [],
92 "listGroupUsaha": [],
93 "listAsset": [],
94 "dataAplikasi": {
95   "orderNo": "1234567890",
96   "surveyId": "1234567890",
97   "submitDt": "2023-10-27 10:00:00",
98   "dupeCheckInfo": "test"
99 },
100 },
101 },
102 "listBouwheer": [],
103 "listGroupUsaha": [],
104 "listAsset": [],
105 "dataAplikasi": {
106   "orderNo": "1234567890",
107   "surveyId": "1234567890",
108   "submitDt": "2023-10-27 10:00:00",
109   "dupeCheckInfo": "test"
110 },
111 },
112 "listBouwheer": [],
113 "listGroupUsaha": [],
114 "listAsset": [],
115 "dataAplikasi": {
116   "orderNo": "1234567890",
117   "surveyId": "1234567890",
118   "submitDt": "2023-10-27 10:00:00",
119   "dupeCheckInfo": "test"
120 },
121 },
122 "listBouwheer": [],
123 "listGroupUsaha": [],
124 "listAsset": []
125 },
126 "dataAplikasi": {
127   "orderNo": "1234567890",
128   "surveyId": "1234567890",
129   "submitDt": "2023-10-27 10:00:00",
130   "dupeCheckInfo": "test"
131 },
132 "listBouwheer": [],
133 "listGroupUsaha": [],
134 "listAsset": []
```

Gambar 3.10. JSON *request* untuk pemanggilan API vendor

Setelah modifikasi diimplementasi, maka JSON *request* yang akan dikirimkan ke API milik vendor akan bertambah satu properti yaitu properti "dupeCheckInfo".

3.3.3 PT. Bank Beta

A. Modifikasi fitur search dengan menambahkan filter pada laman web "Assign Task" FIMS Mobile Collection

Laman web "Assign Task" FIMS Mobile Survey merupakan laman web yang memiliki tujuan untuk menyediakan *user interface* bagi pengguna web, agar lebih mudah dalam memberikan tugas kepada pegawai. Untuk memberikan tugas, pengguna web terlebih dahulu harus mengisi beberapa filter untuk mempersempit pilihan pegawai yang akan diberi tugas.

Terdapat lima tipe pengguna, yaitu tipe pengguna "A", "B", "C", "D", dan "E". Akan, tetapi tipe pengguna yang dapat mengakses laman "Assign Task" hanya tipe pengguna "A", "B", "C", dan "D". Setiap tipe pengguna memainkan peran yang berbeda dalam hirarki organisasi, di mana tipe pengguna yang pertama disebut memiliki otoritas tertinggi, dan tipe pengguna yang terakhir disebut memiliki otoritas terendah.

Ketika pengguna membuka laman "Assign Task", jumlah filter yang tersedia akan bervariasi tergantung pada tipe pengguna yang sedang digunakan:

1. Tipe Pengguna A dan B:

- Mereka dapat melihat filter untuk memilih pengguna bertipe "C", "D", dan "E".

2. Tipe Pengguna C:

- Dapat melihat filter untuk memilih pengguna bertipe "D" dan "E".

3. Tipe Pengguna D:

- Dapat melihat filter untuk memilih pengguna bertipe "E".

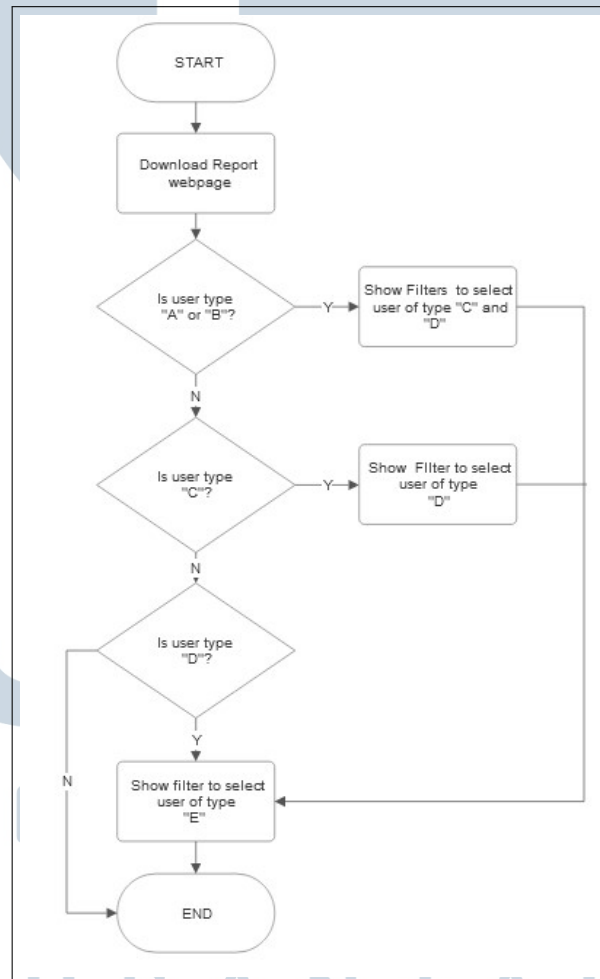
Dapat dilihat, meskipun jumlah filter yang tersedia berbeda-beda untuk setiap tipe pengguna, kesamaannya adalah bahwa setiap tipe pengguna akan melihat filter untuk memilih pengguna dengan tipe "E".

