

BAB III

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

Program kerja magang sebagai *Intern Account Data Management* (DaMan) dilakukan di unit *Planning, Performance and Support* (PPS) Divisi *Government Service* (DGS) pada perusahaan PT. Telekomunikasi Indonesia dan ditempatkan ke dalam tim DaMan yang beranggotakan empat orang, dimana tim ini terdiri dari satu orang *Manager* DaMan yaitu Pak Agus Nawan, dua orang *Officer Interfacing* DaMan yaitu Kak Afdhal Afrilliansyah dan Kak Firmansyah Kurniawan Ramdhani dan satu orang *intern* pada DaMan.

DaMan secara umum memiliki ruang lingkup sesuai dengan program studi Sistem Informasi. Dengan tanggung jawab atas sistem pelurusan data dalam bisnis, pembuatan atau penginputan data, pemenuhan permintaan data dari *client* dan *controlling* serta *monitoring* terhadap data-data *customer* dari DGS. Seluruh pembelajaran, pekerjaan, dan tahapan selama program kerja magang ini dibimbing dan diarahkan penuh secara langsung selama program kerja magang berlangsung oleh *Manager* dan *Officer Interfacing* DaMan dari PT. Telekomunikasi Indonesia yang bertanggung jawab secara penuh dalam alur sesuai dengan kebutuhan perusahaan.

Dari adanya program kerja magang yang dilakukan selama delapan ratus jam kerja atau sekitar 142 hari atau 800 jam ini sangatlah bermanfaat dalam membangun *soft skills*, *hard skills* serta berbagai pengalaman dan perspektif baru yang berkaitan dengan dunia kerja secara nyata di perusahaan tempat pelaksanaan program kerja magang. Mendapatkan pekerjaan dan pengalaman yang dapat membantu mendorong mahasiswa agar mampu berpikir secara kritis dan cepat menyelesaikan suatu masalah menjadi ilmu yang di dapatkan selama program kerja magang berlangsung. Selain itu, dengan adanya program kerja magang ini mahasiswa memperoleh pengetahuan lebih tentang pelajaran,

pemahaman dan proses kerja tim DaMan yang sesuai dengan kondisi kebutuhan perusahaan baik secara dokumen bisnis maupun teknis.

3.2 Tugas dan Uraian Kerja Magang

Pada program kerja magang yang dalam waktu kurang lebih delapan ratus (800) jam kerja atau lima (5) bulan, mahasiswa telah membuat *timeline* yang berisi tentang pekerjaan magang yang telah dilakukan pada perusahaan tempat program kerja magang dilakukan dan berhubungan dengan bidang Sistem Informasi pada kepengurusan data-data *customer* yang ada di unit PPS di DGS. Tabel 3.1 dibawah ini merupakan uraian pekerjaan selama program kerja magang pada PT. Telekomunikasi Indonesia dengan periode mulai dari tanggal 15 Agustus 2022 sampai dengan 31 Januari 2023:

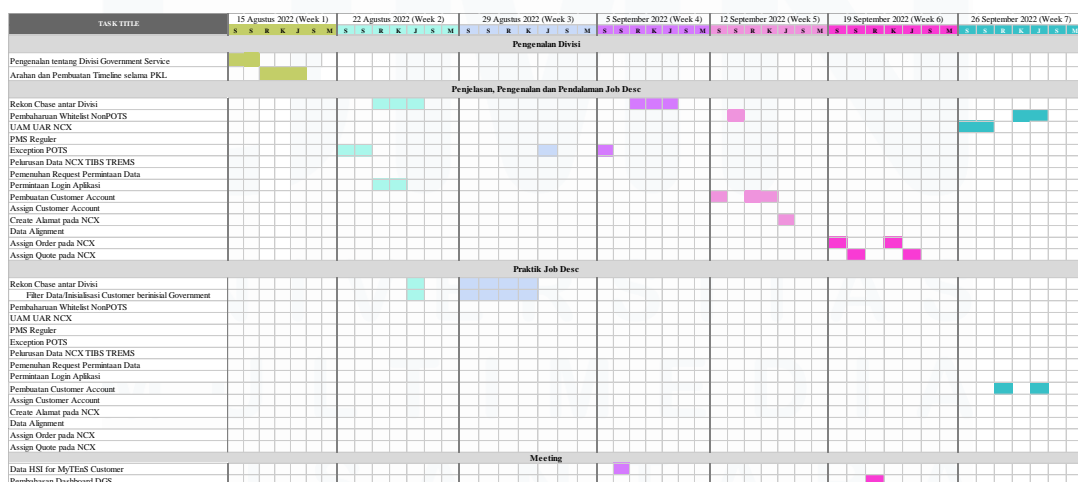
Tabel 3. 1 Tugas Selama Program Kerja Magang

NO.	DESKRIPSI PEKERJAAN	
TAHUNAN		
1	Rekon <i>Customer Base</i> (CBase) tahun 2023 antar divisi.	
	1.1	Penerimaan bahan rekon CBase tahun 2023 POTS dan Non POTS.
	1.2	<i>Filter</i> data/memilih <i>customer</i> yang memiliki nama <i>Government</i> dari bahan rekon CBase tahun 2023.
2	Pembaharuan Whitelist Non POTS.	
	2.1	Penetapan Nota Dinas (NoDin) dari EVP DGS.
TRIWULAN		
3	<i>User Access Matrix</i> (UAM) dan <i>User Access Review</i> (UAR) pada aplikasi NCX.	
	3.1	Data dalformasi DGS.
BULANAN		
4	<i>Performance Management System</i> (PMS) reguler.	
	4.1	Mendapatkan <i>source</i> data PMS POTS dan Non POTS dari Divisi IT (DIT).
	4.2	Melengkapi <i>Mapping</i> berdasarkan <i>ACCOUNT_NUM</i> yang dilakukan oleh segment.
	4.3	Pembuatan <i>table summary</i> untuk dijadikan excel <i>closing</i> PMS.
5	Exception POTS.	
	5.1	Eksekusi <i>source</i> data ke <i>database</i> .
	5.2	Pencarian NCLI dan pengecekan duplikasi data dari nomor telepon.
	5.3	<i>Insert</i> data ke <i>table</i> eksepsi POTS dan SAP.
HARIAN		
6	Pelurusan data di aplikasi NCX, TIBS dan TREMS.	
	6.1	Ticket antrian terkait pelurusan data.
	6.2	Memastikan kelurusan data di semua aplikasi (NCX, TIBS dan TREMS).
7	Pemenuhan <i>request</i> permintaan data.	
8	Permintaan <i>login</i> aplikasi.	

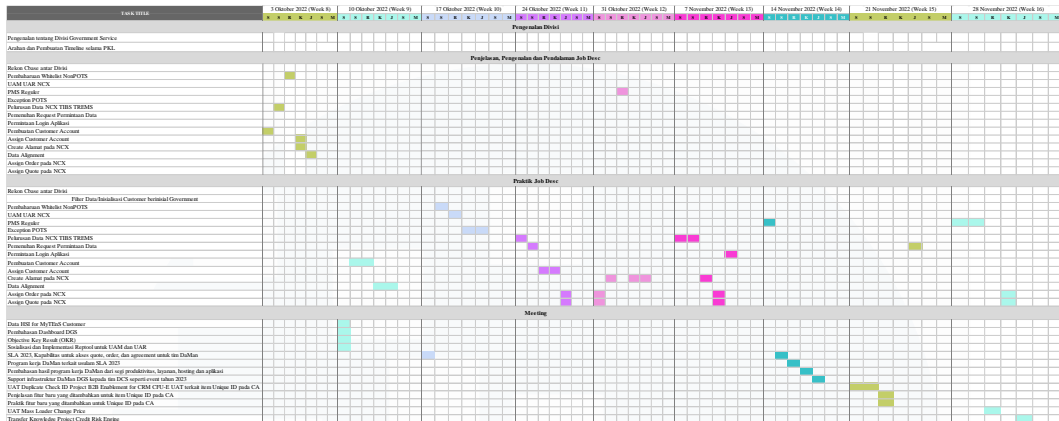
NO.	DESKRIPSI PEKERJAAN	
9	Pembuatan <i>Customer Account</i> (CA).	
	9.1	Pengecekan table CBase
	9.2	Pembuatan CA sesuai kebutuhan data.
10	<i>Assign Customer Account.</i>	
11	<i>Create alamat di aplikasi NCX.</i>	
12	<i>Data Alignment.</i>	
	12.1	Mendapatkan <i>data source</i> dari NoDin <i>Digital Connectivity Service</i> (DCS) ke <i>Business Unit Division</i> (BUD).
	12.2	Validasi data yang diberikan dengan melakukan <i>profiling</i> .
	12.3	<i>Feedback</i> hasil konfirmasi dari tim <i>Revenue Assurance & Quality Management</i> (RevAs) dan segment.
	12.4	<i>Report</i> ke unit DCS terkait hasil konfirmasi dan eksekusi data.
13	<i>Assign order di aplikasi NCX.</i>	
14	<i>Assign quote di aplikasi NCX.</i>	

3.2.1 Tugas Kerja Magang

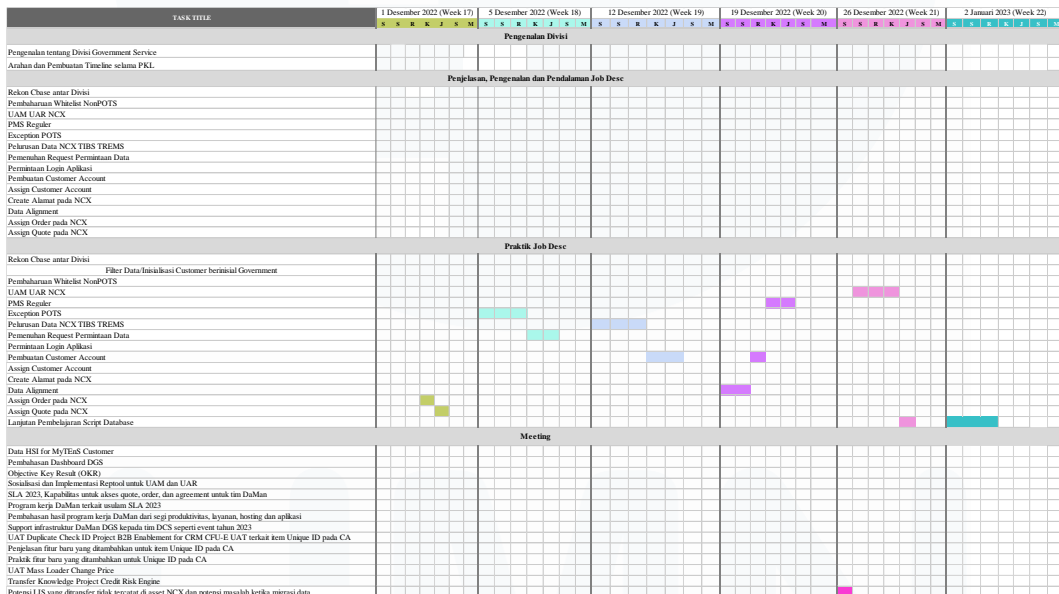
Berikut merupakan waktu pelaksanaan dari uraian kerja magang yang dilakukan oleh mahasiswa:



Gambar 3. 1 Uraian Kerja Magang Bulan Agustus-September



Gambar 3. 2 Uraian Kerja Magang Bulan Oktober-November



Gambar 3. 3 Uraian Kerja Magang Bulan Desember-Januari

3.2.2 Uraian Kerja Magang

A. Tahunan

A.1 Rekon Customer Base (CBase) tahun 2023 antar divisi

Customer Base (CBase) adalah daftar atau *list* konsumen yang ada di Divisi *Government Service* (DGS), jadi rekon CBase merupakan penetapan untuk *list* konsumen-konsumen yang ada di DGS untuk tahun yang akan datang. Pembuatan rekon CBase

dilakukan satu tahun sekali berdasarkan bahan rekon CBase yang telah diberikan oleh segment. Pada PT. Telekomunikasi Indonesia terdapat 3 divisi yaitu Divisi *Government Service* (DGS), Divisi *Enterprise Service* (DES) dan Divisi *Business Service* (DBS). Bahan rekon CBase ini berisi tentang data dari masing-masing divisi yang akan digunakan sebagai *customer* untuk divisi tersebut di tahun yang akan datang. Dikarenakan mahasiswa berada dalam DGS, maka tim DaMan DGS melakukan cek dan *filter* data dari divisi DES dan DBS. Apabila dalam data tersebut masih ada *customer* yang memiliki nama-nama *Government* seperti POLRES, apabila ada maka tim DaMan berhak untuk mengambil *customer* tersebut dari divisi yang bersangkutan, baik dari data konsumen POTS (produk yang berisi *Wireline* dan Indihome) atau Non POTS (produk yang berisi tentang layanan ASTINet, VPN, dan IP Transit). Tahapan dalam *filter* data dalam rekon CBase untuk pengambilan data yang berinisial *Government* dapat dijelaskan sebagai berikut:

