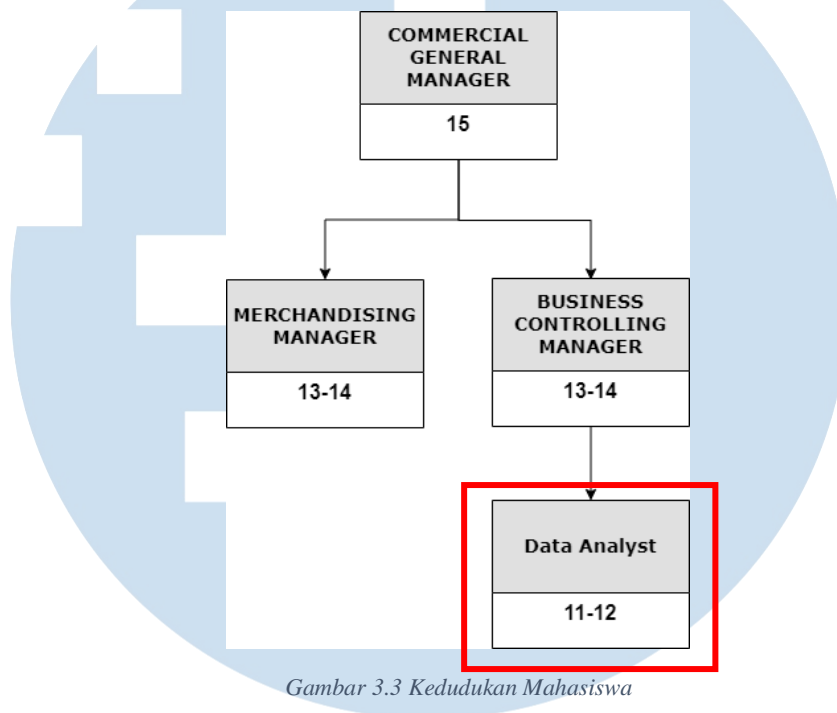


BAB III

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan Dalam Kerja Magang



Gambar 3.3 Kedudukan Mahasiswa

Dapat dilihat pada **gambar 3.1**, merupakan kedudukan dan jabatan yang diikuti selama melaksanakan program kerja magang pada PT Sumber Trijaya Lestari. Posisi yang ditempatkan dalam program kerja magang ada pada divisi *Business Controlling* yang berperan sebagai *Data Analyst*. Dalam pelaksanaan program kerja magang, pelaksanaan program kerja magang dibimbing oleh Ibu Tatung Nur Pertiwi sebagai *Business Controlling Manager* yang memiliki peran untuk memastikan data yang akan dikeluarkan sudah siap dan benar untuk keperluan divisi lain. *Data Analyst* memiliki tanggung jawab untuk mengolah data mentah untuk menjadi data yang sudah siap untuk digunakan untuk keperluan divisi lain.

3.2 Tugas dan Uraian dalam Kerja Magang

3.2.1 Tugas Kerja Magang

Sebagai seorang *Data Analyst* di PT. Sumber Trijaya Lestari, ada beberapa tugas dan tanggung jawab yang dipercayakan dan harus dikerjakan. Tugas yang diberikan beragam, mulai dari pembuatan

dashboard berbasis *web* ataupun data dalam bentuk *excel*. Dikarenakan PT Sumber Trijaya Lestari belum memiliki integrasi data yang bagus, proses perpindahan data biasanya dilakukan dengan menggunakan *tools Talend Open Studio*, untuk melakukan proses transmigrasi data dari berbagai sumber *database* untuk dijadikan satu sumber.

3.2.2 Uraian Pelaksanaan Kerja Magang

Tabel 3.1 Pekerjaan yang dilakukan

| NO | JENIS PEKERJAAN | START | END |
|---|--|-----------|-----------|
| Mempelajari tools dan database yang akan digunakan | | | |
| A.1 | Mempelajari aplikasi <i>Dbeaver</i> | 10-Jan-22 | 13-Jan-22 |
| A.2 | Mempelajari proses ETL dengan menggunakan aplikasi <i>Talend Open Studio</i> | 14-Jan-22 | 18-Jan-22 |
| A.3 | Mempelajari Web SpagoBI Server | 18-Jan-22 | 20-Jan-22 |
| A.4 | Mempelajari MS Excel untuk pivot table | 21-Jan-22 | 21-Jan-22 |
| Membuat Automail untuk menampilkan report terkait akuisisi member | | | |
| B.1 | Melakukan Proses ETL dari <i>Postgre</i> ke <i>MySql</i> | 24-Jan-22 | 24-Jan-22 |
| B.2 | Membuat <i>query</i> | 25-Jan-22 | 28-Jan-22 |
| B.3 | <i>Import query</i> pada <i>Microsoft Excel</i> | 31-Jan-22 | 31-Jan-22 |
| B.4 | Membuat program <i>auto -mail</i> | 02-Feb-22 | 04-Feb-22 |
| Membuat Automail untuk menampilkan report terkait National Sales | | | |
| C.1 | Melakukan Proses ETL dari <i>Oracle</i> ke <i>MySql</i> | 07-Feb-22 | 07-Feb-22 |
| C.2 | Membuat <i>query</i> | 08-Feb-22 | 09-Feb-22 |
| C.3 | <i>Import query</i> pada <i>Microsoft Excel</i> | 09-Feb-22 | 09-Feb-22 |
| C.4 | Membuat program <i>auto -mail</i> | 10-Feb-22 | 11-Feb-22 |
| Membuat Automail untuk menampilkan report terkait informasi transaksi member | | | |
| D.1 | Melakukan Proses ETL dari <i>Postgre</i> ke <i>MySql</i> | 14-Feb-22 | 14-Feb-22 |

| | | | | |
|---|--|--|-----------|-----------|
| | D.2 | Membuat <i>query</i> | 15-Feb-22 | 17-Feb-22 |
| | D.3 | <i>Import query</i> pada <i>Microsoft Excel</i> | 17-Feb-22 | 17-Feb-22 |
| | D.4 | Membuat program <i>auto -mail</i> | 18-Feb-22 | 18-Feb-22 |
| | Membuat Automail untuk menampilkan report kinerja setiap toko | | | |
| E | E.1 | Membuat <i>query</i> | 21-Feb-22 | 25-Feb-22 |
| | E.2 | <i>Import query</i> pada <i>Microsoft Excel</i> | 01-Mar-22 | 01-Mar-22 |
| | E.3 | Membuat program <i>auto -mail</i> | 02-Mar-22 | 04-Mar-22 |
| | Membuat Automail untuk menampilkan report transaksi voucher | | | |
| F | F.1 | Melakukan Proses ETL dari <i>Postgre</i> ke <i>MySql</i> | 07-Mar-22 | 09-Mar-22 |
| | F.2 | Membuat <i>query</i> | 10-Mar-22 | 16-Mar-22 |
| | F.3 | <i>Import query</i> pada <i>Microsoft Excel</i> | 17-Mar-22 | 17-Mar-22 |
| | F.4 | Membuat program <i>auto -mail</i> | 18-Mar-22 | 23-Mar-22 |

3.2.2.1 Mempelajari *tools* dan *database* yang akan digunakan

Pada minggu pertama, hal yang dilakukan pada program kerja magang yang dilakukan yaitu pengenalan pada divisi BI yang akan bekerja sama nantinya untuk mengeluarkan *reporting* untuk kebutuhan divisi yang ada serta pembuatan *dashboard*. Kemudian, setelah itu di jelaskan oleh *BC Manager* terkait proyek apa saja yang telah *pending* untuk dibuat *reporting* nantinya.

Tugas yang pertama kali diberikan adalah untuk melakukan eksplorasi terhadap *software* apa saja yang akan dipakai untuk membantu menjalan tugasnya nanti sebagai *Data Analyst*.

Software yang akan digunakan yaitu, *Dbeaver*, *Robo3t*, *Talend Open Studio*, *SpagoBI Server*.



Gambar 3.4 Aplikasi DBeaver

Dbeaver merupakan alat manajemen basis data *multiplatform open source* dan *SQL Client* untuk pengembang dan administrator basis data. *DBeaver* yaitu **Gambar 3.2**, diatas dapat digunakan untuk mengakses *database* seperti *Oracle, SQL Server, MySQL, Postgresql*. Aplikasi ini biasanya digunakan untuk mengolah *database* dan melakukan *data cleansing* dan terhadap aplikasi yang sedang di *develop* apakah data nya masuk ke *database* atau tidak.

Mahasiswa sendiri diberikan akses *database* sebanyak 13 db yang aktif untuk divisi BI yang terdiri dari “Aksesmu_BI”, “mro_report”, “mro_trx_prd”, “prd_promo_stl_v2”, “SATECOMM”, “STLDCSBGR”, “STLDCSJBR”, “STLDCSCKK”, “STLDCSSMG”, “STLIMSCKK”, “STLIMSJBR”, “STLIMSBGR”, “STLIMSSMG”. Masing – masing *database* memiliki kebutuhannya masing – masing seperti “Aksesmu_BI” digunakan untuk tempat menyimpan *mirror* data dari berbagai tabel yang ada pada *database* yang lain.