Isi *list of value* pada filter untuk memilih pengguna akan bergantung pada filter atasnya ataupun pengguna yang membuka laman. Yang dimaksud adalah, contoh terdapat pengguna bertipe "C" bernama Ari. Ari memiliki 2 anak buah yakni Yanto dan Anton. Maka, ketika Ari dipilih pada filter untuk memilih pengguna

bertipe "C", isi *list of value* pada filter untuk memilih pengguna bertipe "D" akan terdapat Yanto dan Anton. Contoh Lainnya apabila pengguna bertipe "A" membuka laman "Assign Task", maka filter untuk memilih pengguna bertipe "C" akan terpopulasi dengan semua pengguna bertipe "C" yang menjadi anak buah pengguna "A" tersebut. Modifikasi secara terutama dilakukan dengan tujuan untuk menambahkan filter - filter tersebut.

1. Flowchart

Gambar 3.11 menunjukkan flowchart yang memvisualisasikan alur dari pemunculan filter pada halaman "Assign Task".



Gambar 3.11. Flowchart filter "Assign Task"

Flowchart ini diawali dengan pengguna web yang akan mengakses laman "Assign Task", lalu akan dicek tipe pengguna tersebut.

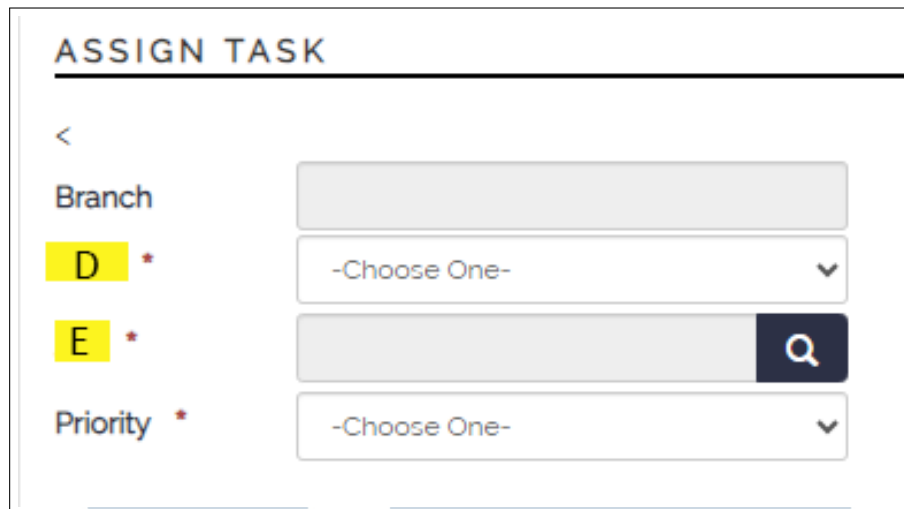
- Apabila user bertipe "A" atau "B", maka akan ditampilkan filter untuk memilih user bertipe "C", filter untuk memilih user bertipe "D", dan filter untuk memilih user bertipe "E".
- Apabila user bertipe "C", maka akan ditampilkan filter untuk memilih user bertipe "D", dan user bertipe "E".
- Apabila user bertipe "D", maka hanya akan ditampilkan filter untuk memilih user bertipe "E".