1. Tim DaMan melakukan query untuk mendapatkan source data dari database SQL divisi lain yaitu divisi DES dan DBS. Gambar 3.4 merupakan contoh query untuk data Non POTS guna mengambil source data dari SQL divisi lain.

```

drop table tmp33334_1 purge;
commit;
create table tmp33334_1 as
SELECT
  case
    when (substr(a.ba_lock,1,4) = 'T711') then 'DES DES'
    when (substr(a.ba_lock,1,4) = 'T941') then 'DES MCS'
    when (substr(a.ba_lock,1,4) = 'T942') then 'DES FMS'
    when (substr(a.ba_lock,1,4) = 'T944') then 'DES HWS'
    when (substr(a.ba_lock,1,4) = 'T945') then 'DES MMS'
    when (substr(a.ba_lock,1,4) = 'T946') then 'DES CTS'
    when (substr(a.ba_lock,1,4) = 'T947') then 'DES RDS'
    when (substr(a.ba_lock,1,4) = 'T948') then 'DES ERS'
    when (substr(a.ba_lock,1,4) = 'T949') then 'DES BMS'
    when (substr(a.ba_lock,1,4) = 'T950') then 'DES PNBS'
    when (substr(a.ba_lock,1,4) = 'T952') then 'DES EMS'
    when (substr(a.ba_lock,1,4) = 'T955') then 'DES MLS'
    when (substr(a.ba_lock,1,4) = 'T957') then 'DES TMS'
    when (substr(a.ba_lock,1,4) = 'T959') then 'DES RBS'
    when (substr(a.ba_lock,1,4) = 'T940') then 'DES AFS'
    when (substr(a.ba_lock,1,4) = 'T956') then 'DES THS'
    when (substr(a.ba_lock,1,4) = 'T958') then 'DES IBS'
    when (substr(a.ba_lock,1,4) = 'T932') then 'DBS TRB'
    when (substr(a.ba_lock,1,4) = 'T933') then 'DBS HTB'
    when (substr(a.ba_lock,1,4) = 'T934') then 'DBS MCB'
    when (substr(a.ba_lock,1,4) = 'T935') then 'DBS CCS'
    when (substr(a.ba_lock,1,4) = 'T937') then 'DBS LEF'
  end SEGME ASAL,
  a.account_num,
  a.customer_ref,
  a.account_name,
  a.ba_tibs,
  a.ba_lock
FROM v_account_lock a
where substr(a.ba_lock,1,3) not in ('T96','T92');
commit;

SELECT
*
FROM TMP33334_1 a;

```

Gambar 3. 4 Query pengambilan data Non POTS

2. Setelah seluruh data didapat maka dilakukanlah inisiliasi atau *filter* data secara manual untuk pengecekan *customer-customer* yang berinisial *Government*.
3. Hasil akhir pada inisialiasi atau filter data rekon CBase dapat digambarkan pada gambar 3.5

SEGME	CGEST	NCLI	SND	NDOS	NAMA_SND	NAMA	ALAMA
DES					POLRES	POLRES	
DES					POLRES	POLRES	

Gambar 3. 5 Hasil akhir inisialiasi pada rekonsiliasi data

Pada contoh gambar 3.5 dapat dilihat bahwa pada data di divisi DES masih terdapat *customer* yang berinisial *Government* yaitu POLRES, maka tim DaMan DGS dapat mengambil *customer* tersebut dari DES. Ketika seluruh data baik POTS dan Non POTS dari berbagai divisi dan segment sudah selesai di kumpulkan maka data hasil *filter* tersebut diberikan ke bagian direktorat setelah itu di eksekusi di

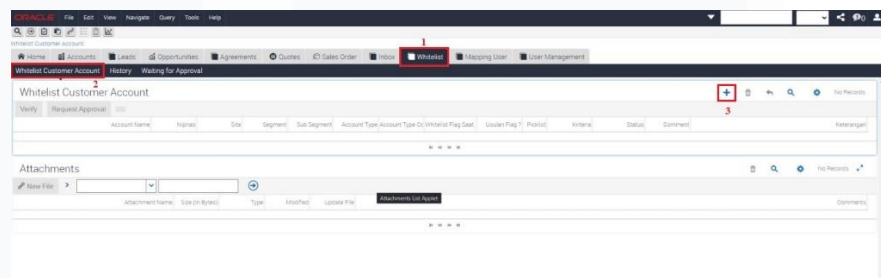
mesin NCX, TIBS, TREMS dan OSS agar datanya menjadi *update*.

A.2 Pembaharuan Whitelist Non POTS

Tujuan dari pembaharuan Whitelist Non POTS adalah memberikan pengecualian kepada layanan komunikasi di *Government Customer* (GC) agar layanan tersebut tidak boleh mati walaupun telat melakukan pembayaran. Layanan ini merupakan salah satu fasilitas yang diberikan oleh PT. Telekomunikasi Indonesia kepada GC yang bersifat *Very Important Person* (VIP) atau *Very Very Important Person* (VVIP). Sebagai contoh yaitu nomor dari Istana Negara, nomor ini tidak boleh mati satu (1) detik pun walaupun telat dalam melakukan pembayaran. Namun tidak semua nomor GC yang ada bukan tidak membayar atau telat membayar, akan tetapi pembayaran yang dilakukan masuk ke dalam dana Anggaran Pemerintah dan Belanja Negara (APBN) atau dana Anggaran Pemerintah dan Belanja Daerah (APBD), dimana pembayaran ini pasti akan mengikuti dana APBN dan APBD yang sudah ditetapkan oleh pemerintah misalkan dana APBN yang ditetapkan adalah enam (6) bulan atau 1 tahun sekali. Data pembaharuan Whitelist ini ditetapkan 1 tahun sekali dari unit *Government Program & Operation* (GPO), lalu dari unit GPO dibuatkan *draft* ke *Executive Vice President* (EVP) kemudian EVP membuatkan NoDin kepada DGS. Proses pembaharuan Whitelist Non POTS dilakukan menggunakan pengecekan berdasarkan No CA dan dapat digambarkan dengan tahapan sebagai berikut:

1. Membuka aplikasi NCX yaitu aplikasi utama dalam database yang digunakan oleh tim DaMan. Seluruh tahapan pada *Whitelist* dapat dilihat pada gambar 3.6.

2. Setelah masuk ke dalam aplikasi NCX, maka tim DaMan memilih menu *Whitelist*.
3. Apabila sudah memilih menu *Whitelist*, selanjutnya tim DaMan memilih menu *Whitelist Customer Account*.
4. Apabila sudah sampai di menu *Whitelist Customer Account* maka tim DaMan akan melakukan *Whitelist* dengan memasukkan memilih tanda + untuk memasukkan No CA yang dituju.



Gambar 3. 6 Menu Whitelist

Pada gambar 3.6 dapat dilihat terdapat kolom Usulan Flag ?, kolom ini lah yang dipilih oleh tim DaMan dengan proses Flag – Unflag – Flag dan kolom Kriteria dengan pilihan APBN atau APBD. Lalu tahapan selanjutnya adalah melakukan *upload* dokumen di bagian *Attachment* untuk melakukan *upload* Nota Dinas penetapan *Whitelist* dari EVP dan yang terakhir adalah *approval* yang dilakukan oleh Manager DaMan terkait *Whitelist* yang telah dilakukan.

B. Triwulan

B.1 User Access Matrix (UAM) dan User Access Review (UAR) pada aplikasi NCX

Pada dasarnya UAM adalah dokumen yang berisi pasangan antara jabatan dengan *role* di aplikasi NCX sesuai dengan data *Human Resource* (HR) yang berisi tentang bahan pendukung untuk untuk membuat dokumen UAR. Untuk melakukan UAM dan UAR, maka PT. Telekomunikasi Indonesia memiliki aplikasi C35 yaitu aplikasi *control*

pendukung untuk melakukan *upload* dokumen-dokumen pendukung untuk permintaan *create login* di aplikasi NCX yaitu pembuatan akun untuk user agar dapat melakukan pengaksesan ke aplikasi NCX., *update responsibility/position* yaitu pemberian batasan akses kepada *user* ketika melakukan akses aplikasi NCX. dan *delete user* yaitu penghapusan akun dari *user* yang tidak pernah aktif atau *login* ke aplikasi NCX dalam kurun waktu 90 hari atau 3 bulan, karena jika tidak ada penghapusan *user* yang sudah tidak aktif maka akan menjadi kerugian bagi perusahaan. Dokumen yang dibutuhkan untuk pembuatan UAM adalah dalformasi DGS setiap triwulan dan data responsibility yang sudah ditetapkan oleh EBIS dan DIT. Untuk contoh dokumen pada UAM dapat digambarkan sebagai berikut:

JABATAN	PTI Account Manager	PTI Address Administrator	PTI Bidding Team	PTI Finance Team	PTI General Manager	PTI Project Management	PTI Revenue Assurance	PTI Sales Engineer	PTI Sales Manager	PTI Sales Quota Administrator	PTI Solution Development & Partnership	PTI User Performance	PTI Customer Data Management	PTI Collection Team	PTI View Order	PTI Manager Customer Data Management
MGR ACCOUNT DATA MANAGEMENT																
OFF 1 IT INTERFACING																
OFF 1 ACCOUNT DATA OPERATION																
OFF 2 IT INTERFACING																

Gambar 3. 7 Dokumen UAM

Pada gambar 3.7 dapat dilihat bahwa jabatan dan *responsibility* yang sudah ditetapkan oleh EBIS dan DIT dalam pengaksesan aplikasi NCX. Jadi ketika tim DaMan ingin membuat *account* di aplikasi NCX tentu dibutuhkan jabatan dan *responsibility* apa saja yang dapat diakses oleh *user* tersebut ketika mengakses aplikasi NCX. Sebagai contoh pada gambar 3.7, mahasiswa yang merupakan intern DaMan diberikan *role* PTI *Address Administrator* dan PTI *Customer Data Management* dalam pengaksesan aplikasi NCX. Untuk penjelasan dari setiap *responsibility* dapat dijelaskan sebagai berikut:

- PTI *Account Manager*: *role* untuk dapat melakukan input order transaksi yang meliputi data mentah (*lid*) – nego

(*opportunity*) – proses input detail dari order yang isinya berupa *billing*, *bandwith* dan sistem pembayaran (*quote*) – data pesanan dari customer yang sudah pasti untuk menjadi pelanggan di perusahaan (*order*).

- *PTI Address Administrator: role* untuk dapat melakukan *create* alamat.
- *PTI Bidding Team: role* untuk dapat melihat atau memproses kontrak yang sudah ada.
- *PTI General Manager: role* untuk dapat melakukan *approval* apabila ada perubahan harga secara tiba-tiba.
- *PTI Revenue Assurance: role* untuk dapat menjamin *revenue* secara legal dan meminimalisir terjadinya *fraud*.
- *PTI Sales Engineer/Offering: role* untuk dapat membantu dalam pembuatan proposal penawaran terkait penawaran produk, harga, dan lain-lain dan diusulkan ke *Bidding Team* untuk dilakukan *approval*.
- *PTI Sales Manager: role* untuk dapat melakukan *approval* kelengkapan dokumen dari kontrak, *offering* atau harga produk dan lain-lain.
- *PTI User Performance: role* untuk melihat performansi dari sisi *view order* transaksi dan yang *action* yang dilakukan adalah melihat harga apakah harga atau transaksi tersebut sudah *complete* atau belum.
- *PTI Customer Data Management: role* untuk dapat melakukan *create* CA dan Whitelist.
- *PTI Collection Team: role* untuk dapat melihat transaksi yang berkaitan secara finansial.
- *PTI View Order: role* untuk melihat *view order* atribut atau *support* yang ada di *customer*.

- *PTI Manager Customer Data Management*: role untuk dapat melakukan *approval* pada *Whitelist*.