Gambar 3.5 Aplikasi Talend

Aplikasi selanjutnya yang digunakan untuk menunjang pekerjaan magang sebagai seorang *Data Analyst* adalah *talend*, yang bisa dilihat pada **gambar 3.3**. Aplikasi ini merupakan salah satu dari sekian aplikasi penunjang *big data* yang bersifat *open source* jadi pengguna dapat dengan mudah mengakses aplikasi ini tanpa harus membayar atau berlangganan untuk menggunakannya. Aplikasi ini digunakan untuk melakukan *data mapping* atau melakukan proses ETL. ETL merupakan istilah yang

kepanjangannya *Extract Transform Load*. Kelebihan dari aplikasi *talend* ini yaitu, *user* dapat dengan mudah melakukan ETL hanya dengan *drag n drop tools* yang sudah disediakan dalam aplikasi. *Talend* sendiri digunakan untuk melakukan transmigrasi data dari *database table to database table*, ataupun *excel to database table*.



Gambar 3.6 Aplikasi SpagoBI Server

Aplikasi yang berikutnya yaitu *SpagoBI Server*, yang bisa dilihat pada **gambar 3.4**. Aplikasi ini sangat penting bagi seorang *data analyst* yang ingin melakukan proses otomatisasi *job ETL* yang telah dibuat pada *local*. *SpagoBI* biasa digunakan untuk melakukan otomatisasi *ETL job* yang telah di *deploy* dari *Talend* untuk melakukan *scheduling* kapan *job* tersebut harus berjalan.

3.2.2.2 Membuat Automail Untuk Menampilkan Report Terkait Akuisisi Member

Tugas yang pertama yaitu untuk pembuatan *reporting* akuisisi member yang dimana berisikan performa setiap cabang yang di miliki pada Aksesmu untuk melihat berapa banyak jumlah *member* baru yang daftar pada setiap harinya untuk periode setiap bulan. Akuisisi member dibagi menjadi 2 kategori, kategori pertama merupakan akuisisi *member non* mandiri yang berarti *member* baru yang daftar pada Aksesmu tidak mengunduh aplikasi lewat *playstore* dan melakukan *register*, melainkan *member* baru ini didaftarkan oleh petugas atau biasa disebut MRO (*Member Relation Officer*). *Reporting* ini sangat penting bagi kepala cabang untuk melakukan evaluasi apakah target harian akuisisi *member* terpenuhi atau tidak.

Proses pertama untuk pembuatan *reporting* ini adalah memastikan data apa saja yang dibutuhkan untuk dilakukan *query*

nantinya. Setelah didapatkan *table* apa saja yang dibutuhkan untuk membuat *reporting*, proses selanjutnya adalah melakukan *query* untuk menarik data apa saja yang ingin di *transfer* ke *database* BI, seperti yang ada pada **gambar 3.5**.

```
select id,
       email,
       password as password,
       salt,
       oba_id,
       oba_online,
       branch_id,
       store_code,
       company_id,
       group_ids,
       split_part(TRIM(location_coordinate ::text,'()'),',',1)latitude,
       split_part(TRIM(location_coordinate ::text,'()'),',',2)longitude,
       registered_at,
       is_account_active,
       is_activated,
       activated_at,
       is_verified,
       is_blacklist,
       is_email_verified,|
       is_old_member,
       is_pin_created,
       updated_at,
       verified_at
from tb_ama_login_auth_user talau
```

