



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB III PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

Dalam melaksanakan kerja magang, penulis diposisikan sebagai *Database Administrator* di tim yang sedang menjalankan pembuatan sebuah proyek yang cukup besar. Proyek tersebut adalah proyek Berleha dimana proyek ini berfokus kepada model B2B. Keluaran dari proyek ini adalah sebuah aplikasi untuk penjualan tiket pesawat kepada sub-agen dengan harga yang bersaing jika dibandingkan dengan para pesaingnya. Penulis mendapat bimbingan dan koordinasi dari Bapak Eric Elkana Tarigan, selaku *Project Manager* dan Bapak Halim selaku *Database Administrator* yang dikirim untuk membantu dalam proyek ini.

Tugas utama dari seorang *Database Administrator* adalah membuat dan memastikan sebuah *database* yang digunakan dapat berjalan dengan baik dan sesuai dengan fungsi yang seharusnya. Selain itu sebagai seorang *Database Administrator* juga harus dapat mengoptimalkan sebuah *server* dari *database* tersebut agar dapat berjalan dengan ringan dan memastikan data yang masuk dan keluar benar adanya sesuai dengan keinginan dan kebutuhan dari user itu sendiri.

3.2 Tugas yang Dilakukan

Tugas-tugas yang dilakukan oleh penulis selama melakukan kerja magang di PT Perkasa Pilar Utama sebagai *Database Administrator* adalah:

1. Pengenalan Proses Bisnis

Penulis mempelajari proses bisnis yang digunakan dalam proyek ini. Dengan mempelajari proses bisnis ini, akhirnya penulis tahu apa yang ingin dicapai dan kebutuhan *database* seperti apa yang ingin dicapai.

2. Pengenalan *Schema Database*

Schema dari sebuah *database* sangat penting untuk dipelajari karena dalam sebuah *database*, setiap *table* dapat saling berhubungan satu dengan lainnya.

3. Latihan Pembuatan *Procedure* dan *Job*

Procedure sendiri adalah sebuah *script* dari awal hingga akhir yang disusun menjadi satu yang dibuat guna mencapai suatu tujuan (contoh: *procedur* untuk melakukan *update* status pesanan). Sedangkan *job* sendiri adalah sebuah *scheduler* dimana kita dapat menentukan sebuah *procedure* kita kapan akan berjalan dan kapan akan berhenti secara otomatis.

4. Pengenalan Jenis-Jenis Partisi di Oracle

Karena adanya permintaan agar *database* dapat berjalan dengan cepat, maka penulis dan Bapak Helmi mencoba mendalami jenis partisi yang mana yang pas dan dapat diterapkan dalam proyek ini.

5. Pembuatan *Partitioning Interval Table Master*

Pembuatan *Partitioning Interval* pada *Table Master* sendiri bertujuan untuk memenuhi permintaan dari *user* agar *database* berjalan dengan cepat dan data H-2 dari hari ini sudah tidak dibaca kembali.

6. Pembuatan *Procedure* dan *Job Update Status Expired*

Penulis membuat *script* ini yang nantinya dapat digunakan dalam beberapa *table* seperti *table* pemesanan tiket maupun untuk pengisian saldo.

7. Pembuatan *Script* Melakukan *Check* Terhadap *Double Booking*

Di sini penulis diminta langsung oleh *Project Manager* untuk membuat sebuah *query/script* yang nantinya akan dimasukkan kedalam aplikasi Berleha supaya tidak terjadi *double booking*.

8. Pembuatan *Table Master* Komisi dan Melakukan Kontrol Inputan

Disini penulis melakukan *design table database* untuk *table master* komisi supaya pada saat komisi berubah dapat dilihat dan diperbaiki dengan mudah.

9. Menghitung Besaran Komisi dan Menginput ke Dalam Table Komisi

Penulis melakukan *check* besaran komisi yang diberikan oleh setiap maskapai yang bekerja sama dengan kita dan memasukkannya ke dalam *database* untuk setiap route penerbangannya.