Kemudian untuk UAR adalah dokumen *control* yang berisi *review user* data organik dan non organik. Maksud dari data organik adalah data-data yang isinya karyawan perusahaan yang memiliki NIK perusahaan dan untuk data non organik adalah data-data yang isinya karyawan yang belum memiliki NIK, biasanya karyawan-karyawan yang sifatnya eksternal atau sedang melakukan dinas di perusahaan PT. Telekomunikasi Indonesia dimana karyawan ini berasal dari luar perusahaan jadi tidak memiliki NIK perusahaan tetapi membutuhkan akses untuk *login* ke dalam aplikasi NCX. Pada pengisian dokumen UAR yang dilakukan oleh tim DaMan, dokumen pendukung yang dibutuhkan oleh tim DaMan untuk mengisi dokumen UAR kepada user organik hanyalah NIK saja dan untuk *user* non organik dokumen pendukung yang harus diberikan kepada tim DaMan dalam pengisian dokumen UAR adalah.

Tujuan dari UAR adalah melakukan pengecekan pada aplikasi NCX apakah *user-user* yang telah dibuatkan UAM nya masih aktif menggunakan aplikasi NCX, karena jika ternyata *user* yang diberikan tidak digunakan dengan baik maka akan menjadi kerugian bagi perusahaan. Tim DaMan akan melakukan penghapusan kepada *user-user* yang memiliki kriteria *review* yaitu apabila *user* tersebut pensiun, tidak melakukan *login* aplikasi NCX selama 90 hari dan adanya *modify responsibility user* sesuai dengan jabatan atau tidak. Untuk contoh UAR yang telah dikumpulkan dari tim DaMan dapat dilihat sebagai berikut:

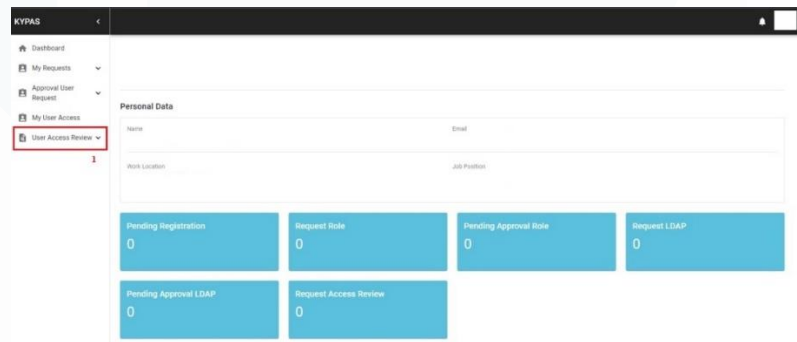
Login	Name	Created	Last Login	Position	Loker	Tanggal P/BPO	Role	Sesuai UA	Sesuai Pei	Exception	Exception	Keterangan	Hasil Review
		2017-04-31	2022-06-31		DIV-DGS	DIV-DGS		Ya	Tidak	None			User Active

Gambar 3. 8 Contoh data organik

NO LOGIN	FULLNAME	RESP	CREATEDON	LASTLOGIN	USERGRUP	JABATAN	DIVISI	STATUS KARYAWAN	TANGGAL PERIKHRABPO	SESUAI UAM	SESUAI PENUGASAN	KETERANGAN	HASIL REVIEW
13537102028	INDA WIRASORIO	PT Account	31/12/2021	31/03/2022	DSS RPS Supp	Mitra	IT/IT-02	Mitra aktif		00%	0%		(Non-organik Adjektif Pengesah)

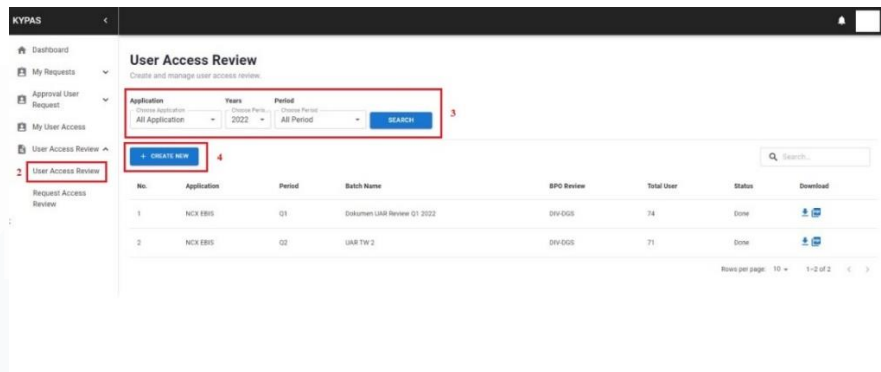
Gambar 3. 9 Contoh data non organik

Akan tetapi, untuk saat ini PT. Telekomunikasi Indonesia sudah melakukan pengecekan *review* terhadap *user* melalui aplikasi *tgkypas.com* dengan memilih menu *User Access Review* (UAR) pada aplikasi NCX seperti gambar 3.10. Menu UAR pada gambar 3.10 digunakan untuk mengunduh data-data akun yang menggunakan aplikasi NCX, sehingga dari data tersebut dapat diputuskan apakah data tersebut masuk ke dalam kategori akun aktif atau *deletion*.



Gambar 3. 10 Menu UAR

Setelah itu, tahapan kedua dapat dilihat pada gambar 3.11, tim DaMan melakukan pengaturan pada *application*, *years* dan *period time* dari *review* yang diinginkan. Setelah semua dipilih maka DaMan dapat membuat dokumen baru dengan memilih *create new* dan DaMan sudah dapat melakukan *review* langsung pada aplikasi tersebut.



Gambar 3. 11 Contoh tampilan untuk review

Lalu pada gambar 3.12 merupakan contoh hasil akhir dari review pada matrix UAR.

LOGIN	FULLNAME	RESP	CREATION	LASTLOGIN	USERGRUP	JABATAN	DIVISI	STATUS KARYAWAN

Gambar 3. 12 Contoh hasil akhir review matrix UAR (1)

TANGGAL PERKIRAAN PENSIUN	BPO	SESUAI UAM	SESUAI PENUGASAN	KETERANGAN	HASIL REVIEW
	DGS	Ya	Tidak	User Aktif	Sesuai UAM

Gambar 3. 13 Contoh hasil akhir review matrix UAR (2)

Akan tetapi apabila *user* tersebut sudah masuk ke dalam salah satu kriteria dalam *deletion user*, untuk *deletion user* ini ditetapkan apabila akun atau *user* tersebut sudah tidak aktif selama 90 hari di aplikasi NCX sehingga dapat menimbulkan kerugian dalam segi biaya kepada perusahaan. Apabila akun tersebut masuk ke dalam kategori *deletion user* maka keterangan dan hasil review dalam matrix UAR adalah

KETERANGAN	HASIL REVIEW
Tidak Login > 90 hari	Deletion Tidak Login > 90 hari

Gambar 3. 14 Contoh keterangan untuk *deletion user*

Lalu jika semua *user* selesai di cek dan dibuatkan keterangannya, akan dilaporkan pada halaman paling bawah matrix UAR seperti gambar 3.15.

Berdasarkan Tabel User Access Review (UAR) di atas, dapat disimpulkan hal-hal seperti dibawah ini:

(i)	Jumlah user yang di review	:	70	Orang
(ii)	Jumlah user aktif sesuai UAM	:	64	Orang
(iii)	Jumlah user aktif dan pengasan	:	0	Orang
(iv)	Jumlah user diusulkan remove karena tidak pernah login > 90 hari	:	5	Orang
(v)	Jumlah user diusulkan remove karena mutasi/ promosi/ pensiun/resign	:	1	Orang
(vi)	Jumlah user aktif yang perlu modifikasi Role	:	1	Orang
(vii)	Jumlah user aktif yang perlu modifikasi position	:	0	Orang
(viii)	Jumlah user yang seharusnya aktif	:	64	Orang

Gambar 3. 15 Contoh keterangan pada matrix UAR

Seluruh dokumen dari UAM dan UAR yang telah dibuat dapat di *control*, disimpan dan dicek dalam aplikasi C36.

C. Bulanan

C.1 Performance Management System (PMS) Reguler

Tujuan dilakukannya PMS Reguler ini adalah mengisi kolom-kolom yang kosong pada data TIBS_NP (yang diberikan oleh tim DIT). Kolom-kolom yang harus diisi oleh tim DaMan adalah NIPNAS GROUP, STANDARD NAME NIPNAS GROUP, WITEL SHIP & WITEL SOLD. Pada PMS regular juga terdapat 2 bagian yaitu PMS POTS dan PMS Non POTS, untuk PMS Non POTS data yang dibutuhkan sebagai bahan pendukung dalam pembuatan PMS Reguler adalah data-data dari TIBS NP dan data selisih SAP yang diberikan oleh tim PPA sedangkan untuk PMS POTS bahan pendukung yang dibutuhkan adalah data TIBS NP dan SISKA yang diberikan oleh tim DIT. Dalam pembuatan PMS Reguler Non POTS tim DaMan melakukan beberapa tahapan, dan query-query dibawah ini merupakan contoh dari tahapan pengambilan PMS Reguler pada PMS Non POTS:

1. Tim DaMan pertama kali akan mendapatkan *source* data PMS Non POTS oleh DIT dan tim PPA setiap awal bulan.

2. Kemudian dari *source* data tersebut, pada intinya tim DaMan akan melengkapi kolom-kolom yang kosong yaitu kolom NIPNAS GROUP, STANDARD NAME NIPNAS GROUP, WITEL SHIP & WITEL SOLD atau dengan kata lain dilakukan *Mapping* dari *source* data dengan melakukan query pada SQL Developer. Mapping pada PMS Reguler dibagi menjadi beberapa tahap, yang pertama adalah Mapping GC, dibawah ini merupakan query dan penjelasan singkat dari Mapping *Government Customer* (GC).

Contoh query pada gambar 3.16 merupakan query yang dilakukan pada Mapping GC untuk mengisi kolom NIPNAS GROUP dan STANDARD NAME NIPNAS GROUP.

```
drop table temp_8912672_1 purge;
commit;
create table temp_8912672_1 as
select distinct
  trim(a.periode)||substr(a.account_num,-4)||substr(a.account_num,1,4) as id
  , a.periode
  , substr(b.segment,-3) as segment
  , a.account_num
  , b.account_name
  , b.nipnas_group
  , b.standard_name
from
  mart_rev_pms_np_dgs_1501 a
  , t_mapping_account_gc_detil b
where
  a.periode = '202210'
  and a.nip_nas_group is null
  and a.source_data = 'TIBS-NP'
  and a.account_num = b.account_num
;
commit;
select * from temp_8912672_1 a; --8332 Rows
```

Gambar 3. 16 Mapping GC Non POTS


```

drop table temp_8912672_2 purge;
commit;
create table temp_8912672_2 as
select
  a.*
  , b.segmnet_fix
  , b.nipnas_group_fix
  , b.standard_name_fix
from
  temp_8912672_1 a
  , p_account_pms_dgs b
where
  a.account_num = b.account_num(+)
  and substr(b.periode_pms,1,4) = '2022'
  and b.periode_pms = (select max(z.periode_pms) from p_account_pms_dgs z where
  substr(z.periode_pms,1,4) = '2022' and z.account_num = b.account_num)
;
commit;
select * from temp_8912672_2 a; --8187 Rows
select a.id, count(*) from temp_8912672_2 a group by a.id having count(*) > 1;

```

Gambar 3. 17 Mapping GC Non POTS (2)

Apabila seluruh kolom sudah terisi sesuai kebutuhan maka tabel tersebut dipilih dan dimasukkan ke dalam kolom TIBS NP yang kosong dengan menggunakan query pada gambar 3.18.

```

select
  *
from
  temp_8912672_2 a
where
  a.segment <> a.segmnet_fix
;

select
  *
from
  temp_8912672_2 a
where
  a.standard_name <> a.standard_name_fix
;

insert into p_account_pms_dgs
select
  a.*
from
  temp_8912672_2 a
where
  not exists
  (
    select
      1
    from
      p_account_pms_dgs b
    where
      b.periode_pms = '202210'
  )
;
commit;
select * from p_account_pms_dgs a where a.periode_pms = '202210';

```

Gambar 3. 18 Insert data ke kolom TIBS NP

Setelah kolom tersebut terisi, maka dilanjutkan dengan pembuatan *table* baru untuk Mapping GC PMS Reguler Non POTS pada bulan tersebut dan melakukan *update* pada tabel bulan sebelumnya agar nilainya menjadi *terupdate* seperti gambar 3.19.

```
update
  mart_rev_pms_np_dgs_1501 a
set
  (
    a.segment_6_lname
    , a.nip_nas_group
    , a.standard_name_nipnas_group
    , a.witel_sold
    , a.treg_sold
  )
=
  (
    select
      z.segment
      , z.nipnas_group
      , z.standard_name
      , z.witel_sold
      , z.treg_sold
    from
      temp_8912672_4 z
    where
      z.account_num = a.account_num
  )
where
  a.periode = '202210'
and a.source_data = 'TIBS-NP'
and a.account_num in
  (
    select distinct
      y.account_num
    from
      temp_8912672_4 y
  )
;
commit;
```