Gambar 3.7 Query 1

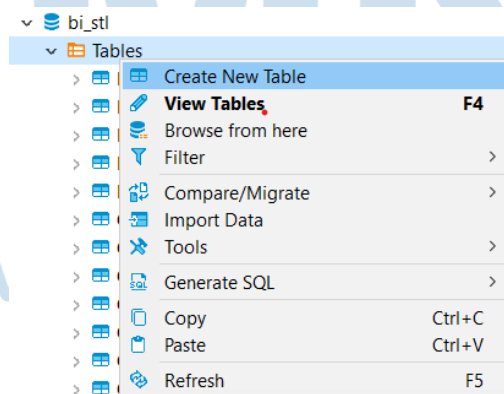
Proses berikutnya adalah pembuatan *table* sebagai sasaran proses transaksi data tersebut. *Table* tersebut terdapat pada *database* BI yang akan disebut “tb_ama_login_auth_user”. Proses pembuatan *table* tersebut dapat menggunakan dua cara antara lain dengan fitur *Dbeaver* atau dengan melakukan *query* yang dapat dilihat pada **gambar 3.6** dibawah ini.



```
-- bi_stl.tb_ama_login_auth_user_ definition
CREATE TABLE `tb_ama_login_auth_user_` (
  `_id` varchar(40) NOT NULL,
  `email` varchar(200) DEFAULT NULL,
  `password` varchar(200) DEFAULT NULL,
  `salt` varchar(200) DEFAULT NULL,
  `obaId` varchar(60) NOT NULL,
  `obaOnline` int(60) DEFAULT NULL,
  `branchId` varchar(60) DEFAULT NULL,
  `storeCode` varchar(60) DEFAULT NULL,
  `companyId` varchar(60) DEFAULT NULL,
  `groupIds` varchar(100) DEFAULT NULL,
  `latitude` varchar(60) DEFAULT NULL,
  `longitude` varchar(60) DEFAULT NULL,
  `registeredAt` timestamp NULL DEFAULT NULL,
  `isAccountActive` int(60) DEFAULT NULL,
  `isActivated` int(3) DEFAULT NULL,
  `activatedAt` timestamp NULL DEFAULT NULL,
  `isVerified` int(60) DEFAULT NULL,
  `isBlacklist` int(60) DEFAULT NULL,
  `isEmailVerified` int(60) DEFAULT NULL,
  `isOldMember` int(60) DEFAULT NULL,
  `isPinCreated` int(60) DEFAULT NULL,
  `updatedAt` timestamp NULL DEFAULT NULL,
  `verifiedAt` timestamp NULL DEFAULT NULL,
  PRIMARY KEY (`_id`,`obaId`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Gambar 3.8 Query 2

Dalam pembuatan table dibutuhkan tipe data setiap atribut atau kolom yang ada pada table tersebut yang ada pada gambar diatas. Karena proses transmigrasi data dilakukan dari *Postgresql* ke *MYSQL* jadi harus dipastikan terlebih dahulu apakah tipe data yang digunakan mendukung atau tidak. Cara kedua untuk membuat tabel yaitu dengan aplikasi *Dbeaver* yang dapat dilihat pada **gambar 3.7** berikut.



Gambar 3.9 Fitur Aplikasi Dbeaver

Dengan menggunakan fitur dari aplikasi *DBeaver*, *user* tidak perlu repot untuk memikirkan *function* apa yang harus dipakai

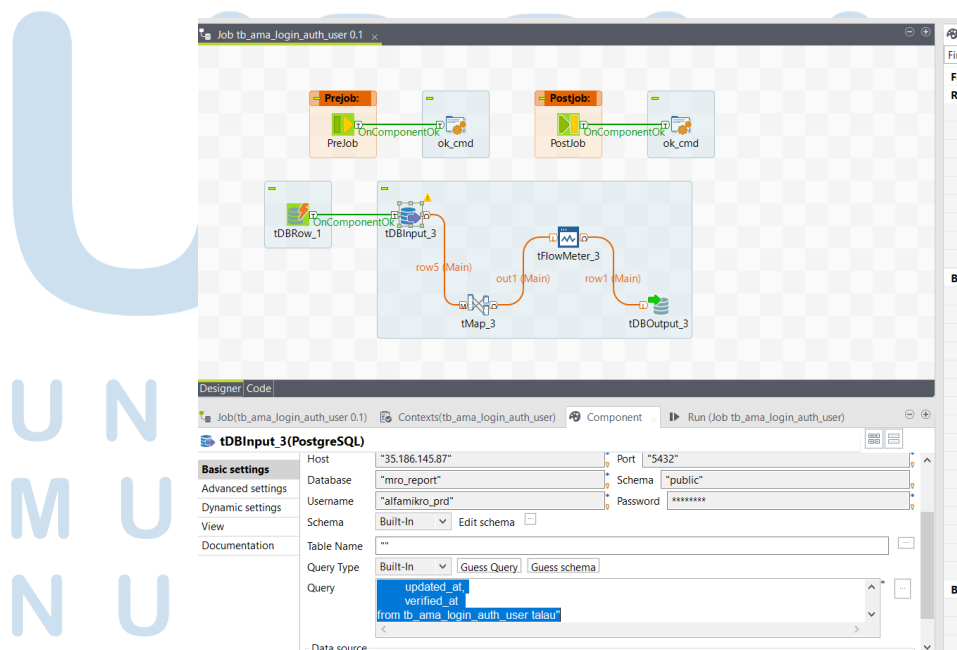
dan hanya tinggal mengisi kolom dan memilih tipe data apa yang akan digunakan pada kolom tersebut pada seperti yang terlihat pada **gambar 3.8**

| Column Name | # | Data Type | Not Null | Auto Increment | Key |
|-----------------|----|------------------|----------|----------------|-----|
| id | 1 | varchar(40) | [v] | [] | PRI |
| email | 2 | BIGINT | [] | [] | |
| password | 3 | BIGINT UNSIGNED | [] | [] | |
| salt | 4 | BINARY | [] | [] | |
| obald | 5 | BIT | [v] | [] | PRI |
| obaOnline | 6 | BLOB | [] | [] | |
| branchId | 7 | CHAR | [] | [] | |
| storeCode | 8 | DATE | [] | [] | |
| companyId | 9 | DATETIME | [] | [] | |
| groupIds | 10 | DECIMAL | [] | [] | |
| latitude | 11 | DOUBLE | [] | [] | |
| longitude | 12 | DOUBLE PRECISION | [] | [] | |
| registeredAt | 13 | ENUM | [] | [] | |
| icAccountActive | 14 | FLOAT | [] | [] | |
| | | INT | [] | [] | |

Gambar 3.10 Pembuatan tabel aplikasi Dbeaver

Selain dapat digunakan untuk membuat *table*, *DBeaver* juga dapat digunakan untuk membuat *Primary Key* dan juga *Indexes* pada *table* yang dapat mempercepat proses *query* jika ingin melakukan *JOIN* pada banyak *table* dan ukuran pada *table* tersebut sangat besar.

Setelah itu proses selanjutnya membuat *map* untuk proses ETL pada aplikasi *Talend* yang dapat dilihat prosesnya pada **gambar 3.9**.



Gambar 3.11 Mapping ETL

Dapat dilihat dari gambar diatas merupakan *mappingan* untuk melakukan proses ETL pada *table* “tb_ama_login_auth_user”. Proses tersebut meliputi tDBRow1 yang digunakan untuk melakukan *truncate* pada *table* yang akan menjadi tujuan atau target pemindahan data tersebut supaya sebelum diisi data yang baru, *table* tersebut akan dikosongkan terlebih dahulu. Setelah itu membuat komponen tDinput yang digunakan untuk sumber data yang akan digunakan untuk pembuatan skema dan data apa saja yang akan di *transfer* nantinya ke *table* yang akan menjadi sasaran. Pembuatan skema tersebut dapa dilihat pada **gambar 3.10**.

| Column | Db Column | K. | Type | DB Type | <input checked="" type="checkbox"/> N. | Date Pattern (Ctrl+...) |
|-----------------|-----------------|-------------------------------------|---------|------------|--|-------------------------|
| _id | _id | <input checked="" type="checkbox"/> | String | VARCHAR | <input checked="" type="checkbox"/> | |
| email | email | <input type="checkbox"/> | String | VARCHAR | <input checked="" type="checkbox"/> | |
| password | password | <input type="checkbox"/> | String | VARCHAR | <input checked="" type="checkbox"/> | |
| salt | salt | <input type="checkbox"/> | String | VARCHAR | <input checked="" type="checkbox"/> | |
| obald | obald | <input checked="" type="checkbox"/> | String | VARCHAR | <input type="checkbox"/> | |
| obaOnline | obaOnline | <input type="checkbox"/> | Integer | INT4 | <input checked="" type="checkbox"/> | |
| branchId | branchId | <input type="checkbox"/> | String | VARCHAR | <input checked="" type="checkbox"/> | |
| storeCode | storeCode | <input type="checkbox"/> | String | VARCHAR | <input checked="" type="checkbox"/> | |
| companyId | companyId | <input type="checkbox"/> | String | VARCHAR | <input checked="" type="checkbox"/> | |
| groupIds | groupIds | <input type="checkbox"/> | String | VARCHAR | <input checked="" type="checkbox"/> | |
| latitude | latitude | <input type="checkbox"/> | String | TEXT | <input checked="" type="checkbox"/> | |
| longitude | longitude | <input type="checkbox"/> | String | TEXT | <input checked="" type="checkbox"/> | |
| registeredAt | registeredAt | <input type="checkbox"/> | Date | TIMESTA... | <input checked="" type="checkbox"/> | "dd-MM-yyyy" |
| isAccountActive | isAccountActive | <input type="checkbox"/> | Integer | INT4 | <input checked="" type="checkbox"/> | |
| isActivated | isActivated | <input type="checkbox"/> | Integer | INT4 | <input checked="" type="checkbox"/> | |
| activatedAt | activatedAt | <input type="checkbox"/> | Date | TIMESTA... | <input checked="" type="checkbox"/> | "dd-MM-yyyy" |
| isVerified | isVerified | <input type="checkbox"/> | Integer | INT4 | <input checked="" type="checkbox"/> | |
| isBlacklist | isBlacklist | <input type="checkbox"/> | Integer | INT4 | <input checked="" type="checkbox"/> | |
| isEmailVerified | isEmailVerified | <input type="checkbox"/> | Integer | INT4 | <input checked="" type="checkbox"/> | |
| isOldMember | isOldMember | <input type="checkbox"/> | Integer | INT4 | <input checked="" type="checkbox"/> | |
| isPinCreated | isPinCreated | <input type="checkbox"/> | Integer | INT4 | <input checked="" type="checkbox"/> | |
| updatedAt | updatedAt | <input type="checkbox"/> | Date | TIMESTA... | <input checked="" type="checkbox"/> | "dd-MM-yyyy" |
| verifiedAt | verifiedAt | <input type="checkbox"/> | Date | TIMESTA... | <input checked="" type="checkbox"/> | "dd-MM-yyyy" |

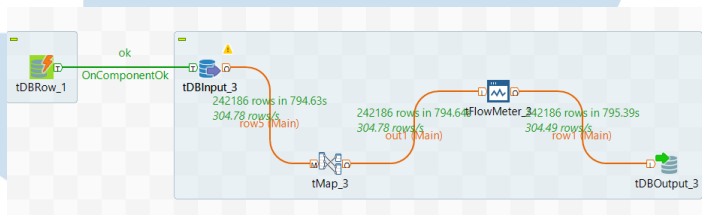
Gambar 3.12 Skema Pemetaan

Pembuatan skema ini merupakan proses untuk pemetaan tipe data yang diambil dari sumber data. Dalam kasus ini sumber data yang dimaksud merupakan “tb_ama_login_auth_user” yang diambil dari *database Postgresql*. Untuk pemilihan tipe data harus disamakan dengan yang ada pada *table* tersebut. Untuk proses selanjutnya ada tMap yang merupakan komponen untuk melakukan pemetaan dari *table A* ke *table B* yang prosesnya dapat dilihat pada **gambar 3.11**

| Column | Expression | Column |
|-----------------|-----------------|-------------|
| id | no5.id | id |
| nama | no5.nama | nama |
| alamat | no5.alamat | alamat |
| no5 | no5.no5 | no5 |
| no5.kelurahan | no5.kelurahan | kelurahan |
| no5.kecamatan | no5.kecamatan | kecamatan |
| no5.kabupaten | no5.kabupaten | kabupaten |
| no5.kode | no5.kode | kode |
| no5.kode_pos | no5.kode_pos | kode_pos |
| no5.kode_pos_2 | no5.kode_pos_2 | kode_pos_2 |
| no5.kode_pos_3 | no5.kode_pos_3 | kode_pos_3 |
| no5.kode_pos_4 | no5.kode_pos_4 | kode_pos_4 |
| no5.kode_pos_5 | no5.kode_pos_5 | kode_pos_5 |
| no5.kode_pos_6 | no5.kode_pos_6 | kode_pos_6 |
| no5.kode_pos_7 | no5.kode_pos_7 | kode_pos_7 |
| no5.kode_pos_8 | no5.kode_pos_8 | kode_pos_8 |
| no5.kode_pos_9 | no5.kode_pos_9 | kode_pos_9 |
| no5.kode_pos_10 | no5.kode_pos_10 | kode_pos_10 |
| no5.kode_pos_11 | no5.kode_pos_11 | kode_pos_11 |
| no5.kode_pos_12 | no5.kode_pos_12 | kode_pos_12 |
| no5.kode_pos_13 | no5.kode_pos_13 | kode_pos_13 |
| no5.kode_pos_14 | no5.kode_pos_14 | kode_pos_14 |
| no5.kode_pos_15 | no5.kode_pos_15 | kode_pos_15 |
| no5.kode_pos_16 | no5.kode_pos_16 | kode_pos_16 |
| no5.kode_pos_17 | no5.kode_pos_17 | kode_pos_17 |
| no5.kode_pos_18 | no5.kode_pos_18 | kode_pos_18 |
| no5.kode_pos_19 | no5.kode_pos_19 | kode_pos_19 |
| no5.kode_pos_20 | no5.kode_pos_20 | kode_pos_20 |

Gambar 3.13 Pemetaan table

Kemudian komponen terakhir adalah tDOuput yang merupakan komponen untuk sasaran dari transmigrasi data tersebut mau di kirim kemana. Untuk kasus yang ini yang menjadi target merupakan *database* “Alfamikro_BI”. Setelah semuanya sudah dibuat langkah terakhir yaitu melakukan *running job* tersebut yang hasilnya akan ditunjukkan pada **gambar 3.12**



Gambar 3.14 Running Job 1

Pada gambar diatas dapat dilihat berapa kecepatan untuk proses transaksi datanya, cepat atau lambatnya proses transmigrasi data tersebut dapat dipengaruhi oleh dua hal yang pertama yaitu bagus atau tidaknya *query* dan yang kedua yaitu koneksi yang digunakan. Dari pengalaman yang telah didapatkan proses transmigrasi sering kali gagal atau lama dikarenakan koneksi yang dimiliki tidak cepat yang mengakibatkan proses transmigrasi data terhenti ditengah jalan atau kasus yang lebih parah transmigrasi data gagal.

Proses selanjutnya yaitu membuat *query* untuk report yang akan dikeluarkan nantinya. Salah satu contoh *query* nya dapat ditunjukkan pada **gambar 3.13**

UNIVERSITAS
MULTIMEDIA
NUSANTARA