2. Implementasi

Gambar 3.12, gambar 3.13, dan gambar 3.14 merupakan hasil implementasi dari penambahan filter pada laman web "Assign Task" FIMS Mobile Collection

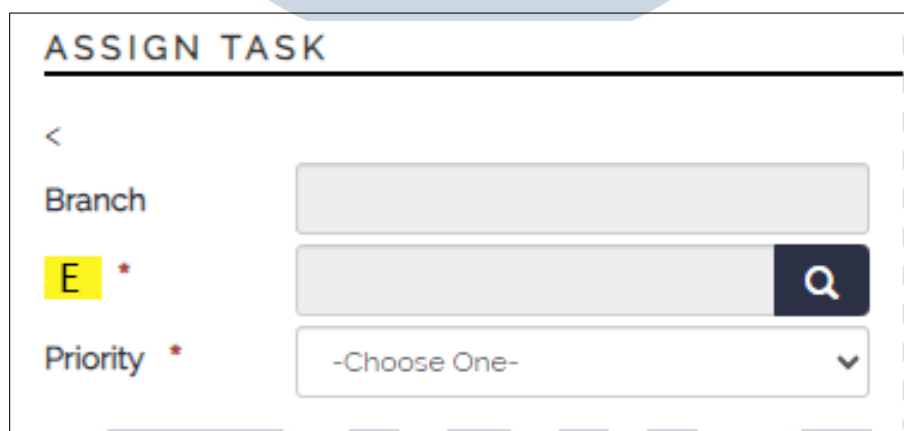
Gambar 3.12. Tampilan Laman "Assign Task" untuk user bertipe "A" atau "B"

Gambar 3.12 menunjukkan filter yang muncul ketika pengguna bertipe "A" atau "B" membuka laman "Assign Task". Hasilnya adalah terdapat 3 filter yang dimunculkan yakni filter pengguna "C", filter pengguna "D", dan filter pengguna "E"



Gambar 3.13. Tampilan Laman "Assign Task" untuk user bertipe "C"

Gambar 3.13 menunjukkan filter yang muncul ketika pengguna bertipe "C" membuka laman "Assign Task". Hasilnya adalah terdapat 2 filter yang dimunculkan yakni filter pengguna "D", dan filter pengguna "E"



Gambar 3.14. Tampilan Laman "Assign Task" untuk user bertipe "D"

Gambar 3.14 menunjukkan filter yang muncul ketika pengguna bertipe "D" membuka laman "Assign Task". Hasilnya adalah terdapat 1 filter yang dimunculkan yakni filter pengguna "E"

B. Modifikasi fitur search dengan menambahkan filter pada laman web "Download Report FIMS Mobile Collection"

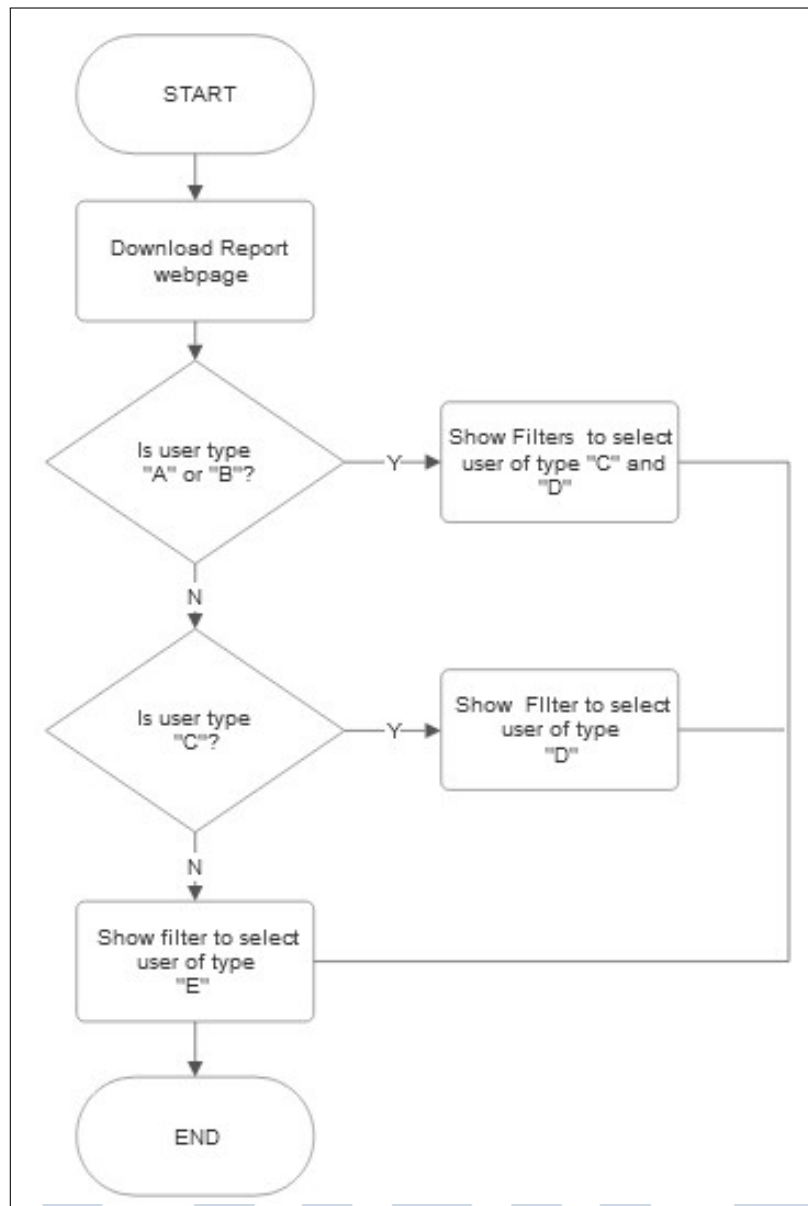
Laman Download Report merupakan laman pada web FIMS Mobile Collection yang memiliki tujuan untuk menyediakan *user interface* bagi pengguna web agar dapat dengan mudah mengunduh file bertipe "Excel" yang berisi informasi tentang tugas yang telah disubmisi oleh pegawai. Untuk memilih data submisi tugas suatu pegawai, maka pengguna web harus mengisi beberapa filter untuk mempersempit pilihan pegawai.