Gambar 3. 19 Update tabel bulan sebelumnya

3. Setelah Mapping GC dilakukan, maka Mapping kedua yang dilakukan adalah Mapping Wilayah Telekomunikasi (WiTel) untuk mengisi kolom WITEL SHIP & WITEL SOLD. Contoh query untuk melakukan Mapping WiTel dapat dilihat pada gambar 3.20.

```

drop table temp_order_tos_pms_1 purge;
commit;
create table temp_order_tos_pms_1 as
select distinct
  a.cust_order_num
  , a.product_label
  , a.account_num
from
  mart_rev_pms_np_dgs_1501 a
where
  a.periode = '202210'
  and a.source_data = 'TIBS-NP'
  and substr(a.cust_order_num,1,3) = 'TOS'
;
commit;
select * from temp_order_tos_pms_1 a; --6806 Rows

```

Gambar 3. 20 Mapping WiTel

Dan jika sudah mendapatkan isi untuk kolom-kolom yang dibutuhkan sesuai dengan kebutuhan maka tahap terakhir pada Mapping WiTel adalah melakukan *update* pada tabel untuk perubahan data seperti pada gambar 3.21.

```

update
  mart_rev_pms_np_dgs_1501 a
set
  (
    a.witel_ship
    , a.treg_ship
  )
=
  (
    select distinct
      z.witel_name_sa
      , z.treg_sa
    from
      temp_order_tos_pms_6 z
    where
      z.cust_order_num || z.product_label = a.cust_order_num || a.product_label
  )
where
  a.periode = '202210'
  and a.source_data = 'TIBS-NP'
  and substr(a.cust_order_num,1,3) = 'TOS'
;
commit;

```

Gambar 3. 21 Update tabel WiTel

4. Tahapan selanjutnya pada PMS Non POTS yaitu melakukan penghitungan untuk mendapatkan selisih SAP yang dimana *source* data ini di dapat dari tim PPA menggunakan contoh query pada gambar-gambar dibawah ini

Query pada gambar 3.21 dan 3.22 menunjukkan pengambilan data selisih SAP untuk bulan baru dan pengambilan source data selisih SAP yang diberikan oleh tim PPA.

```
select
*
from
tmp_np_selisih_sap_rev a
where
a.periode = '202210'
;

delete
from
tmp_np_selisih_sap_rev a
where
a.periode = '202210'
;
commit;
```

Gambar 3. 22 Pengambilan data selisih SAP (1)

```
select
*
from
mart_rev_pms_np_dgs_1501 a
where
a.periode = '202210'
and a.source_data = 'SELISIH SAP'
;

delete
from
mart_rev_pms_np_dgs_1501 a
where
a.periode = '202210'
and a.source_data = 'SELISIH SAP'
;
commit;
```

Gambar 3. 23 Pengambilan data selisih SAP (2)

Hasil-hasil Mapping yang telah dilakukan sebelumnya memberikan hasil *revenue* dimana hasil *revenue* ini dimasukkan ke dalam *source* data selisih SAP yang baru

diberikan oleh tim PPA, untuk query *insert* data tersebut dapat dilihat pada gambar-gambar dibawah ini

```
insert into mart_rev_pms_np_dgs_1501
(
  periode
  , curr_type
  , customer_ref
  , cust_name
  , account_num
  , bill_prd
  , invoice_num
  , npwp
  , revenue_code_id
  , product_group
  , prod_period
  , gl_account
  , business_share
  , bill_mny
  , product_name
  , product_id
  , product_label
  , product_seq
  , owner_code
  , cid
  , circuit_name
  , circuit_speed
  , contract_no
  , msid
  , cust_order_num
  , ba_selling
```

Gambar 3. 24 *Insert data ke SAP (1)*

```
, ba_support
, ba_perc
, start_dat
, end_dat
, abonemen
, ovr_start_dat
, ovr_end_dat
, supplier_order_num
, parent_product_seq
, desc2
, local_amount
, nip_nas
, standard_name
, segment_6_lname
, nip_nas_group
, c_cluster
, divisi
, nik_am
, nama_am
, nik_kam
, nama_kam
, source_data
, group1
, group2
, group3
, group4
, uner
, standard_name_nipnas_group
, wil_id_sold
, witel_sold
, treg_sold
```

Gambar 3. 25 Insert data ke SAP (2)

```
, wil_id_bill  
, witel_bill  
, treg_bill  
, wil_id_ship  
, witel_ship  
, treg_ship  
, cust_order_num_plain  
, lcd  
)  
select  
  a.periode  
  , a.curr_type  
  , a.customer_ref  
  , a.cust_name  
  , a.account_num  
  , a.bill_prd  
  , a.invoice_num  
  , a.npwp  
  , a.revenue_code_id  
  , a.product_group  
  , a.prod_period  
  , lpad(a.gl_account, 10, '0') as gl_account  
  , a.business_share  
  , a.bill_mny  
  , a.product_name  
  , a.product_id  
  , a.product_label  
  , a.product_seq  
  , a.owner_code  
  , a.cid  
  , a.circuit name
```

Gambar 3. 26 Insert data ke SAP (3)

```
, a.circuit_speed  
, a.contract_no  
, a.msid  
, a.cust_order_num  
, a.ba_selling  
, a.ba_support  
, a.ba_perc  
, a.start_dat  
, a.end_dat  
, a.abonemen  
, a.ovr_start_dat  
, a.ovr_end_dat  
, a.supplier_order_num  
, a.parent_product_seq  
, a.desc2  
, a.local_amount  
, a.nip_nas  
, a.standard_name  
, trim(a.segment_6_lname) as segment_6_lname  
, a.nip_nas_group  
, a.c_cluster  
, a.divisi  
, a.nik_am  
, a.nama_am  
, a.nik_kam  
, a.nama_kam  
, a.source_data  
, b.group1  
, b.group2  
, b.group3
```

Gambar 3. 27 Insert data ke SAP (4)

```

, b.group4
, a.uner
, a.standard_name_nipnas_group
, a.wil_id_sold
, a.witel_sold
, a.treg_sold
, a.wil_id_bill
, a.witel_bill
, a.treg_bill
, a.wil_id_ship
, a.witel_ship
, a.treg_ship
, a.cust_order_num_plain
, c.lcd
from
tmp_np_selisih_sap_rev a
, pms_glacc b
, p_gl_account_to_lcd c
where
a.periode = '202210'
and lpad(a.gl_account, 10, '0') = b.gl_acc(+)
and lpad(a.gl_account, 10, '0') = c.gl_account(+)
and to_date('202210', 'yyyymm') between c.valid_start(+) and c.valid_end(+)
;
commit;

```

Gambar 3. 28 Insert data ke SAP (5)

5. Lalu tahap terakhir yaitu dibuatkan excel *table summary* 80_20 oleh tim DaMan yang nantinya diserahkan kembali ke tim PPA yang isinya berupa *revenue* untuk di analisa dan di validasi sampai akhirnya data tersebut di konfirmasi *closing* sampai *final* ke cloud drive DGS.

Dan untuk PMS Reguler yang kedua yaitu PMS POTS, kolom yang diisi pada TIBS NP sama seperti PMS Non POTS yaitu NIPNAS GROUP, STANDARD NAME NIPNAS GROUP, WITEL SHIP dan WITEL SOLD. Tahapan yang dilakukan kurang lebih sama antara PMS Non POTS dan PMS POTS hanya saja sumber data yang dibutuhkan berbeda, jika pada PMS Non POTS yang dibutuhkan adalah TIBS NP dari tim DIT dan selisih SAP dari tim PPA, maka pada PMS POTS data yang dibutuhkan adalah TIBS NP dan SISKKA dimana kedua data ini di dapat dari tim DIT. Untuk tahapan pada PMS POTS adalah sebagai berikut:

1. Tahapan pertama pada PMS POTS adalah melakukan Mapping GC. Seluruh tahapan Mapping dilakukan dengan *case* pengambilan data dari seluruh T-REG dari divisi DES dan DBS beserta dengan segment-segment yang ada di divisi tersebut menggunakan query seperti pada gambar-gambar dibawah ini.

```
drop table temp_ncli_pms purge;
commit;
create table temp_ncli_pms as
select distinct
  a.periode
  , a.ncli
  , a.segment_6_lname
from
  mart_rev_pms_pots_dgs_2022 a
where
  a.periode = '202210'
;
commit;
select * from temp_ncli_pms; --99376 Rows

delete
from
  p_ncli_pms_dgs a
where
  a.periode = '202210'
;
commit;
```

Gambar 3. 29 Mapping GC POTS (1)

```
drop table temp_ncli_pms_1 purge;
commit;
create table temp_ncli_pms_1 as
select distinct
  a.periode || a.ncli as id
  , a.periode
  , a.ncli
  , a.segment_6_lname
  , b.segment_fix
  , b.nipnas_group
  , b.standard_name
from
  temp_ncli_pms a
  , p_ncli_pms_dgs b
where
  a.ncli = b.ncli(+)
  and substr(b.periode,1,4) = '2022'
  and b.periode = (select max(z.periode) from p_ncli_pms_dgs z where substr(z.periode,1,4) = '2022'
  and z.ncli = a.ncli)
;
commit;
select * from temp_ncli_pms_1; --98357 Rows
```

Gambar 3. 30 Mapping GC POTS (2)


```

select
  a.ncli
  , count(*)
from
  temp_ncli_pms_1 a
group by
  a.ncli
having
  count(*) > 1
;

drop table temp_78yuihb1_1 purge;
commit;
create table temp_78yuihb1_1 as
select distinct
  a.nipnas_group
  , a.standard_name
from
  temp_ncli_pms_1 a
;
commit;

```

Gambar 3. 31 Mapping GC POTS (3)

```

update
  mart_rev_pms_pots_dgs_2022 a
set
  (
    a.segment_6_lname
    , a.nip_nas_group
    , a.standard_name_nipnas_group
  )
=
  (
    select
      b.segment_fix
      , b.nipnas_group
      , b.standard_name
    from
      p_ncli_pms_dgs b
    where
      b.periode = '202210'
      and b.ncli = a.ncli
  )
where
  a.periode = '202210'
  and a.ncli in
  (
    select distinct
      z.ncli
    from
      p_ncli_pms_dgs z
    where
      z.periode = '202210'
  )
;
commit;

```

Gambar 3. 32 Mapping GC POTS (4)

2. Setelah dilakukan Mapping GC, maka tahapan kedua adalah melakukan Mapping WiTel untuk memenuhi kolom WITEL SHIP dan WITEL SOLD dengan query sebagai berikut

```
select
  numbertools.thousand(sum(a.local_amount))
from
  mart_rev_pms_pots_dgs_2022 a
where
  a.periode = '202210'
  and a.standard_name_nipnas_group is null
;

select distinct
  numbertools.thousand(sum(a.local_amount))
-- a.ncli
-- , a.telp
from
  mart_rev_pms_pots_dgs_2022 a
where
  a.periode = '202210'
-- and a.standard_name_nipnas_group is null
  and a.witel is null
;
```

Gambar 3. 33 Mapping WiTel POTS

3. Kemudian tahap terakhir adalah melakukan *update* pada file PMS POTS yang nantinya akan di input ke dalam file SISKA.

```
update
  mart_rev_pms_pots_dgs_2022 a
set
  (
    a.group1
    , a.group2
    , a.group3
    , a.group4
    , a.desc2
  )
=
  (
    select
      z.group1
      , z.group2
      , z.group3
      , z.group4
      , z.desc2
    from
      pms_glacc z
    where
      z.gl_acc = a.gl_acc
  )
where
  a.periode = '202210'
;
commit;
```

Gambar 3. 34 Update data ke SISKA

C.2 Exception List POTS

Tujuan dari Exception List POTS kurang lebih sama seperti Whitelist yaitu untuk mencegah nomor *customer* terisolir tapi untuk Exception List hanya untuk data-data POTS saja berdasarkan nomor telepon/NCLI seperti nomor pejabat yang tidak boleh mati dalam 1 bulan. Exception juga berguna untuk mengecek apakah dalam nama *customer* yang ada di DGS, *revenue* dari *customer* tersebut memang masuk ke DGS atau tidak. Contoh data *customer* POLRI ada di DGS, pembayaran yang dilakukan juga diberikan ke DGS akan tetapi *revenue* yang di dapat bukan ke DGS melainkan ke divisi lain, Exception List dapat mencegah terjadi hal seperti ini dan mengecek apabila terjadi hal seperti ini. Exception dilakukan dengan cara memenuhi kolom *corporate name* dari data yang diberikan oleh tim *Cash Collection & Debt Management* (CDM) dan melakukan eksekusi di SQL Developer. Tahapan dalam Exception List POTS dapat dilihat pada gambar 3. 35 dan 3. 36 dengan keterangan sebagai berikut:

1. Pertama kali tim DaMan akan mendapatkan *source* data berdasarkan nomor telepon yang telah di *Mapping* oleh segment ke *Government Customer* nya.
2. Kemudian dilakukan eksekusi *database* dengan membuat *temporary table* untuk menampung hasil *Mapping* nomor telepon ke *Government Customer* nya dari segment.
3. Setelah dibuatkan *temporary table* maka dilakukan pencarian NCLI terhadap nomor-nomor telepon yang diusulkan untuk di Exception List dengan cara pengecekan ke *table* yang telah diberikan oleh segment di mybrain untuk mendapatkan NCLI dari nomor telepon dan didapatkan table

A, lalu dilakukan pengecekan duplikasi data dari nomor telepon yang ada di table A apakah ada data *duplicate* dari data yang di *Mapping* NCLI oleh segment.