```

3919 SELECT a.STATUS_MEMBER,
3920 SUM(a.1+a.2+a.3+a.4+a.5+a.6+a.7+a.8+a.9+a.10+a.11+a.12+a.13+a.14+a.15+a.16+a.17+a.18+a.19+a.20+a.21
3921 a.1,a.2,a.3,a.4,a.5,a.6,a.7,a.8,a.9,a.10,a.11,a.12,a.13,a.14,a.15,a.16,a.17,a.18,a.19,a.20,a.21,a.2
3922
3923 FROM(
3924 SELECT
3925 CASE WHEN tm.created_by_user = 'self_registration' then 'MANDIRI'
3926 else 'NON MANDIRI'
3927 end as STATUS_MEMBER,
3928 COUNT(case when extract(day from registeredAt)=1 then 1 end) as '1',
3929 COUNT(case when extract(day from registeredAt)=2 then 1 end) as '2',
3930 COUNT(case when extract(day from registeredAt)=3 then 1 end) as '3',
3931 COUNT(case when extract(day from registeredAt)=4 then 1 end) as '4',
3932 COUNT(case when extract(day from registeredAt)=5 then 1 end) as '5',
3933 COUNT(case when extract(day from registeredAt)=6 then 1 end) as '6',
3934 COUNT(case when extract(day from registeredAt)=7 then 1 end) as '7',
3935 COUNT(case when extract(day from registeredAt)=8 then 1 end) as '8',
3936 COUNT(case when extract(day from registeredAt)=9 then 1 end) as '9',
3937 COUNT(case when extract(day from registeredAt)=10 then 1 end) as '10',
3938
3939 )
3940
3941
3942
3943
3944
3945
3946
3947
3948
3949
3950
3951
3952
3953
3954
3955
3956
3957
3958
3959
3960
3961
3962
3963
3964
3965
3966
3967
3968
3969
3970
3971
3972
3973
3974
3975
3976
3977
3978
3979
3980
3981
3982
3983
3984
3985
3986
3987
3988
3989
3990
3991
3992
3993
3994
3995
3996
3997
3998
3999
4000

```

| STATUS_MEMBER | TOTAL | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | |
|---------------|-------|----|-----|-----|-----|----|-----|-----|-----|-----|-----|-----|----|-----|-----|----|----|----|----|----|----|----|--|
| MANDIRI | 4759 | 26 | 248 | 269 | 259 | 22 | 258 | 338 | 278 | 332 | 309 | 258 | 17 | 312 | 341 | | | | | | | | |
| NON MANDIRI | 4266 | 20 | 223 | 245 | 207 | 8 | 267 | 260 | 278 | 283 | 254 | 274 | 8 | 308 | 316 | | | | | | | | |

Gambar 3.15 Query 3

Dalam pembuatan *report*, PT STL menggunakan program yang bernama *Automail*. Dimana *automail* sendiri merupakan proses otomatisasi pengiriman email berbasis PHP, HTML, dan juga CSS. Dalam pembuatan *program automail* akan dibimbing oleh *Data Mining Specialist Mas Rifan*. Untuk pembuatan program *automail* ini hanya akan melakukan *copy paste* dan modifikasi program yang sudah ada sebelumnya. Langkah pertama untuk melakukan modifikasi yaitu membukan *folder controller* yang ada pada “*automail_akuisisi_member*” kemudian mengubah *file .php* tersebut seperti **gambar 3.14**

```

<?php if (! defined('BASEPATH')) exit('No direct script access allowed');

class akuisisi_nasional extends CI_Controller {

    public function send_akuisisi_nasional_report()

        $runCommand = 'C:\BusinessIntelligence\Automail\AIFamiko\vbs\MemberAkuisisiNasional.vbs -t 60';
        echo "<br> run Command : ".shell_exec($runCommand);

        $this->load->library('email');

        $message = $this->load->view('automail_nasional', '', true);
        $receiptto = array(
            'tri.bartoyo@aksesmu.id',
            'padiyanto@aksesmu.id',
            'saryadi.yadi@aksesmu.id',
            'oscar.manik@aksesmu.id',
            'taufik.hidayanto@aksesmu.id'
        );

        /*ini untuk live email*/
        $result = $this->email
            ->From('bi@alfacart.com')
            ->To($receiptto)
            ->Cc('lukedarmawan@gmail.com')
            ->Cc('hanna.c.p.hapsari@alfadigital.id')
            ->Cc('tatung.n.pertiwi@alfadigital.id')
            ->Cc('yosef.risdianto@aksesmu.id')
            ->Cc('tri.w.sputro@aksesmu.id')
            ->subject(['Automail'] Monitoring Akuisisi Member Nasional')
            ->message($message)
            ->send();

```

Gambar 3.16 Query 4

Pada *folder controller* ini digunakan untuk mengatur bagaimana penerima dan juga subjek yang akan ditampilkan nantinya. Langkah berikutnya merupakan pembuatan dashboard pada *google mail* yang akan dibuat melalui *folder views*. *Folder views* ini diisi juga dengan *file .php* yang berisi *function* untuk

memanggil *query* pada sumber data yang ingin dipanggil dengan fungsi dibawah berikut seperti **gambar 3.15**

```
error_reporting(0);  
$connec = new PDO('mysql:host=10.222.12.154:3306;dbname=bi_stl','bi_stl','BI_stl_2021');  
$sql = "
```

Gambar 3.17 Query 5

Dimana selanjutnya merupakan pembuatan *table* berbasis HTML yang dapat ditunjukkan pada **gambar 3.16**.

```
<br><br>  
<b>Akuisisi Member Nasional</b>  
<br>  
<table border='4px' bordercolor='#00000'>  
<td>  
<?php  
$color = '#629Pf5';  
echo  
"  
<table cellpadding:'700'border:'10'>  
<font size='3'>  
<h4 style=text-align:center;>KETERANGAN</h4>  
<tr bgcolor='#629df5'>  
<th tr bgcolor='#d1d1d1'>Status Member</th>  
</tr>;  
  
foreach ($connec->query($sql) as $row)  
{  
$color = ($color == '#efedf0') ? '#d1d1d1' : '#efedf0';  
echo "<tr bgcolor='\" . $color . \"'>  
<td style= white-space:nowrap;>\" . $row['STATUS_MEMBER'].\"</td>  
</tr>\"  
};  
echo "<tr bgcolor='#629df5'>  
<td tr bgcolor='#d1d1d1'>\" . \"NEW MEMBER\". \"</td>  
</tr>  
</td>\"  
};  
foreach ($connec->query($sql) as $row)  
{  
$color = ($color == '#efedf0') ? '#d1d1d1' : '#efedf0';  
echo "<tr bgcolor='\" . $color . \"'>  
<td style= white-space:nowrap;>\" . $row['STATUS_MEMBER'].\"</td>
```

Gambar 3.18 Query 6

Setelah dilakukan modifikasi program *automail* yang sudah ada, maka diperlukannya *sheduler* untuk mengatur kapan dijalankannya program tersebut dengan menggunakan *Task Scheduler* yang dapat dilihat pada **gambar 3.17**

MYSQL. Untuk langkah pertama sama seperti sebelumnya, membuat *query* untuk menarik data dari *table* yang ada di *database Oracle*. *Query* tersebut terdiri dari beberapa *table* yang diharuskan menggunakan *function JOIN* untuk mengumpulkan data yang akan di transfer nantinya ke sasaran. *Query* tersebut dapat dilihat pada **gambar 3.19**