10. Pembuatan *Query* Insert Seluruh Data

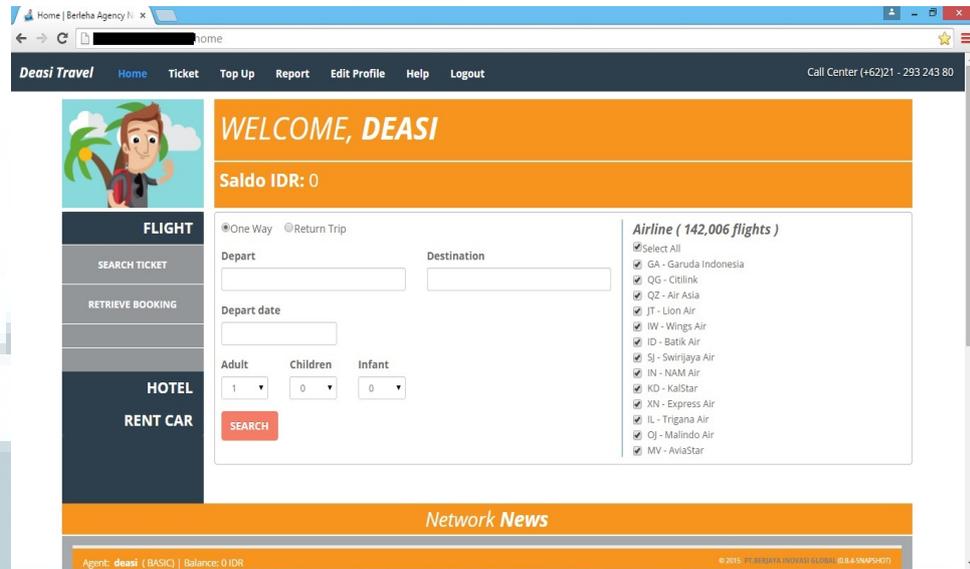
Penulis melakukan export seluruh isi data dalam database tersebut dan menjadikannya sebuah *query insert* guna untuk melakukan pencadangan data maupun untuk melakukan *refresh* data ketika melakukan UAT.

11. Dokumentasi Database

Penulis membuat dokumentasi untuk databasenya tersebut berupa seluruh *query* yang ada di databasenya guna untuk dianalisa dan menjaga – jaga jika sewaktu – waktu *database*-nya terjadi *crash*.

3.3 Uraian Pelaksanaan Kerja Magang

3.3.1 Pengenalan Proses Bisnis



Gambar 3.3.1.1 Tampilan Awal Aplikasi Berleha

Pekerjaan pertama yang penulis lakukan adalah mempelajari proses bisnis yang digunakan saat ini. Proyek Berleha ini awalnya adalah sebuah bisnis yang bergerak dalam bidang B2C (*Business to Customer*). Dalam proyek aplikasi untuk berleha sendiri diharapkan dapat memenuhi kebutuhan setiap masyarakat yang ingin melakukan *traveling* ke mana pun. Harapan dan rencana awal dari proyek ini adalah terdapat sebuah tempat (Berleha) dimana setiap orang dapat memesan tiket pesawat sekaligus memesan hotel maupun kendaraan seperti taxi.

Setelah sampai pada setengah perjalanan pembuatan aplikasi, pihak *management* memutuskan untuk merubah arah menjadi B2B (*Business to Business*) dimana kita membuka jasa untuk orang – orang dapat bergabung menjadi sub-agen kita.

Dengan model bisnis ini diharapkan pendapatan yang diperoleh menjadi lebih baik jika dibandingkan dengan model B2C. Akan tetapi tidak tertutup kemungkinan bahwa suatu saat model B2C-nya akan dijalankan juga agar sesuai dengan namanya Berleha. Pengambilan nama Berleha sendiri memiliki tujuan yaitu agar orang dapat berleha – leha / santai ketika ingin melakukan *travel* ke mana pun karena di dalam satu aplikasi ini menyediakan seluruh kebutuhan dari pemesanan tiket pesawat, tiket hotel hingga untuk transportasi seperti penyewaan mobil.