4. Kemudian yang terakhir melakukan *insert* ke *table* utama di SAP.

Untuk contoh dari query tahapan-tahapan diatas dapat dilihat pada gambar dibawah ini

The image shows a series of SQL queries with red boxes highlighting specific parts and blue lines pointing to descriptive text:

- Query 1:** `create table TEMPF_19OKTO22_001_MPS (NOTEL varchar2(20), NIPNAS_GROUP varchar2(20), STANDARD_NAME varchar2(100));` - Annotation: *table temporary untuk pengecekan*
- Query 2:** `SELECT a FROM TEMPF_19OKTO22_001_MPS a;` - Annotation: *select table temporary*
- Query 3:** `create table TEMPF_19OKTO22_001_MPS_CEK as select distinct '19-10-2022' LAST_UPD, 'DGS' DIVISI, 'MPS' SEGMENT, b.NCLI, a.NIPNAS_GROUP, a.STANDARD_NAME from TEMPF_19OKTO22_001_MPS a, ODS_ISISHA.DOSLIG@DBL_AON_ODSNASDB_USER1 b, MYBRAINS.ttt_cuat@db1_billcoa b where a.NOTEL = b.SND(+);` - Annotation: *pembuatan table dari table temporary untuk diambil beberapa kolom yang utama*
- Query 4:** `select * from TEMPF_19OKTO22_001_MPS_CEK a;` - Annotation: *select table yang sudah dibuat sebelumnya*
- Query 5:** `select * from temp_map_ncli_to_nipnas_dgs a where a.NCLI in (select b.NCLI from TEMPF_19OKTO22_001_MPS_CEK b);` - Annotation: *pengecekan apakah NCLI dari data tersebut sudah pernah ada sebelumnya (duplikasi data)*

Gambar 3.35 Exception List (1)

The image shows a series of SQL queries with red boxes highlighting specific parts and blue lines pointing to descriptive text:

- Query 1:** `delete from temp_map_ncli_to_nipnas_dgs a where a.NCLI in (select b.NCLI from TEMPF_19OKTO22_001_MPS_CEK b);` - Annotation: *menghapus duplikasi data dari NCLI sebelumnya*
- Query 2:** `insert into temp_map_ncli_to_nipnas_dgs select * from TEMPF_19OKTO22_001_MPS_CEK a;` - Annotation: *insert kembali NCLI yang sudah dihapus jadi tidak akan ada duplikasi data*
- Query 3:** `select a.NCLI, count(*) from temp_map_ncli_to_nipnas_dgs a group by a.NCLI having count(*)>1;` - Annotation: *pengecekan apakah dalam 1 Government Customer memiliki banyak NCLI*
- Query 4:** `delete map_ncli_to_NIPNAS_ALL x where exists (select 1 from temp_map_ncli_to_nipnas_dgs where ncli = x.ncli);` - Annotation: *proses delete NCLI tanpa mapping ke SAP*
- Query 5:** `insert into map_ncli_to_NIPNAS_ALL select * from temp_map_ncli_to_nipnas_dgs;` - Annotation: *proses insert NCLI ke SAP*

Gambar 3.36 Exception List (2)

5. Setelah melakukan *insert table* dari SQL Developer dimana data ini terhubung dengan SAP yang dapat diakses oleh tim CDM dan jika keseluruhan data sudah terisi dimana nomor-nomor yang ada diberikan penjelasan terisolir/tidak maka Exception List POTS sudah selesai dilakukan.

D. Harian

D.1 Pelurusan data di aplikasi NCX, TIBS dan TREMS

Pelurusan data ini bertujuan untuk meluruskan data yang bermasalah dengan melakukan *request* antrian sesuai dengan permasalahan yang ada di data, sehingga data yang bermasalah dapat lurus dan pada akhirnya akan menjadi *close order*. Pelurusan data ini dilakukan dengan cara pengecekan dan konfirmasi terhadap data-data yang terkait dan dikirimkan ke pembuat antrian berdasarkan NoDin, menindaklanjuti hasil pengecekan untuk *update* data di masing-masing aplikasi dan konfirmasi serta eksekusi oleh bagian *service desk* yaitu DIT aplikasi NCX, TIBS dan TREMS.

Penyebab terjadinya pelurusan data bisa dikarenakan beberapa hal, yaitu:

1. NIPNAS yang tidak sesuai

Untuk penyebab yang pertama yaitu NIPNAS tidak sesuai, karena ketidaksesuaian ini maka data yang ada menjadi tidak lurus atau biasa disebut sebagai ketidakcocokan antara satu data dengan data yang lain sehingga menimbulkan kegagalan pada data tersebut. Tahapan yang pertama apabila terjadi ketidaklurusan data adalah meminta No Service ID (SID) pada data yang tidak sesuai kepada pembuat *request*. Sebagai contoh pelurusan data dengan penyebab NIPNAS tidak sesuai dapat dilihat pada contoh dibawah ini.

Pada contoh kasus di gambar 3.37, dapat dilihat bahwa NIPNAS yaitu NIPNAS 3333 yang terdapat pada kolom CA_SITE dan NIPNAS_TIBS dimiliki oleh 2 *customer* atau 2 GC yang berbeda yaitu customer Kementerian Keuangan dan customer POLRI, sedangkan setiap NIPNAS tidak boleh dimiliki oleh 2 *customer* yang berbeda. Tujuan dari pelurusan ini adalah membuktikan bahwa NIPNAS 3333 sebenarnya dimiliki oleh customer Kementerian Keuangan atau POLRI. Setelah diketahui kepemilikan NIPNAS sebenarnya, maka dilakukan perbaikan pada data yang tidak sesuai tersebut di SQL Developer.

CA_SITE	CA_NAME	NIPNAS_TIBS	NIPNAS_NAME
3333	KEMENTRIAN KEUANGAN	3333	POLRI

Gambar 3. 37 Contoh kasus NIPNAS tidak sesuai

Tahapan selanjutnya adalah tim DaMan melakukan pengecekan yaitu dengan menanyakan kepada tim Regional bahwa *customer* manakah yang sebetulnya memiliki NIPNAS 3333 dan kemudian dilakukan pengecekan juga yang dilakukan sendiri oleh tim DaMan pada *database* yang dimiliki oleh tim DaMan menggunakan query pada gambar 3.38.

```

SELECT
  a.segment,
  a.li_sid,
  a.order_created_date,
  a.order_id,
  a.order_subtype,
  a.order_status,
  a.li_status,
  a.li_milestone,
  a.ba_site,
  a.ca_site_parent_ba,
  a.ca_name_parent_ba,
  a.accountnas,
  a.nipnas_trems as nipnas_tibs,
  b.standard_name as nipnas_name_tibs,
  a.sa_site,
  a.quote_num
FROM zzz_ncx_ebis a,
     customer_master_dgs_2022 b
where a.li_sid = '
and a.nipnas_trems = b.nipnas_ncx(+)
order by a.order_created_date;

```

Gambar 3. 38 Pengecekan di SQL

Setelah dilakukan pencocokan antara tim Regional dan tim DaMan, maka tim DaMan meminta tim Regional untuk dibuatkan NoDin terkait perubahan data tersebut karena tim DaMan tidak dapat merubah secara langsung tanpa adanya NoDin yang dibuatkan. Apabila NoDin sudah dibuatkan dan diturunkan kepada tim DaMan, maka perubahan dapat dilakukan pada aplikasi NCX yang langsung terhubung dengan *database* pada SQL Developer.

2. Berdasarkan hasil rekon jadi hanya dilakukan 1 tahun sekali.

Penyebab kedua sehingga dilakukan pelurusan data dari beberapa aplikasi dikarenakan adanya perubahan dari hasil rekon yang terjadi hanya satu tahun secalin inti dari perubahan ini yaitu hanya melakukan perubahan dari NCLI lama ke NCLI baru dari satu Nomor Telepon (NoTel) *customer* yang ada di DGS.

Contoh pada kasus ini seperti gambar 3.39 yaitu PEMKOT Bandung dengan NoTel A belum memiliki NCLI di daerah nya untuk dipasangkan layanan dari perusahaan dimana daerah yang ingin dipasangkan layanan perusahaan hanya dapat dimasukkan ke daerah PEMKAB Bandung. Oleh karena itu NCLI dari NoTel A yang sebelumnya masuk ke NCLI PEMKAB Bandung dibuatkan NCLI baru atau NCLI yang tersedia di DGS khusus untuk PEMKOT Bandung agar tidak tercampur dengan NCLI PEMKAB Bandung. Perubahan pada NCLI ini dilakukan satu tahun sekali setiap pembuatan dan keputusan hasil rekonsiliasi. Untuk query dari penyebab perubahan hasil rekon memang tidak pernah disimpan karena pasti selalu custom setiap tahunnya disesuaikan dengan kebutuhan dan keinginan dari *customer* dan perusahaan.

NOTEL	NCLI_LAMA	NCLI_BARU
12345678	3333	4444

Gambar 3. 39 Contoh kasus berdasarkan hasil rekon

3. Permintaan dari pelanggan yang meminta untuk diubah data *customer* nya.

Lalu penyebab terakhir dari pelurusan data yaitu customer sendiri yang meminta untuk dilakukan *update* atau perubahan pada data nya yang ada di perusahaan. Sebagai contoh untuk penyebab ini dapat dilihat pada gambar 3.40. Pada gambar tersebut dapat dilihat bahwa pada BA SITE atau SID 2131 yang dimiliki oleh *customer account* PEMKOT Makassar dari Sulawesi Selatan ingin dipindahkan ke *customer account* baru yaitu PEMKAB Maros. Pelurusan ini juga bertujuan untuk meningkatkan performansi dari *Account Manager* yang menangani masalah ini, misalkan masalah ini ditangani oleh *Account Manager* A ketika customer account nya masih di PEMKOT Makassar, akan tetapi ketika ingin dilakukan perubahan ditangani oleh Account Manager B maka salah satu akibat jika tidak diluruskan perubahan data ini akan berdampak ke performansi *Account Manager* (AM) yang bertugas sebagai pencari *customer* atau *project*.