```

SELECT
  tgl_trs,
  CASE WHEN LENGTH(KODE_TOKO||'/'||'SSP'||'/'||to_char(tgl_trs, 'MM')||N_PRC
  ELSE kode_toko||'/'||'SS'||n_promo
  END
  AS SHIPMENT_ID,
  trans.KODE_TOKO,
  ast.NAMA_STORE,
  trans.MEMBERS,
  head.plu,
  head.DESCP,
  head.KATAGORI || ' - ' || kat.NAMA_KATA AS KATEGORI,
  head.SUBDEPT || ' - ' || sub.NAM_SUB AS SUB_DEPT,
  head.PABRIK || ' - ' || pabrik.NAMA_PAB AS PRINSIPAL,
  head.S_DEPT || ' - ' || dep.DEPARTEMEN AS DEPARTEMEN,
  head.TAG1_PROD AS TAG,
  sum( trans.qty ) qty,
  round(sum(
    nvl( trans.price, 0 ) * nvl( trans.qty, 0 ) * decode( trans.dec, 1, 1
    1 - nvl( trans.disc, 0 ) ) * (
    1 - ( decode( nvl( trans.ppn, 0 ), 0, 0, 0.11 ) / 1.11 ) ) ) ) net_sales,
  round(sum(
    nvl( trans.qty, 0 ) * decode( nvl( trans.dec, 0 ), 0, 1, 1 / 1000 ) *
    nvl( trans.price, 0 ) * (
    1 - nvl( trans.disc, 0 ) ) * (
    1 - ( decode( nvl( trans.ppn, 0 ), 0, 0, 0.11 ) / 1.11 ) ) - (
    nvl( trans.avg_cost, 0 ) )
    ) ) ) gm,
  count(
  DISTINCT
  CASE
    WHEN TYPE NOT LIKE '%V%'
    AND TYPE NOT LIKE '%R%'
    AND TYPE NOT LIKE '%X%' THEN
      tgl_trs || kode_kasir || no_bon ELSE NULL
    END
  ) struk_qty
FROM
  AMU_TRANS_T trans
  JOIN AMU_MASTER_HEAD_T head ON trans.PLU = head.plu || '2' AND KD_
  LEFT JOIN AMU_PABRIK_T pabrik ON head.PABRIK = pabrik.KODE_PAB
  LEFT JOIN AMU_DEPT_T dep ON head.S_DEPT = dep.DEPT
  LEFT JOIN AMU_KATAGORI_T kat ON kat.NO_KATA = head.KATAGORI

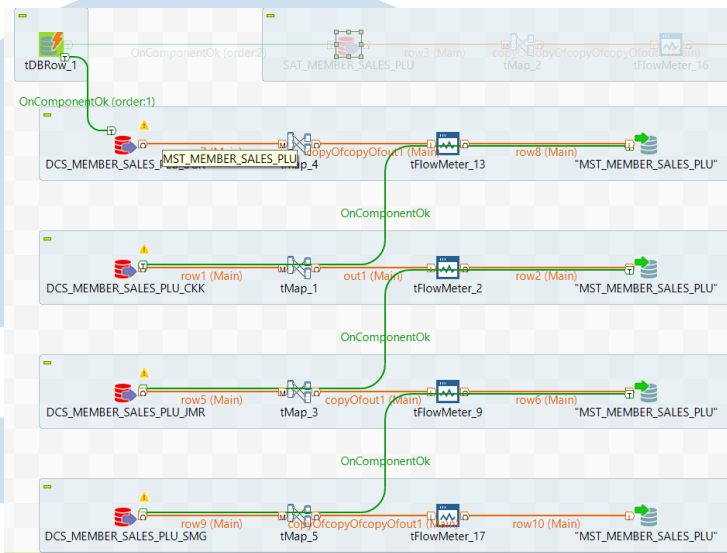
```

Gambar 3.21 Query 7

Langkah selanjutnya membuat *mapping* ETL pada *Talend*.

Pada kasus ini *mapping* ETL dibuat 4 tDInput dikarenakan untuk mengambil data transaksi atau *sales* PT Sumber Trijaya Lestari membagi *database* berdasarkan area yang dimiliki. PT STL memiliki 4 Area yaitu Bogor, Cikokol, Jember, dan Semarang. Oleh karena itu sumber data yang ada dikumpulkan semua menjadi satu kesatuan dalam satu *table* yang nantinya akan

memudahkan untuk mengolah data tersebut. Bentuk *mapping* ETL dapat dilihat pada **gambar 3.20**



Gambar 3.22 Mapping ETL 1

Setelah dibuat proses ETL tersebut proses selanjutnya adalah melakukan *import* data pada excel. File excel ini digunakan untuk melihat data secara detail. Setelah data *mirror* yang telah dibuat sudah ada pada *table* tujuan, data tersebut masih disebut data mentah karena masih belum mengerucut untuk kebutuhannya. Langkah selanjutnya yaitu membuat *query* untuk keperluan data detail yang akan diisi pada excel.

File tersebut nantinya akan *terupdate* secara otomatis dengan menggunakan VBS. Vbs sendiri merupakan program yang sama dengan *Visual Basic Scripting* yang dikembangkan oleh *Microsoft* yang memiliki bahasa pemrograman disebut VBScript. VBS dalam divisi BI digunakan untuk melakukan *refresh* pada suatu file excel yang dapat dilihat *scriptnya* seperti **gambar 3.21**