Modifikasi dilakukan dengan menambahkan filter tipe pengguna sesuai dengan hierarki yang telah dijelaskan dalam modifikasi fitur sebelumnya, yaitu pada laman web "Assign Task" FIMS Mobile Collection. Namun, terdapat perbedaan penting dalam akses pengguna terhadap laman "Download Report". Hanya pengguna bertipe "A", "B", dan "C" yang memiliki akses untuk membuka laman "Download Report".

1. Flowchart

Gambar 3.15 menunjukkan flowchart yang memvisualisasikan alur dari pemunculan filter pada halaman "Download Report".





Gambar 3.15. Flowchart filter "Download Report"

Flowchart ini diawali dengan pengguna web yang akan mengakses laman "Assign Task", lalu akan dicek tipe pengguna tersebut.

- Apabila user bertipe "A" atau "B", maka akan ditampilkan filter untuk memilih user bertipe "C", "D", dan "E"
- Apabila user bertipe "C", maka akan ditampilkan filter untuk memilih user bertipe "D", dan "E".

2. Implementasi

Gambar 3.16, gambar 3.17, dan gambar 3.14 merupakan hasil implementasi dari penambahan filter pada laman web "Download Report" FIMS Mobile Collection

DOWNLOAD TASK RESULT	
Assignment Date	Daily
Form	[Redacted]
Form Version	[Redacted]
Branch	All Branch
C	All User
D	All User
E	All User

Gambar 3.16. Tampilan Laman "Download Report" untuk user bertipe "A" atau "B"

Gambar 3.12 menunjukkan filter yang muncul ketika pengguna bertipe "A" atau "B" membuka laman "Download Report". Hasilnya adalah terdapat 3 filter yang dimunculkan yakni filter pengguna "C", filter pengguna "D", dan filter pengguna "E"

DOWNLOAD TASK RESULT	
Assignment Date	Daily
Form	[Redacted]
Form Version	[Redacted]
Branch	All Branch
D	All User
E	All User

Gambar 3.17. Tampilan Laman "Download Report" untuk user bertipe "C"

Gambar 3.13 menunjukkan filter yang muncul ketika pengguna bertipe "C" membuka laman "Download Report". Hasilnya adalah terdapat 2 filter yang dimunculkan yakni filter pengguna "D", dan filter pengguna "E"

3.4 Kendala dan Solusi yang Ditemukan

Pada bagian ini dibahas kendala yang ditemukan selama menjalankan program magang pada PT. Adicipta Inovasi Teknologi seperti sebagai berikut.

1. Kurang pemahaman atas *framework* yang digunakan untuk pengembangan seperti *framework* "Spring" dan "Hibernate".
2. Kurang pemahaman atas alur proses dari berbagai fitur yang sudah termasuk pada basis kode produk.
3. Kesalah pahaman terhadap permintaan perubahan fitur.

Berikut adalah solusi yang ditemukan untuk kendala yang telah disebutkan sebelumnya.

1. Mempelajari ulang materi program training, melakukan pembelajaran mandiri melalui internet, dan bertanya kepada supervisor ataupun kolega.
2. Bertanya kepada supervisor ataupun kolega yang sudah menguasai alur proses dari fitur yang ditanyakan.
3. Melatih diri untuk lebih rinci dan men-detail ketika bertanya mengenai *requirement* dari permintaan perubahan fitur.