BA_SITE	CA_LAMA	CA_BARU
2131	PEMKOT MAKASSAR	PEMKAB MAROS

Gambar 3. 40 Contoh kasus karena permintaan pelanggan

Untuk tahapan yang dilakukan pada pelurusan data yang dikarenakan permintaan dari pelanggan sendiri adalah sebagai berikut:

1. Dilakukan perubahan dengan melakukan query pada SQL Developer oleh tim DaMan. Sebagai contoh dari kasus ini maka query nya seperti pada gambar 3.41.


```

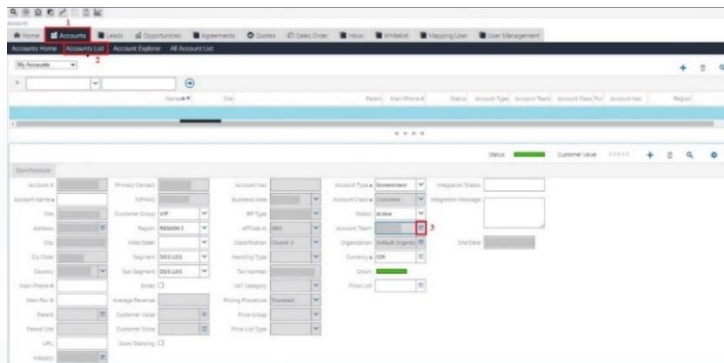
SELECT
  a.segment,
  a.li_sid,
  a.order_created_date,
  a.order_id,
  a.order_subtype,
  a.order_status,
  a.li_status,
  a.li_milestone,
  a.ba_site,
  a.ca_site_parent_ba,
  a.ca_name_parent_ba,
  a.accountnas,
  a.nipnas_trems as nipnas_tibs,
  b.standard_name as nipnas_name_tibs,
  a.sa_site,
  a.quote_num
FROM zzz_ncx_ebis a,
     customer_master_dgs_2022 b
where a.li_sid = '
and a.nipnas_trems = b.nipnas_ncx(+)
order by a.order_created_date;

SELECT
  *
FROM customer_master_dgs_2022 a
where a.standard_name like '%MAROS%';

```

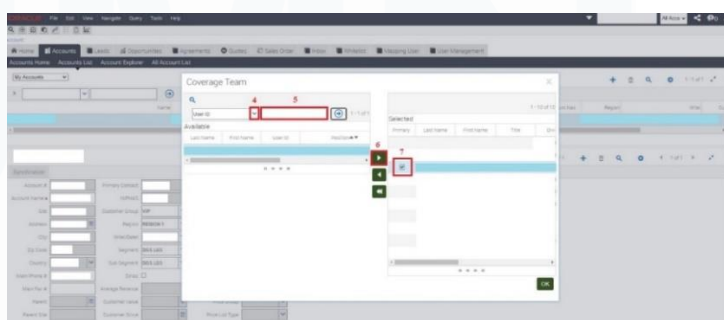
Gambar 3. 41 Perubahan data pada database

2. Lalu jika sudah dilakukan perubahan dalam SQL Developer untuk database perusahaan maka tahapan selanjutnya yang dilakukan adalah melakukan perubahan pada aplikasi NCX dimana tujuannya untuk merubah *Account Manager* yang bertanggung jawab dalam *customer* atau *project* ini. Ketika membuka aplikasi NCX hal yang pertama kali dilakukan adalah memilih fitur *Account – Account List* dan mengisi SID dari *customer* yang ingin dilakukan perubahan. Setelah itu akan muncul detail dari seluruh data *customer account* tersebut, kemudian tim DaMan memilih bagian *Account Team* seperti pada gambar 3.42.



Gambar 3. 42 Perubahan pada aplikasi NCX

Setelah memilih bagian *Account Team*, maka akan muncul *popup* untuk perubahan nama dengan mengisi kolom User ID untuk mengisi Nomor Induk Karyawan (NIK) dari *Account Manager* yang baru dan NIK atau nama dari *Account Manager* tersebut dijadikan *primary* dimana ini menandakan bahwa *Account Manager* tersebut merupakan orang yang bertanggung jawab dalam *customer* atau *project* ini sehingga performansi dari *Account Manager* ini juga akan berpengaruh untuk kedepannya nanti.



Gambar 3. 43 Perubahan pada aplikasi NCX (2)

D.2 Pemenuhan request permintaan data

Pemenuhan *request* permintaan data ini berisi tentang pemenuhan data yang ada di aplikasi NCX, TIBS, TREMS, PMS atau *Mapping Account*. Pemenuhan *request* permintaan data ini biasanya berkaitan sesuai dengan kebutuhan dari *user* unit, jadi sudah pasti pemenuhan *request* ini sifatnya berbeda-beda dan tidak menentu tergantung dengan kebutuhan dari data atau *user*. Beberapa contoh dari *request* permintaan data seperti pemenuhan data performansi yaitu seperti ingin melihat *sales*, melakukan *Mapping* pelanggan, melihat atau melakukan pengecekan pada data *sales bandwith*, data *sales* internet dan telepon. Dibawah ini merupakan contoh query dari pengambilan *list* populasi layanan POTS untuk segment CGS dan DGS.

```
drop table TMP_HCNJU_78_1 purge;
commit;
create table TMP_HCNJU_78_1 as
SELECT DISTINCT
  a.~,
  b.nipnas_group,
  b.standard_name
FROM TMP_HCNJU_78_89 a,
  p_ncli_pms_dgs b
where a.ncli = b.ncli(+)
and b.periode = (select max(z.periode) from p_ncli_pms_dgs z where z.ncli = b.ncli)
order by a.ncli;
commit;

SELECT
  ~
  ~
FROM TMP_HCNJU_78_1 a;

drop table TMP_HCNJU_78_2 purge;
commit;
create table TMP_HCNJU_78_2 as
SELECT DISTINCT
  a.~,
  b.witel_name as witel_ship,
  b.treg as treg_ship,
  c.witel_sold,
  c.treg_sold
FROM TMP_HCNJU_78_1 a,
  zzz_datel_witel_2022 b,
  customer_master_dgs_2022 c
where a.centite = b.centite(+)
and a.nipnas_group = c.nipnas_group(+)
order by a.ncli;
commit;
--122859 Rows

SELECT
  ~
  ~
FROM TMP_HCNJU_78_2 a;
```

Gambar 3. 44 Contoh kasus pemenuhan request data (1)

```

drop table tmp_hcnju_78 purge;
commit;
create table tmp_hcnju_78 as
SELECT DISTINCT
case
  when (substr(b.cgest,1,2) = 'G1') then 'DGS CGS'
end segment,
b.cgest,
a.ncli
FROM MYBRAINS.ttt_cust@db1_billcos a,
v_ncli_lock b
where a.ncli = b.ncli(+)
and substr(b.cgest,1,2) = 'G1';
commit;
--and a.datrs is null;

SELECT
*
FROM TMP_HCNJU_78 a;
--35185 Rows

drop table TMP_HCNJU_78_89 purge;
commit;
create table TMP_HCNJU_78_89 as
SELECT DISTINCT
a.*,
b.nama_ncli,
b.snd as notel,
b.nama as nama_notel,
b.datms as tgl_aktif,
b.datrs as tgl_cabut,
b.centive,
b.ALAMAT_INST,
b.BUNDLING
FROM TMP_HCNJU_78 a,
MYBRAINS.ttt_cust@db1_billcos b
where a.ncli = b.ncli(+)
commit;

SELECT
*
FROM TMP_HCNJU_78_89 a;

```

Gambar 3. 45 Contoh kasus pemenuhan request data (2)

D.3 Permintaan *login* aplikasi

Tujuan dari permintaan *login* aplikasi yang dilakukan oleh tim DaMan adalah untuk memeriksa dan melaporkan hasil dari pengecekan NoDin sampai dengan kelengkapan data dan informasi terkait yang akan dikirimkan berhubungan dari pengirim NoDin untuk *user* yang ingin melakukan *login* aplikasi. Untuk tahapan yang dilakukan untuk melakukan pemenuhan permintaan login aplikasi adalah

1. Tahapan pertama adalah tim DaMan mengunduh seluruh lampiran yang diberikan dalam NoDin untuk tim DaMan.

Lampiran ini berisi file *Form*, Pakta Integritas yaitu perjanjian yang merupakan persetujuan yang dibuat oleh dua pihak atau lebih tentang kesepakatan akan mutu, potensi, dan/atau kemampuan dalam menjaga wibawa atau kejujuran [5], serta foto kartu pegawai yang telah diisi untuk *user* yang organik. Kemudian untuk lampiran *user* Non Organik adalah file *Form*, Pakta Integritas dan foto KTP. Untuk contoh *form user* organik dapat dilihat pada gambar 3. 46.

A	B	C	D	E	F
BP_ID	Mr/Ms:	First Name	Last Name	Division	Position Type/Role
12345	Ms	KEZIA	VICTORIA	DGS	Data Management

Gambar 3. 46 *Form user* organik

Dan untuk contoh dari file *form user* non organik seperti gambar 3. 47.

A	B	C	D	E	F	G	H	I
Login Tindakan (jika ada)	Tgl lahir (DDMMYYYY)	Mr/Ms	First Name	Last Name	E-mail ID	Position Type/Role	Divisi & Segmen	HR Access
12345	31/12/1999	MS	KEZIA	VICTORIA	kezia.victoria@ptelabs.com	Internal Data Management	DGS PPS	HRMS

Gambar 3. 47 *Form user* Non organik

2. Kemudian tahapan kedua yang dilakukan oleh tim DaMan adalah melakukan *follow up* kepada tim DIT atau tim yang bersangkutan sampai diberikan konfirmasi E-mail jawaban terkait hasil eksekusi mengenai pengecekan kelengkapan data yang dilakukan oleh tim DaMan yang dilakukan oleh pihak *service desk* DIT atau tim yang bersangkutan.
3. Setelah seluruh dokumen yang dibutuhkan sudah selesai diisi dan dicek kelengkapan datanya oleh tim DaMan, maka dokumen tersebut dapat diberikan ke tim DIT atau tim yang bersangkutan dalam bentuk NoDin.

D.4 Pembuatan *Customer Account* (CA)

Pembuatan CA hanya digunakan khusus untuk layanan Non POTS seperti ASTInet, VPN, dan IP Transit, nantinya CA ini akan terhubung dari No CA dan No NIPNAS (Nomor Induk Penduduk Nasional). Sedangkan untuk layanan POTS tidak membutuhkan pembuatan CA karena sudah memiliki NCLI dimana NCLI disini bersifat sejajar dengan No CA. Tujuan dari pembuatan CA ini yaitu dilakukan pembuatan CA bagi *customer* yang akan memiliki layanan telekomunikasi di PT. Telekomunikasi Indonesia. Sebagai contoh adalah Kementerian A meminta perusahaan untuk dibuatkan layanan ASTINet sebagai layanan *internal* yang akan digunakan di 20 lantai dengan kecepatan *unlimited*, maka *customer* ini baru bisa dibuatkan *Customer Account* nya. Tahapan dalam pembuatan CA dapat dijelaskan dan digambarkan sebagai berikut:

1. Tim DaMan akan mengirimkan file Form Permintaan CA CA kepada unit atau segment yang melakukan *request* pembuatan CA. Form ini berisi tentang data-data yang dibutuhkan oleh DaMan nantinya dalam pembuatan CA. Untuk contoh Form Permintaan CA dapat dilihat pada gambar 3. 48.

<i>Account Detail</i>	
Nama Customer Account	
Address	
City	
Zip Code	
Country	
Main Phone (+62)	
Primary Contact	
Region	
Vitel/Date1	
Segment	
Sub Segment	
Account Type	
Account Team (NIKAM & Finance TREG, Sales, Sales Engineer)	
NPI/VP	
<i>Account Address</i>	
House Number	
Floor	
Building Name	
Street Address	
Province	
District	
City	
Zip Code	
Country	
Latitude	
Longitude	
<i>Contact Person CC</i>	
First Name	
Last Name	
Mobile	
Job Title	
Work Phone (+62)	
Work Fax (+62)	
Email	
Mobile Phone (+62)	

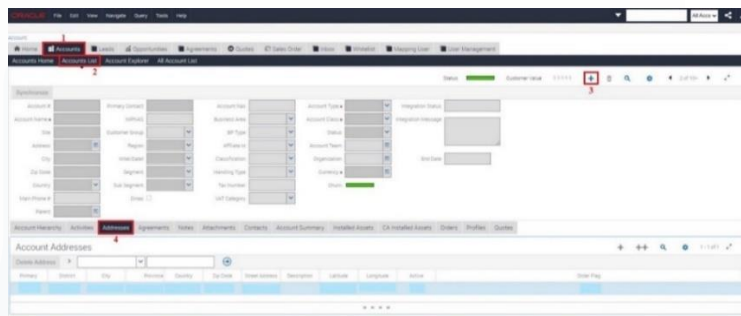
Gambar 3. 48 Form Permintaan CA

2. Kemudian setelah unit atau segment yang melakukan *request* telah mengirimkan kembali *form* yang sudah diisi sesuai dengan kebutuhan, maka tim DaMan akan melakukan pengecekan di *table master* CBase apakah *Government* atau *Customer* tersebut sudah memiliki *CA_site* atau *CA_no*. Pengecekan ini dilakukan oleh tim DaMan di SQL Developer dengan query seperti gambar 3. 49.

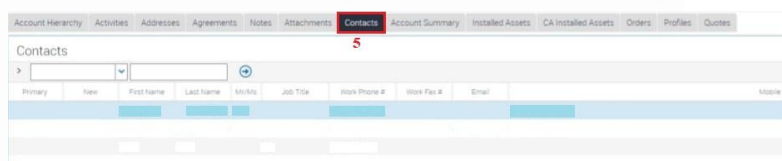


Gambar 3. 49 Pengecekan pada *table master*

3. Untuk tahapan terakhir yaitu jika memang betul *customer* tersebut belum ada *CA* nya dalam *database* maka *Government* atau *Customer* tersebut bisa dibuatkan *CA* nya. Pembuatan *CA* dilakukan pada aplikasi NCX dengan mengisi kolom *Header*, *Address* dan *Contact* sesuai dengan isi yang ada di file Form Permintaan *CA*. Pengisian pada aplikasi NCX dapat dilihat pada gambar 3. 50 dan 3. 51.