```
Dim objExcel
Dim objWorkBook, objWorksheet
Set objExcel = CreateObject("Excel.Application")
'Changed By Yudis 6 Aug 2020
Set objWorkBook = objExcel.Workbooks.Open("C:\BusinessIntelligence\Automail\Alfamikro\report\SSP SALES MTD.xlsx")
'Set objWorkBook = objExcel.Workbooks.Open("C:\BusinessIntelligence\Automail\Monthly\report\Monthly Transaction Gomart.xlsx")
'Set objWorksheet = objWorkBook.Worksheets("AvgDisc Report")
objLogFile.WriteLine (" " & DATE & " " & TIME & " Starting Refresh(Monthly Transaction Gomart FIX.xlsx)")
```

Gambar 3.23 VBScript

Untuk langkah selanjutnya yaitu membuat program *automail*, yang akan digunakan untuk melakukan otomisasi pengiriman *email*. Bentuk *dashboard* yang sudah jadi dan sudah dikirimkan dapat dilihat pada **gambar 3.22**

Tampilan monitoring Sales & Margin (Actual vs Budget) Periode MTD sd : 20-Jun-2022

| BRANCH NAME | Budget (NS) | Budget (GM) | % Budget (GM) | Actual NS | Actual GM | % Actual GM | Gap NS | % Actual NS | Gap GM |
|--------------------|----------------------------|---------------------------|---------------|----------------------------|--------------------------|-------------|-----------------------------|--------------|----------------------------|
| DC BALARAJA | Rp. 21.862.500.000 | Rp. 874.500.000 | 4% | Rp. 11.071.933.268 | Rp. 403.564.512 | 3.6% | Rp. -10.790.567.084 | 50.6% | Rp. -470.935.456 |
| DC BALI | Rp. 7.500.000.000 | Rp. 300.000.000 | 4% | Rp. 1.104.503.183 | Rp. 37.722.584 | 3.4% | Rp. -6.395.497.073 | 14.7% | Rp. -262.277.416 |
| DC BANDUNG 1 | Rp. 17.500.000.000 | Rp. 700.000.000 | 4% | Rp. 10.678.957.850 | Rp. 217.985.094 | 2% | Rp. -6.821.042.406 | 61% | Rp. -482.014.906 |
| DC BANDUNG 2 | Rp. 15.962.500.000 | Rp. 638.500.000 | 4% | Rp. 6.247.239.023 | Rp. 125.582.694 | 2% | Rp. -9.715.261.073 | 39.1% | Rp. -512.917.274 |
| DC BANJARMASIN | Rp. 8.250.000.000 | Rp. 330.000.000 | 4% | Rp. 920.198.369 | Rp. 74.561.549 | 8.1% | Rp. -7.329.801.503 | 11.2% | Rp. -255.438.451 |
| DC BEKASI | Rp. 10.400.000.000 | Rp. 416.000.000 | 4% | Rp. 6.262.133.902 | Rp. 259.765.818 | 4.2% | Rp. -4.137.866.098 | 60.2% | Rp. -156.234.182 |
| DC BOGOR | Rp. 13.475.000.000 | Rp. 539.000.000 | 4% | Rp. 4.785.394.872 | Rp. 203.025.866 | 4.2% | Rp. -8.689.605.448 | 35.5% | Rp. -335.974.134 |
| DC CIANJUR | Rp. 9.100.000.000 | Rp. 364.000.000 | 4% | Rp. 2.360.316.474 | Rp. 63.175.527 | 2.7% | Rp. -6.739.683.782 | 25.9% | Rp. -300.824.473 |
| DC CIKOKOL | Rp. 7.412.500.000 | Rp. 296.500.000 | 4% | Rp. 4.544.181.495 | Rp. 222.273.981 | 4.9% | Rp. -2.868.318.473 | 61.3% | Rp. -74.226.019 |
| DC CILACAP | Rp. 20.950.000.000 | Rp. 838.000.000 | 4% | Rp. 4.753.042.678 | Rp. 219.901.192 | 4.6% | Rp. -16.196.956.938 | 22.7% | Rp. -618.098.808 |
| DC CILEUNGS | Rp. 14.400.000.000 | Rp. 576.000.000 | 4% | Rp. 3.563.178.937 | Rp. 141.434.896 | 4% | Rp. -10.836.821.063 | 24.7% | Rp. -434.565.104 |
| DC JAMBI | Rp. 8.250.000.000 | Rp. 330.000.000 | 4% | Rp. 806.428.704 | Rp. 25.161.470 | 3.1% | Rp. -7.443.571.168 | 9.8% | Rp. -304.838.530 |
| DC JEMBER | Rp. 15.275.000.000 | Rp. 611.000.000 | 4% | Rp. 4.896.418.060 | Rp. 271.258.661 | 5.5% | Rp. -10.378.581.748 | 32.1% | Rp. -339.741.339 |
| DC KARAWANG | Rp. 13.700.000.000 | Rp. 548.000.000 | 4% | Rp. 5.334.108.829 | Rp. 179.139.608 | 3.4% | Rp. -8.365.890.915 | 38.9% | Rp. -368.860.392 |
| DC KLATEN | Rp. 18.600.000.000 | Rp. 744.000.000 | 4% | Rp. 11.157.542.116 | Rp. 789.631.344 | 7.1% | Rp. -7.442.457.372 | 60% | Rp. 45.631.344 |
| DC KOTABUMI | Rp. 10.850.000.000 | Rp. 434.000.000 | 4% | Rp. 1.465.044.659 | Rp. 36.304.746 | 2.5% | Rp. -9.384.955.213 | 13.5% | Rp. -397.695.254 |
| DC LAMPUNG | Rp. 6.650.000.000 | Rp. 266.000.000 | 4% | Rp. 1.253.608.244 | Rp. 34.614.215 | 2.8% | Rp. -5.396.391.628 | 18.9% | Rp. -231.385.785 |
| DC LOMBOK | Rp. 12.700.000.000 | Rp. 508.000.000 | 4% | Rp. 4.319.829.747 | Rp. 73.290.215 | 1.7% | Rp. -8.380.170.509 | 34% | Rp. -434.709.785 |
| DC MAKASAR | Rp. 11.850.000.000 | Rp. 474.000.000 | 4% | Rp. 1.380.052.108 | Rp. 47.259.924 | 3.4% | Rp. -10.469.948.276 | 11.6% | Rp. -426.740.076 |
| DC MALANG | Rp. 19.987.500.000 | Rp. 799.500.000 | 4% | Rp. 6.934.321.398 | Rp. 208.554.523 | 3% | Rp. -13.053.177.610 | 34.7% | Rp. -590.945.509 |
| DC MANADO | Rp. 11.400.000.000 | Rp. 456.000.000 | 4% | Rp. 2.409.741.854 | Rp. 120.581.486 | 5% | Rp. -8.990.257.634 | 21.1% | Rp. -335.418.514 |
| DC MEDAN | Rp. 15.250.000.000 | Rp. 610.000.000 | 4% | Rp. 2.137.888.937 | Rp. 92.214.555 | 4.3% | Rp. -13.112.110.935 | 14% | Rp. -517.785.445 |
| DC PALEMBANG | Rp. 10.500.000.000 | Rp. 420.000.000 | 4% | Rp. 2.851.701.344 | Rp. 95.791.599 | 3.4% | Rp. -7.648.298.400 | 27.2% | Rp. -324.208.401 |
| DC PARUNG | Rp. 15.200.000.000 | Rp. 608.000.000 | 4% | Rp. 5.153.077.242 | Rp. 337.965.829 | 6.6% | Rp. -10.046.922.758 | 39.9% | Rp. -270.034.171 |
| DC PEKANBARU | Rp. 11.350.000.000 | Rp. 454.000.000 | 4% | Rp. 1.574.724.541 | Rp. 70.819.813 | 4.5% | Rp. -9.775.275.075 | 13.9% | Rp. -383.180.187 |
| DC PLUMBON | Rp. 13.106.800.000 | Rp. 524.273.000 | 4% | Rp. 5.281.169.764 | Rp. 166.384.935 | 3.2% | Rp. -7.825.650.332 | 40.3% | Rp. -357.887.865 |
| DC PONTIANAK | Rp. 9.550.000.000 | Rp. 382.000.000 | 4% | Rp. 3.062.230.842 | Rp. 90.349.438 | 3% | Rp. -6.487.769.286 | 32.1% | Rp. -291.650.562 |
| DC REMBANG | Rp. 20.650.000.000 | Rp. 826.000.000 | 4% | Rp. 3.721.965.584 | Rp. 118.229.050 | 3.2% | Rp. -16.928.034.800 | 18% | Rp. -707.770.950 |
| DC SEMARANG | Rp. 19.100.000.000 | Rp. 764.000.000 | 4% | Rp. 4.773.261.219 | Rp. 220.299.485 | 4.6% | Rp. -14.326.739.037 | 25% | Rp. -543.700.515 |
| DC SERANG | Rp. 13.950.000.000 | Rp. 558.000.000 | 4% | Rp. 4.662.229.486 | Rp. 150.002.787 | 3.2% | Rp. -9.287.770.642 | 33.4% | Rp. -407.997.213 |
| DC SIDOARJO | Rp. 15.525.000.000 | Rp. 621.000.000 | 4% | Rp. 4.021.809.109 | Rp. 254.677.194 | 6.3% | Rp. -11.503.191.083 | 25.9% | Rp. -366.322.806 |
| Grand Total | Rp. 420.256.800.000 | Rp. 16.810.273.000 | 4% | Rp. 133.489.233.838 | Rp. 5.351.924.590 | 4% | Rp. -286.768.566.162 | 31.8% | Rp. -11.458.748.410 |

Gambar 3.24 Hasil Reporting National Sales

3.2.2.4 Membuat Automail Untuk Menampilkan Report Terkait Informasi Transaksi Member

Untuk tugas yang berikutnya adalah untuk membuat *report* terkait informasi dari setiap transaksi yang dilakukan oleh *member*. Informasi yang nantinya akan ditampilkan merupakan hasil diskusi dari pihak divisi logistik, untuk keperluan rekap data yang ada. *Reporting* ini berguna untuk melihat bagaimana setiap *member* melakukan transaksi dengan melalui aplikasi atau transaksi melalui bantuan pihak MRO atau petugas. Serta juga untuk melihat *behavior* dari *member* yang berbelanja setiap kapan dan melihat *items* apa saja yang paling sering di beli atau di transaksikan.

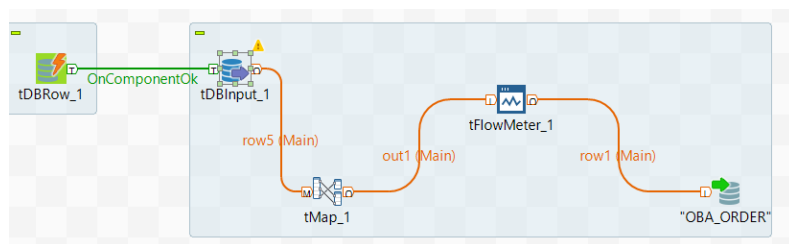
Untuk mendukung *reporting* ini, akan dibutuhkan data transaksi yang akan diambil dari *database Postgresql*. Namun, sebelum melakukan proses ETL, akan dibuat *table* terlebih

dahulu pada *database* BI dengan nama *table* “oba_order” yang dapat dilihat pada **gambar 3.23**.

```
CREATE TABLE `OBA_ORDER` (
  `TANGGAL_ORDER` timestamp NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE
  CURRENT_TIMESTAMP,
  `PACKED_AT` timestamp NULL DEFAULT NULL,
  `PICKING_TIME` int(11) DEFAULT NULL,
  `LOGIN_BY` varchar(30) DEFAULT NULL,
  `ID_OBA` varchar(22) DEFAULT NULL,
  `KODE_TOKO` varchar(12) DEFAULT NULL,
  `NAMA_TOKO` varchar(55) DEFAULT NULL,
  `BRANCH` varchar(55) DEFAULT NULL,
  `ORDER_ID` varchar(55) NOT NULL,
  `NO_SPL` varchar(55) NOT NULL,
  `PAYMENT_ID` varchar(55) NOT NULL,
  `PAYMENT` varchar(22) DEFAULT NULL,
  `PAYMENT_STATUS` varchar(55) DEFAULT NULL,
  `SETORAN` text,
  `STATUS_KIRIM_ORDER` text,
  `TANGGAL_KIRIM_KETOKO` timestamp NULL DEFAULT NULL,
  `TANGGAL_SAMPAI_DITOKO` timestamp NULL DEFAULT NULL,
  `TOTAL_ITEMS` int(4) DEFAULT NULL,
  `AMOUNT_PAID` decimal(10,0) DEFAULT NULL,
  `KODE_VOUCHER` varchar(22) DEFAULT NULL,
  `UPDATE_ORDER` timestamp NULL DEFAULT NULL,
  PRIMARY KEY (`NO_SPL`,`ORDER_ID`,`PAYMENT_ID`),
  KEY `OBA_ORDER_FK_1` (`ORDER_ID`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Gambar 3.25 Query 8

Setelah dibuatnya *table* tersebut, lalu akan dibuat *mapping* ETL pada *talend* yan dapat dilihat pada **gambar 3.24**



Gambar 3.26 Mapping ETL 2

Setelah dibuat proses ETL tersebut proses selanjutnya adalah melakukan *import* data pada excel. File excel ini digunakan untuk melihat data secara detail. Setelah data *mirror* yang telah dibuat sudah ada pada *table* tujuan, data tersebut masih disebut data mentah karena masih belum mengerucut untuk kebutuhannya. Langkah selanjutnya yaitu membuat *query* untuk keperluan data *detail* yang akan diisi pada excel. Bentuk untuk *report* yang sudah jadi dapat dilihat pada **gambar 3.25**.

| Summary Jenis Transaksi | | | | | | | | | | | | |
|-------------------------|----------------------------|----------|--------|-------------------------|-----------------|---------------------|--------------------------------|----------|--------|---------------------------|----------|--------|
| BRANCH | JUMLAH ORDER PER TRANSAKSI | | | NET SALES PER TRANSAKSI | | | % CONTR ORDER ID PER TRANSAKSI | | | % NET SALES PER TRANSAKSI | | |
| | AlfamikoApps | Whatsapp | Office | AlfamikoApps | Whatsapp | Office | AlfamikoApps | Whatsapp | Office | AlfamikoApps | Whatsapp | Office |
| Branch ID | 11699 | 28 | 11233 | Rp. 6.496.284.714 | Rp. 49.894.598 | Rp. 23.813.807.674 | 51% | 0.1% | 49% | 21% | 0.2% | 78% |
| B000 | 12531 | 77 | 21366 | Rp. 4.454.363.515 | Rp. 29.441.093 | Rp. 20.579.155.579 | 37% | 0.2% | 63% | 18% | 0.1% | 82% |
| B000 | 19475 | 58 | 14866 | Rp. 8.006.590.738 | Rp. 48.649.153 | Rp. 25.600.715.289 | 51% | 0.2% | 49% | 24% | 0.1% | 76% |
| B050 | 17732 | 93 | 16310 | Rp. 11.922.154.789 | Rp. 219.090.067 | Rp. 32.232.449.219 | 52% | 0.3% | 48% | 27% | 0.3% | 73% |
| Total | 57634 | 256 | 64755 | Rp. 30.876.493.747 | Rp. 347.074.911 | Rp. 102.225.937.761 | 47% | 0.2% | 53% | 23% | 0.3% | 77% |

| Summary Payment Method | | | | | | | | | | | | | | | | | | | | |
|------------------------|---------------------------------|---------|-----------|------|------------------------------|---------------------|-------------------|-------------------|-------------------------------------|-------------------|---------|---------|--------------------------------|------|--------|---------|---------|-----------|------|--------|
| BRANCH | JUMLAH ORDER PER PAYMENT METHOD | | | | NET SALES PER PAYMENT METHOD | | | | % CONTR ORDER ID PER PAYMENT METHOD | | | | % NET SALES PER PAYMENT METHOD | | | | | | | |
| | SAT-COD | SAT-COP | Tokomodal | VRGO | VA-BCA | SAT-COD | SAT-COP | Tokomodal | VRGO | VA-BCA | SAT-COD | SAT-COP | Tokomodal | VRGO | VA-BCA | SAT-COD | SAT-COP | Tokomodal | VRGO | VA-BCA |
| Branch ID | 19650 | 1240 | 2136 | 76 | 37 | Rp. 27.240.652.772 | Rp. 992.888.826 | Rp. 1.594.345.581 | Rp. 17.475.200 | Rp. 514.484.607 | 85% | 5.5% | 9% | 0.3% | 0.2% | 90% | 3% | 5% | 5% | 1.7% |
| B000 | 31331 | 504 | 1986 | 121 | 32 | Rp. 22.359.769.789 | Rp. 538.342.682 | Rp. 1.289.566.704 | Rp. 24.928.571 | Rp. 850.952.441 | 92% | 1.5% | 6% | 0.4% | 0.1% | 89% | 2% | 5% | 5% | 3.4% |
| B000 | 28613 | 1494 | 2022 | 234 | 36 | Rp. 26.178.555.130 | Rp. 2.319.159.761 | Rp. 1.717.264.433 | Rp. 46.509.453 | Rp. 394.466.403 | 88% | 4.5% | 7% | 0.8% | 0.1% | 87% | 7% | 5% | 5% | 3.2% |
| B050 | 29406 | 427 | 4038 | 183 | 81 | Rp. 37.659.491.292 | Rp. 567.734.761 | Rp. 2.888.304.661 | Rp. 32.466.296 | Rp. 3.235.807.526 | 86% | 1.2% | 12% | 0.5% | 0.2% | 85% | 1% | 7% | 7% | 7.3% |
| Total | 107000 | 3667 | 10102 | 618 | 189 | Rp. 116.828.828.856 | Rp. 6.819.116.836 | Rp. 7.489.381.399 | Rp. 102.776.536 | Rp. 4.985.710.977 | 89% | 3% | 8% | 0.5% | 0.2% | 87% | 3% | 6% | 3% | 6% |

Gambar 3.27 Hasil Reporting Informasi Transaksi Member

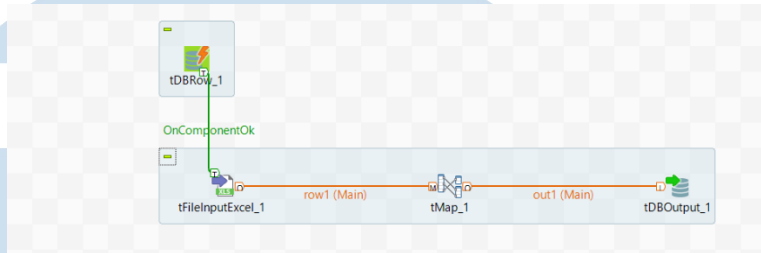
Dari hasil *dashboard* tersebut kita bisa melihat seberapa banyak *order* yang masuk, *sales* yang terbentuk dan juga *order* yang terbentuk berdasarkan transaksi mandiri atau transaksi *non* mandiri dari tanggal satu sampai tanggal berjalan dalam periode satu bulan.

3.2.2.5 Membuat Automail Untuk Menampilkan Report Kinerja Setiap Toko

Tugas terakhir yang diberikan adalah pembuatan *report* untuk kinerja setiap toko yang ada pada cabang PT Sumber Trijaya Lestari. Tugas ini merupakan permintaan dari divisi *operations* yang paling sering berinteraksi dengan petugas – petugas aplikasi Aksesmu atau MRO. Dalam *report* ini berisi hal yang memuat sebuah *dashboard* yang berisikan poin kinerja setiap toko yang ada dibawah PT STL. Poin tersebut mencakup nilai *sales*, akuisisi member pebulan, *margin*.

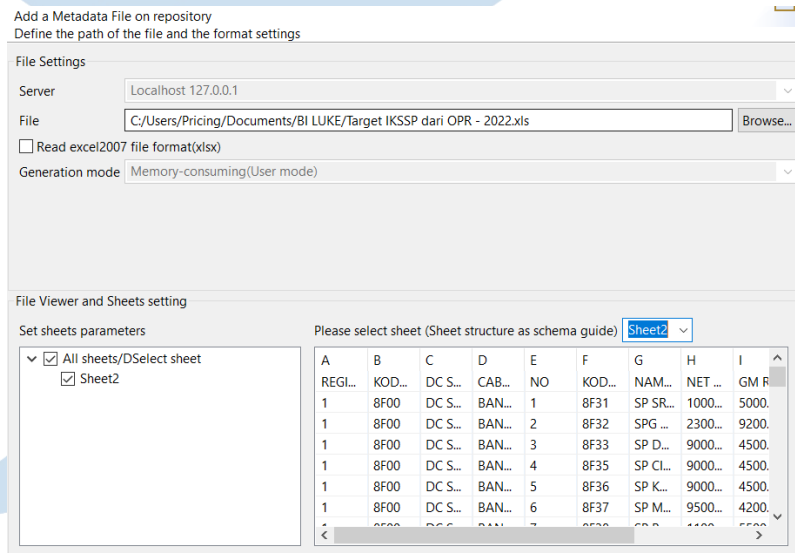
Untuk tugas pertama yang harus dilakukan adalah mengumpulkan data apa saja yang diperlukan untuk membuat *report* tersebut. Data pertama yang diperlukan merupakan target dari setiap toko yang ada, dikarenakan target bersifat dinamis dan belum diintegrasikan ke dalam *database* milik IT PT STL sehingga didapatkan data tersebut berupa excel yang harus di *import* ke dalam *table database* “Alfamikro_BI”.

Untuk langkah awal dalam proses import data excel ke *table* adalah membuat *mapping* ETL tersebut pada aplikasi *Talend* yang prosesnya dapat dilihat pada **gambar 3.26**



Gambar 3.28 Mapping ETL 3

Pada gambar *mapping* dapat dilihat bahwa yang terjadi adalah proses *input* melalui file excel kemudian tipe data di *convert* yang otomatis menyesuaikan tipe data MYSQL. Setelah membuat *mapping*, langkah kedua kemudian import file excel pada atribut *tfileinputexcel* yang prosesnya dapat dilihat pada **gambar 3.27**



Gambar 3.29 Importing Excel

Pada gambar diatas didapatkan informasi, melakukan *import* file excel yang berasal dari directory local. Dikarenakan table yang menjadi sasaran belum tersedia sehingga perlu dibuat *table* tersebut terlebih dahulu yang bernama “TARGET_IKSSP”.