Gambar 3. 50 Pembuatan *CA* di aplikasi NCX (1)

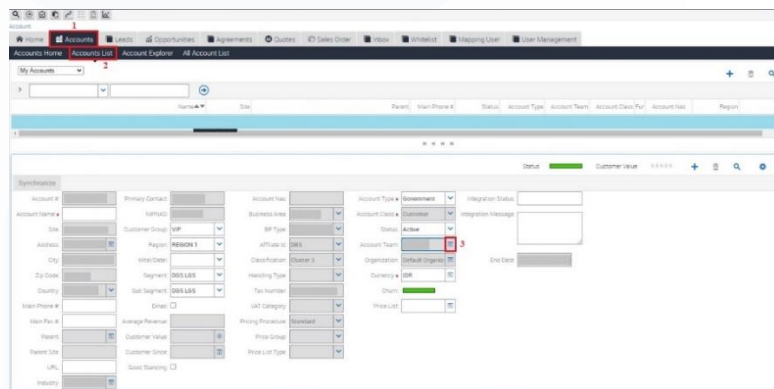


Gambar 3. 51 Pembuatan *CA* di aplikasi NCX (2)

D.5 Assign Customer Account

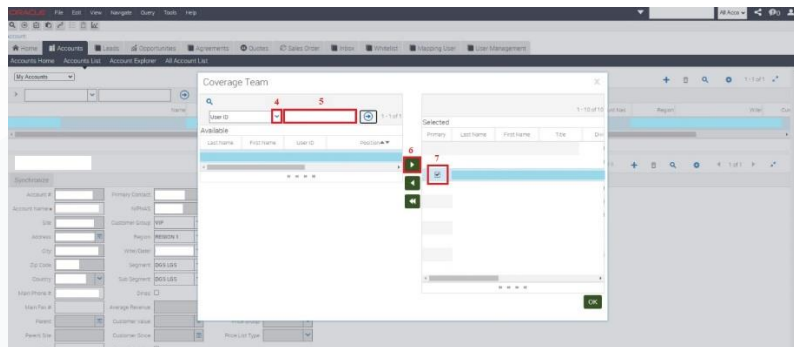
Pada *assign Customer Account*, tim DaMan akan menerima *request* dari Whatsapp/Telegram/E-mail terkait dengan *assign CA* sesuai dengan kebutuhan yang dibutuhkan. Tujuan dari *Assign CA* adalah agar AM tersebut dapat melihat *user* yang ingin dicari atau dibutuhkan. Misalkan dalam tim A yang beranggotakan 3 orang terdiri dari 1 AM dan 2 orang *Officer*, ketika AM tersebut membutuhkan *detail* data dari *Officer A* atau *customer A* tentu AM tersebut dapat melihat setelah dilakukan *assign CA* yang telah diberikan oleh tim DaMan, sehingga AM tersebut tidak perlu meminta dari unit atau segment lain. Tahapan dalam *assign CA* adalah sebagai berikut:

1. Pada tahap pertama di aplikasi NCX, tim DaMan memilih *accounts list* dan memilih *icon drop down* pada bagian *account team* untuk melakukan *assign CA*.



Gambar 3. 52 Assign customer account (1)

2. Kemudian tahapan selanjutnya adalah tim DaMan memilih *user ID* dan mengisi kolom ID sesuai dengan ID dari AM yang bersangkutan menjadikan ID tersebut sebagai *primary*.

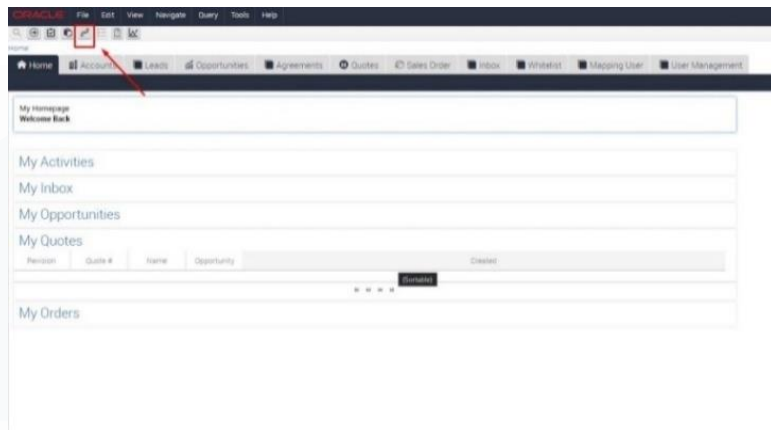


Gambar 3. 53 Assign customer account (2)

D.6 Create alamat di aplikasi NCX

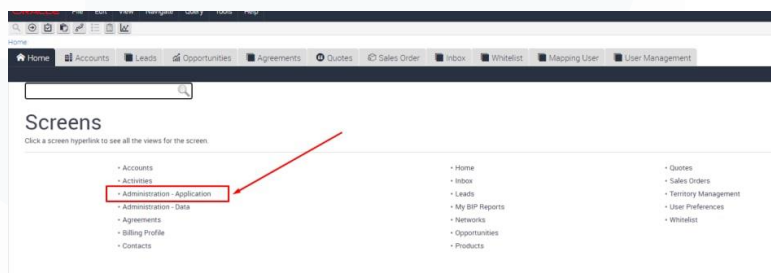
Pembuatan alamat pada aplikasi NCX ini dilakukan berdasarkan sumber dari *request create* alamat yang bertujuan untuk membuat atau menambahkan alamat baru di aplikasi NCX. Pembuatan alamat di aplikasi NCX biasanya dilakukan jika ada *customer* yang ingin melakukan pemasangan alat atau penggunaan layanan perusahaan di suatu tempat namun tempat tersebut belum terdeteksi, maka dilakukan pembuatan alamat secara manual oleh tim DaMan atau tim DIT. Pembuatan alamat secara manual ini biasanya dikarenakan alamat tersebut memang belum terdeteksi kemungkinan karena berada di pelosok-pelosok daerah. Untuk tahapan dalam pembuatan alamat di aplikasi NCX dilakukan sebagai berikut:

1. Pertama kali tim DaMan memilih fitur *create* alamat pada aplikasi NCX seperti gambar 3. 54.



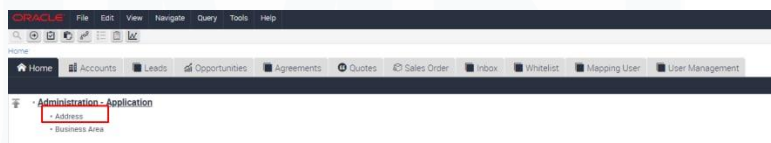
Gambar 3. 54 Pembuatan alamat (1)

2. Kemudian selanjutnya akan muncul halaman baru dan memilih bagian *Administration – Application*.



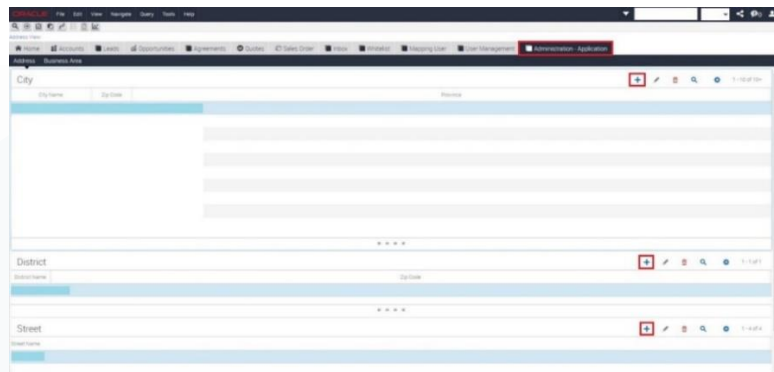
Gambar 3. 55 Pembuatan alamat (2)

3. Lalu memilih bagian *Address*.



Gambar 3. 56 Pembuatan alamat (3)

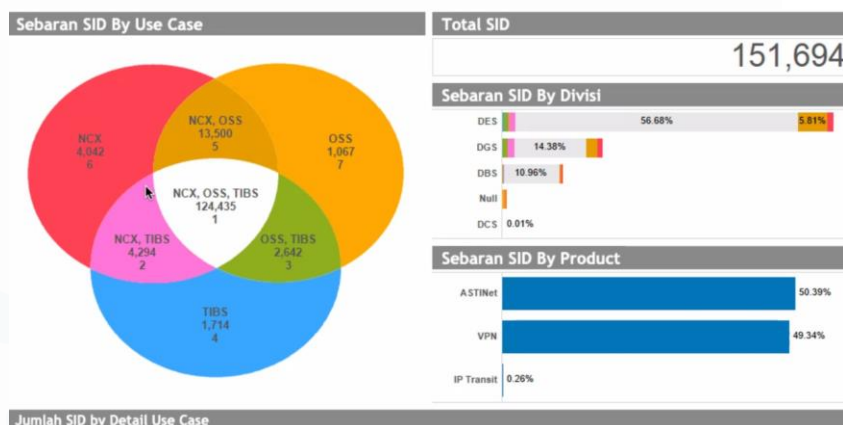
4. Dan untuk tahapan yang terakhir adalah melakukan *input* seluruh detail data di bagian *City*, *District*, dan *Street* sesuai dengan isi dari *request* dari alamat pemasangan untuk layanan perusahaan.



Gambar 3. 57 Pembuatan alamat (3)

D.7 Data Alignment

Tujuan dari *Data Alignment* adalah pengecekan data dan *support* data yang dibutuhkan untuk di *profiling* agar mudah di *follow up* dan selanjutnya data tersebut akan disampaikan ke unit RevAs untuk mengetahui apakah ada *revenue* dari data yang telah di *profiling* dari tim DaMan. *Data Alignment* ini pada dasarnya memiliki tujuh (7) *use case* yang kemungkinan dapat terjadi, dapat digambarkan seperti gambar 3. 58.



Gambar 3. 58 Sebaran 7 use case

Sebagai penjelasan dari gambar 3. 58 NCX merupakan aplikasi untuk melakukan *input* data *customer*, kemudian OSS adalah aplikasi atau kabel atau alat yang akan dipasang di tempat


```

SELECT DISTINCT
  a.li_PRODUCT_NAME,
  a.SID_UC2,
  a.SID_UC3,
  a.ORGAN,
  a.ORGANBY,
  b.ca_bill_payment_no,
  b.ca_bill_payment_no,
  a.PREV_ORDER,
  b.order_subtype as prev_order_subtype,
  b.order_status as prev_order_status,
  b.li_status as prev_li_status,
  b.li_billcode as prev_li_billcode,
  a.LAST_ORDER,
  a.LAST_ORDER_SUBTYPE,
  a.ORDER_CREATE_DATE,
  a.LAST_ORDER_STATUS,
  a.LAST_LI_STATUS,
  a.LAST_LI_BILLCODE,
  a.IS_SID_ON_TIME,
  a.IS_ORDER_ON_TIME,
  a.WITH_HOLD,
  a.TIME_HOLD
FROM TMS_SID_UC2_00422_L_0
  a,a
where a.prev_order(a.sid_uc2 = b.order_id+)|b.li_billcode|
--

```

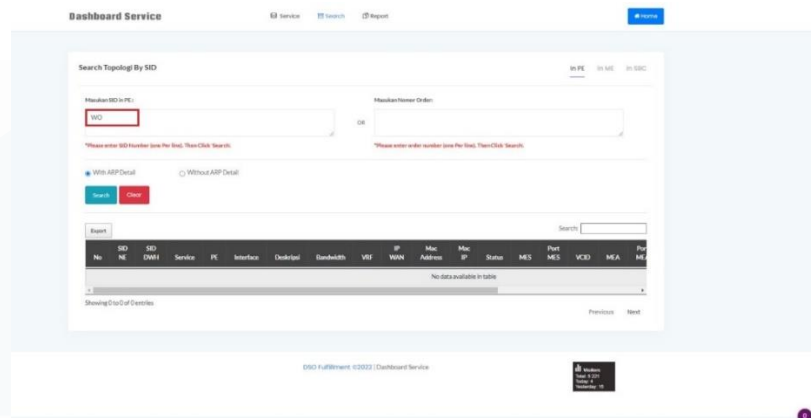
menampilkan previous order

Gambar 3. 60 Menampilkan *previous order*

Pada gambar 3.59 dan 3.60, dibuatkan query untuk mengecek data dari aplikasi NCX dan TIBS. Untuk pengecekan di aplikasi NCX dan TIBS menggunakan SQL Developer dan dibutuhkan *previous order* untuk mengeluarkan data-data yang *abandoned* agar memudahkan tim DCS untuk fokus melihat data yang akan digunakan atau dibutuhkan.