Proses pembuatan *table* dapat dilihat melalui **gambar 3.28**

```

SQL Preview:
1 -- bi_stl.TARGET_IKSSP definition
2
3 CREATE TABLE `TARGET_IKSSP` (
4   `REGIONAL` int(11) DEFAULT NULL,
5   `KODE_DC` varchar(100) DEFAULT NULL,
6   `DC_STL` varchar(100) DEFAULT NULL,
7   `CABANG` varchar(100) DEFAULT NULL,
8   `NO` int(11) DEFAULT NULL,
9   `KODE_TOKO` varchar(100) DEFAULT NULL,
10  `NAMA_TOKO` varchar(100) DEFAULT NULL,
11  `NET_SALES` decimal(10,0) DEFAULT NULL,
12  `GM_RP` int(11) DEFAULT NULL,
13  `MEMBER_AKTIF` int(11) DEFAULT NULL,
14  `SALES_TAG` int(11) DEFAULT NULL,
15  `PERIODE` date DEFAULT NULL
16 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

Gambar 3.30 Query 9

Setelah *table* sudah dibuat kemudian langkah terakhir yaitu untuk menjalankan proses ETL pada talend dan kemudian data berhasil diimport pada *table*. Kemudian untuk pembuatan *dashboard report* kinerja setiap toko dibutuhkan *query* yang memerlukan *JOIN* karena harus mengambil informasi *sales* dan juga informasi *member*. Untuk proses pembuatan *query*nya dapat dilihat melalui **gambar 3.29**

```

SELECT
wss.STORE_ID,
wss.STORE_NAME,
wss.KET,
wss.OLD_BRANCH BRANCH,
SUM(mmps.NET_SALES) NET_SALES,
SUM(mmps.GM) GM,
ma.M_AKUISISI NOO,
ft.M_TRS PERTAMA,
'' FOLLOW_MEDIA,
psale.NET_SALES TARGET_NS,
psale.GM_RP TARGET_GM,
'150' TARGET_NOO,
'120' TARGET_M_TRS PERTAMA,
'' TARGET_FOLLOW_MEDIA,
'' POIN_NET_SALES,
'' POIN_GM,
ROUND((ma.M_AKUISISI/150)*50,2) POIN_NOO,
ROUND((ft.M_TRS PERTAMA/120)*20,2) POIN_M_TRS PERTAMA,
'' POIN_FOLLOW_MEDIA
FROM stl_bc.MST_MEMBER_PLU_SALES_mmps
LEFT JOIN stl_bc.WEC_STORE_STL_wss
on wss.STORE_ID = mmps.KODE_TOKO
LEFT JOIN(SELECT
t.storeCode,
COUNT(obaId) M_AKUISISI
from bi_stl.tb_ama_login_auth_user_ t
LEFT join bi_stl.tb_member tm
on tm.oba_id = t.obaid
WHERE YEAR(registeredAt) = YEAR(CURDATE())
AND MONTH(registeredAt) = MONTH(CURDATE())
AND companyId IN ('STL')
AND tm.status in ('approved','approved-partial','pending') and t.isAccountActive = 1
GROUP BY t.storeCode
)ma ON ma.storeCode = mmps.KODE_TOKO
LEFT JOIN(SELECT
a.KODE_TOKO,
COUNT(a.MEMBERS) M_TRS PERTAMA
FROM(
SELECT

```

Gambar 3.31 Query 10

Untuk penampilan *dashboard* yang telah dibuat dapat dilihat pada **gambar 3.30**

| PERIODE | NS | GM | %GM | TOTAL TOKO SP | SPD | AVG STOCK | DSI | PERIODE | JML. Member | JML. Member Akuisisi | JML. Akuisisi Mandiri | JML. Akuisisi |
|---------|---------------------|-------------------|------|---------------|----------------|-------------------|-----|---------|-------------|----------------------|-----------------------|---------------|
| TGL 1 | Rp. 238.814.331 | Rp. 8.108.442 | 3.4% | 269 | Rp. 15.920.955 | Rp. 798.730.534 | 52 | TGL 1 | 157473 | 46 | 26 | |
| TGL 2 | Rp. 12.960.130.400 | Rp. 383.896.867 | 3.0% | 269 | Rp. 48.539.814 | Rp. 60.581.036 | 1 | TGL 2 | 157953 | 472 | 249 | |
| TGL 3 | Rp. 10.364.211.120 | Rp. 316.731.330 | 3.1% | 270 | Rp. 38.672.430 | Rp. 672.766.632 | 18 | TGL 3 | 158460 | 514 | 269 | |
| TGL 4 | Rp. 12.282.865.651 | Rp. 377.648.614 | 3.1% | 270 | Rp. 47.241.791 | Rp. 669.538.485 | 15 | TGL 4 | 158918 | 466 | 259 | |
| TGL 5 | Rp. 410.773.842 | Rp. 4.686.179 | 1.1% | 270 | Rp. 41.077.384 | Rp. 813.477.766 | 20 | TGL 5 | 158931 | 30 | 22 | |
| TGL 6 | Rp. 9.006.386.403 | Rp. 290.188.687 | 3.2% | 270 | Rp. 33.605.919 | Rp. 0 | | TGL 6 | 159112 | 525 | 258 | |
| TGL 7 | Rp. 7.597.176.848 | Rp. 349.172.843 | 4.6% | 270 | Rp. 28.347.675 | Rp. 864.697.024 | 32 | TGL 7 | 159719 | 598 | 338 | |
| TGL 8 | Rp. 6.639.468.250 | Rp. 278.879.070 | 4.2% | 270 | Rp. 25.149.501 | Rp. 902.403.956 | 37 | TGL 8 | 160183 | 556 | 278 | |
| TGL 9 | Rp. 6.183.404.677 | Rp. 252.015.447 | 4.1% | 271 | Rp. 23.421.987 | Rp. 889.366.799 | 40 | TGL 9 | 160483 | 616 | 333 | |
| TGL 10 | Rp. 6.929.565.080 | Rp. 325.258.028 | 4.7% | 271 | Rp. 25.760.465 | Rp. 912.209.684 | 37 | TGL 10 | 161055 | 563 | 309 | |
| TGL 11 | Rp. 7.997.217.371 | Rp. 286.681.924 | 3.6% | 271 | Rp. 29.729.433 | Rp. 930.404.515 | 32 | TGL 11 | 161597 | 532 | 258 | |
| TGL 12 | Rp. 7.522.051.157 | Rp. 318.433.347 | 4.2% | 272 | Rp. 27.859.449 | Rp. 0 | | TGL 12 | 161613 | 25 | 17 | |
| TGL 13 | Rp. 7.047.609.915 | Rp. 247.067.934 | 3.5% | 272 | Rp. 26.102.259 | Rp. 994.662.794 | 39 | TGL 13 | 162236 | 622 | 314 | |
| TGL 14 | Rp. 7.087.043.345 | Rp. 318.064.923 | 4.5% | 272 | Rp. 26.248.309 | Rp. 982.896.439 | 39 | TGL 14 | 162890 | 658 | 342 | |
| TGL 15 | Rp. 6.966.917.874 | Rp. 358.522.539 | 5.1% | 273 | Rp. 26.795.838 | Rp. 997.582.420 | 39 | TGL 15 | 163496 | 596 | 329 | |
| TGL 16 | Rp. 7.073.275.489 | Rp. 412.918.013 | 5.8% | 273 | Rp. 26.100.648 | Rp. 973.133.399 | 40 | TGL 16 | 164043 | 548 | 270 | |
| TGL 17 | Rp. 8.005.936.631 | Rp. 363.960.610 | 4.5% | 274 | Rp. 29.984.781 | Rp. 1.051.975.868 | 37 | TGL 17 | 164555 | 533 | 288 | |
| TGL 18 | Rp. 9.297.545.847 | Rp. 466.921.587 | 5.0% | 274 | Rp. 34.182.154 | Rp. 0 | | TGL 18 | 165025 | 458 | 234 | |
| TGL 19 | Rp. 9.704.510.659 | Rp. 537.595.623 | 5.5% | 274 | Rp. 35.810.002 | Rp. 1.166.215.087 | 34 | TGL 19 | 165029 | 24 | 22 | |
| TGL 20 | Rp. 143.314.904.890 | Rp. 5.896.752.007 | 4.1% | | Rp. 31.339.363 | | | TGL 20 | 165686 | 672 | 373 | |
| TGL 21 | | | | | | | | TGL 21 | 165659 | 659 | 345 | |

Gambar 3.32 Hasil Reporting Kinerja Setiap Toko

3.3 Kendala yang Ditemukan

Berikut merupakan kendala yang ditemukan saat melakukan program kerja magang pada PT Sumber Trijaya Lestari :

- Kurangnya bimbingan dari atasan, karena tidak memiliki *background* dan pengalaman di bidang IT.
- Ketidapahaman atas *database* yang dimiliki oleh perusahaan.
- Kurangnya penjelasan tentang *report* yang akan dibuat.

3.4 Solusi Atas Kendala yang Ditemukan

Berikut merupakan solusi yang menjawab kendala saat melakukan program kerja magang pada PT Sumber Trijaya Lestari :

- Keharusan untuk melakukan eksplorasi tentang sistem dan *tools* yang digunakan dalam perusahaan.
- Melakukan pembelajaran mandiri dengan bertanya langsung dengan departemen IT.
- Bertanya lebih detail terkait *report* yang akan dibuat.

UNIVERSITAS
MULTIMEDIA
NUSANTARA