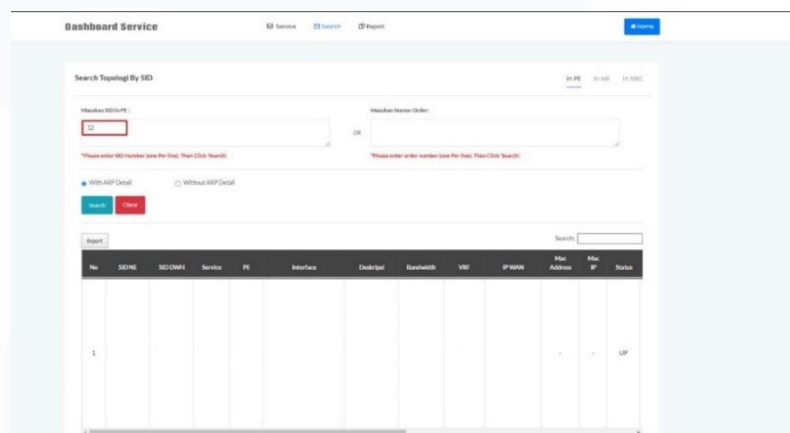
Tahapan selanjutnya setelah di dapatkan hasil berupa kolom-kolom yang dibutuhkan ketika melakukan di SQL Developer untuk pengecekan aplikasi NCX dan TIBS, kolom-kolom tersebut kemudian di *export* dan dijadikan excel baru.

2. Ketika pengecekan dua (2) aplikasi sudah dilakukan, maka dilakukanlah pengecekan di aplikasi terakhir, yaitu aplikasi OSS. Untuk pengecekan aplikasi OSS tidak menggunakan SQL Developer, melainkan menggunakan *website internal* perusahaan khusus untuk aplikasi OSS yaitu SIMS.Telekomunikasi.co.id. Pada pengecekan di *website* tersebut, yang perlu dicari adalah data berdasarkan No SID, maka tim DaMan akan memasukan No SID dari *customer* tersebut, apabila data tersebut memang betul tidak ada di aplikasi OSS, maka hasilnya akan seperti pada gambar 3. 61.



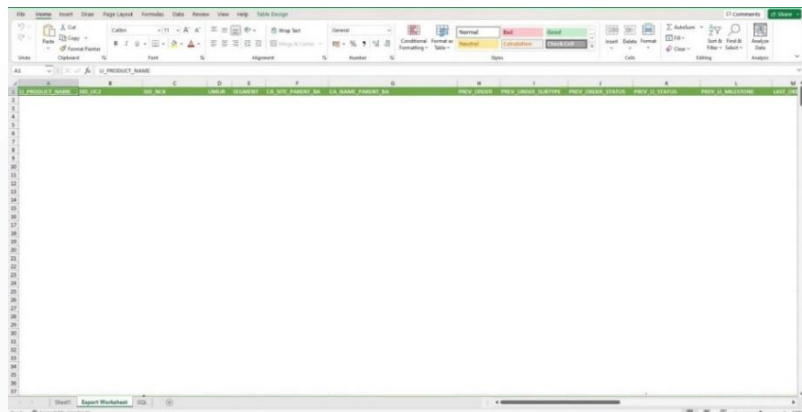
Gambar 3. 61 Tampilan data tidak ada di aplikasi OSS

Namun apabila ternyata data tersebut ada di dalam aplikasi OSS maka hasilnya akan seperti pada gambar 3. 62.

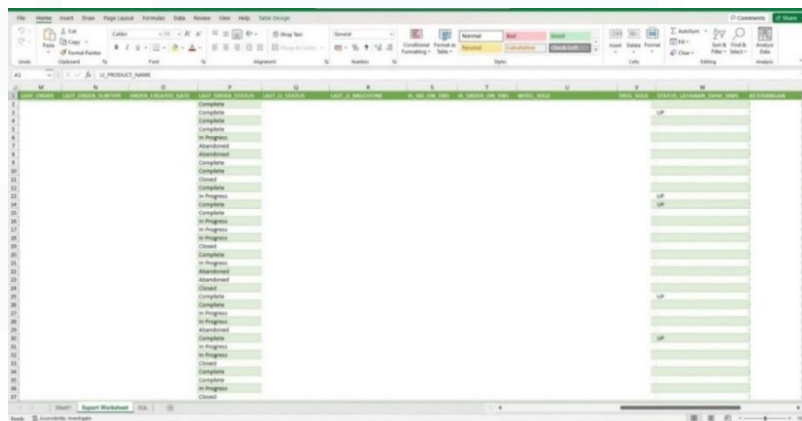


Gambar 3. 62 Tampilan data ada di aplikasi OSS

3. Kemudian data yang berada di *website* tersebut di *export* dan akan menjadi excel baru. Setelah mendapatkan 2 excel baru selanjutnya tim DaMan akan melakukan VLOOKUP terhadap kedua data tersebut berdasarkan No SID.
4. Setelah melakukan VLOOKUP maka di dapatkan hasil untuk *use case 2* seperti gambar berikut



Gambar 3. 63 Hasil VLOOKUP dari hasil SQL dam aplikasi OSS (1)



Gambar 3. 64 Hasil VLOOKUP dari hasil SQL dam aplikasi OSS (2)

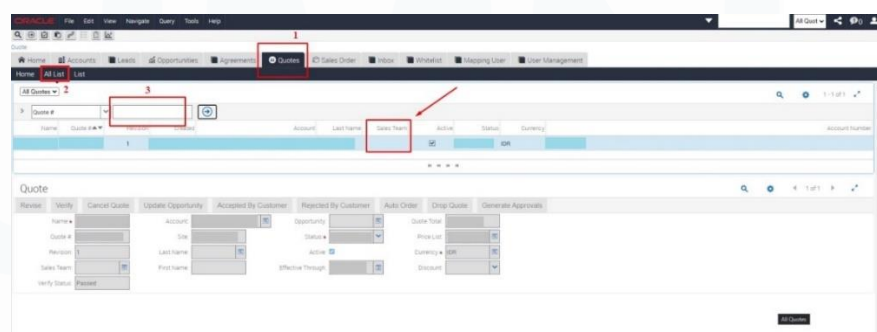
Pada gambar 3.63 dan 3.64 dapat dilihat bahwa ada data yang *abandoned* dikolom `LAST_ORDER_STATUS`, data tersebut didapatkan dari query *previous order* yang telah dilakukan sebelumnya di SQL Developer. Lalu untuk kolom `STATUS_LAYANAN_DASH_SIMS` adalah kolom hasil VLOOKUP dari aplikasi OSS dan SQL Developer, untuk *customer* yang memiliki status layanan UP berarti aplikasi/kabel/alat dari *customer* masih berjalan di lapangan, data ini lah yang nanti akan dijadikan temuan audit oleh tim DCS.

5. Tahapan yang terakhir yang dilakukan oleh tim DaMan adalah memberikan keseluruhan excel dari hasil VLOOKUP

ke tim DCS dan RevAs agar dapat dilihat apakah data yang sudah di *profiling* memang betul keadaannya seperti masalah yang terjadi atau tidak dan apakah data *customer* tersebut memiliki *revenue* atau tidak.

D.8 Assign quote di aplikasi NCX

Sumber *request assign quote* dari aplikasi NCX didapatkan dari Whatsapp/Telegram/E-mail terkait *assign quote* di aplikasi NCX. Tujuan dari *Assign Quote* adalah untuk melihat *quote* dari *user* yang bersangkutan apabila ada beberapa kejadian. Untuk contoh permasalahan adalah apabila AM lama sudah pensiun dan data *quote* dari *customer* A masih berada dibawah naungan AM lama maka AM baru belum bisa melihat data dari *customer* tersebut karena data *customer* tersebut masih terdaftar di AM yang lama. Oleh karena itu dilakukan *assign quote* untuk merubah data AM yang bersangkutan. Untuk tahapan dari *assign quote* di aplikasi NCX dapat dilihat pada gambar 3. 65 dengan memasukkan nomor *quote* dari *customer* dan memilih *sales team* dengan memilih AM yang baru.



Gambar 3. 65 Tampilan dan tahapan *assign quote*

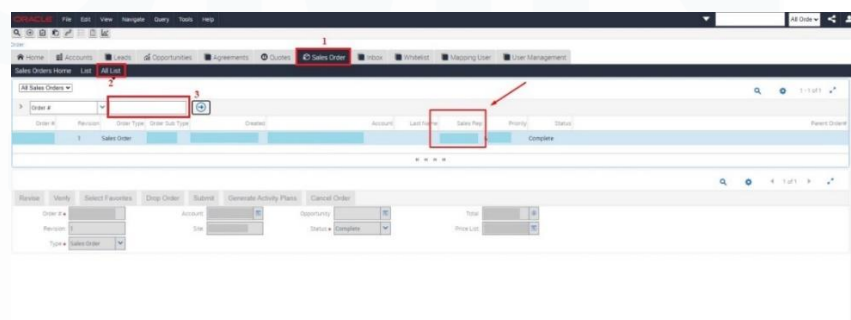
D.9 Assign order di aplikasi NCX

Sumber *request assign order* dari aplikasi NCX didapatkan dari Whatsapp/Telegram/E-mail terkait *assign order* di aplikasi NCX. Tujuan dari *assign order* kurang lebih sama seperti *quote* tapi tujuannya untuk melihat *order* dari *user* yang bersangkutan.



Gambar 3. 66 Proses *assign quote* dan *assign order*

Pada dasarnya proses yang terjadi *assign quote* dan *assign order* pada aplikasi NCX mengikuti proses pada gambar 3. 66, yang pertama adalah melakukan pengisian quote pada aplikasi NCX dan melakukan *request* sampai quote tersebut disetujui. Lalu melakukan *provisioning* yaitu pengecekan jaringan sampai jaringan tersebut *complete* atau jaringan tersebut sudah berhasil atau memenuhi syarat untuk pemasangan alat. Kemudian ada proses *pending BASO* dimana *pending* ini terdapat di aplikasi NCX, *pending* ini ada ketika jaringan adanya pembuatan *request* pada jaringan yang sudah *complete* sampai disetujui melalui aplikasi NCX. Kemudian dilakukan pengecekan tagihan ke pelanggan yang nantinya akan masuk ke TIBS melalui *pending billing approval*. Setelah seluruh proses selesai dilakukan maka di dapatkan hasil *order*. Dan untuk tahapan pada *assign order* dapat dilihat pada gambar 3. 67.



Gambar 3. 67 Tampilan dan tahapan *assign order*

3.3 Kendala yang Ditemukan

Selama pelaksanaan program kerja magang di PT. Telekomunikasi Indonesia sebagai *Intern DaMan*, terdapat beberapa kendala yang dialami oleh mahasiswa, diantaranya adalah sebagai berikut:

1. Kurangnya pengetahuan mahasiswa mengenai bahasa-bahasa *internal* seperti akronim atau singkatan yang digunakan pada perusahaan sehingga dapat menyebabkan keterlambatan dalam segi pengetahuan, komunikasi serta pemahaman mahasiswa untuk pekerjaan atau pembahasan yang akan dilakukan selanjutnya.
2. Keterbatasan dalam mengakses *website* atau aplikasi milik perusahaan sehingga mahasiswa tidak dapat melakukan latihan secara mandiri.

3.4 Solusi atas Kendala yang Ditemukan

Berikut merupakan beberapa solusi yang dapat digunakan dalam mengatasi kendala-kendala yang dihadapi oleh mahasiswa selama proses program kerja magang di PT. Telekomunikasi Indonesia berlangsung, diantaranya sebagai berikut:

1. Membiasakan mahasiswa untuk bertanya ketika bahasa-bahasa yang kurang dipahami muncul, dengan ini maka dapat membantu mahasiswa juga dalam meningkatkan *soft skills* dalam kemampuan dalam berkomunikasi.
2. Ketika *officer* atau *supervisor* ada di kantor maka mahasiswa meminta izin untuk melakukan akses *website* atau aplikasi perusahaan agar tetap bisa di pantau oleh *officer* atau *supervisor* dan tetap bisa melakukan latihan mandiri di kantor sesuai dengan kendala kedua yang terjadi pada perusahaan.