3.3.2 Pengenalan *Schema Database*

Table	Schema	Tablespace	Last Analyzed	Num Rows	Created	Last DDL
ACC_TRX_DTL	WVR	WVR_TABLE1	9/28/2015 12:25:51 AM	7	4/14/2015 3:07:12 PM	9/28/2015
ACC_TRX_DTL_AUD	WVR	WVR_TABLE1	9/28/2015 12:25:51 AM	6	17/2015 5:02:45 PM	8/27/2015
AGENT_PACKAGE	WVR	WVR_TABLE1	9/28/2015 12:25:52 AM	11	17/2014 3:46:44 PM	7/23/2015
AGENT_TRX	WVR	WVR_TABLE1	9/28/2015 12:25:52 AM	5	1/5/2015 6:02:11 PM	7/23/2015
AGENT_TRX_USD	WVR	WVR_TABLE1	9/28/2015 12:25:52 AM	6	13/2015 6:26:41 PM	7/23/2015
APP_USER	WVR	WVR_TABLE1	9/28/2015 12:25:52 AM	16	9/30/2014 5:05:28 PM	7/23/2015
APP_USER_AUD	WVR	WVR_TABLE1	9/28/2015 12:25:52 AM	4	6/13/2015 11:51:39 AM	7/23/2015
BACKUP_FAS_CLEAN	WVR	WVR_TABLE1	8/4/2015 6:35:44 AM	113,169	3/31/2015 1:11:10 PM	5/5/2015
BANKACCOUNT	WVR	WVR_TABLE1	8/4/2015 6:35:44 AM	10	28/2014 2:31:51 PM	10/28/2015
BOOKING_FLIGHT	WVR	WVR_TABLE1	9/28/2015 12:25:53 AM	1,433	1/5/2015 6:00:28 PM	7/23/2015
BOOKING_FLIGHT_DETAIL	WVR	WVR_TABLE1	9/28/2015 12:25:53 AM	2,567	1/6/2015 11:48:56 AM	7/14/2015
BOOKING_FLIGHT_FELIX	WVR	WVR_TABLE1	8/4/2015 6:35:45 AM	7	13/2015 9:38:15 AM	7/13/2015
BOOKING_FLIGHT_SCHEDULE	WVR	WVR_TABLE1	9/28/2015 12:25:54 AM	2,198	1/16/2015 9:01:09 AM	8/13/2015
BOOKING_FLIGHT_TEST	WVR	WVR_TABLE1	8/4/2015 6:35:45 AM	639	5/21/2015 9:46:31 AM	5/26/2015
BOOKING_FLIGHT_TRANS	WVR	WVR_TABLE1	9/28/2015 12:25:54 AM	1,137	4/28/2015 10:58:40 AM	5/26/2015
BOOKING_HOTEL	WVR	WVR_TABLE1	9/28/2015 12:25:54 AM	5	18/2015 12:07:41 PM	7/23/2015
CIMBIVANCOUNTER	WVR	WVR_TABLE1	9/28/2015 12:25:54 AM	11	11/26/2014 6:29:35 AM	11/26/2015
COMMISSION	WVR	WVR_TABLE1	8/4/2015 6:35:45 AM	13	4/8/2015 6:18:11 PM	4/28/2015
CUSTOMER	WVR	WVR_TABLE1	8/4/2015 6:35:45 AM	9	30/2014 5:05:28 PM	7/23/2015
CUSTOMERS	WVR	WVR_TABLE1	8/4/2015 6:35:45 AM	8	8/2014 4:11:17 PM	8/8/2014
CUSTOMREVINFOENTITY	WVR	WVR_TABLE1	9/28/2015 12:25:51 AM	7	5/18/2015 11:46:11 AM	7/23/2015
DBUSER	WVR	WVR_TABLE1	8/4/2015 6:35:45 AM	2	10/30/2014 1:42:47 PM	7/23/2015
EMAIL	WVR	WVR_TABLE1	9/28/2015 12:25:54 AM	17	3/16/2015 1:58:49 PM	3/16/2015
ERR_LOGS	WVR	WVR_TABLE1	8/4/2015 6:35:46 AM	7	24/2015 4:56:47 AM	7/24/2015
ERROR_LOG	WVR	WVR_TABLE1	9/28/2015 12:25:55 AM	446	5/25/2015 3:32:00 PM	5/25/2015
FCF_KALSTAR_DEAST	WVR	WVR_TABLE1	8/4/2015 6:35:46 AM	17,291	3/6/2015 2:56:54 PM	3/6/2015

Gambar 3.3.2.1 *Schema Database* Berleha

Pekerjaan berikutnya yang penulis terima adalah pengenalan *schema database* yang digunakan pada proyek Berleha ini. Dapat kita lihat

bahwa dalam *schema* ini memiliki banyak sekali table yang harus dikelola secara baik. Tetapi selama penulis melakukan kerja magang, hanya ada beberapa *table* yang menjadi fokus utama, yaitu *table* BOOKING_FLIGHT, BOOKING_FLIGHT_DETAIL, FLIGHT_CLASS_FARE, FLIGHT_CLASS_FARE_AIRASIA dan seluruh table yang mengandung kata FLIGHT_CLASS_FARE.

Table – table tersebut menjadi fokus kita pada *Database Administrator* dikarenakan table – table tersebut setiap detik akan bertambah datanya secara signifikan sehingga dikawatirkan dapat memperlambat kinerja *server* dari Berleha ini sendiri.

Selain adanya *schema* utama ini yaitu VWR, kita juga menggunakan sebuah *schema* lagi yaitu BIG_TRANS. Dalam *schema* ini mengatur jumlah besaran komisi yang diterima oleh setiap sub-agen yang mendaftar di Berleha ini dan berhasil menjual tiket.

3.3.3 Latihan Pembuatan *Procedure* dan *Job*

Penulis diharuskan mengerti dan dapat membuat sebuah *procedure* dan *job* yang kemudian dijalankan di *server*. Penulis melakukan pembelajaran secara *online* melalui situs resmi dari Oracle. Setelah mengerti dan memahami konsep dasar dari *procedure* dan *job*, barulah penulis mencoba untuk menjalankannya di *server* dan berhasil berjalan. Berikut adalah hasil uji coba pembuatan *procedure* dan *job* yang penulis buat.

```

=====
Membuat Procedure
=====
CREATE OR REPLACE PROCEDURE INSERT_PRO
IS code int;
BEGIN
insert into c values (1,sysdate);
END INSERT_PRO;
/

=====
Menjalankan Procedure
=====
exec insert_pro

=====
Membuat JOB
=====
BEGIN
SYS.DBMS_SCHEDULER.CREATE_JOB
(
  job_name      => 'SYSTEST2'
, start_date   => TO_TIMESTAMP_TZ('2015/07/06 21:48:00.000000 +07:00','yyyy/mm/dd hh24:mi:ss.ff tztz')
, repeat_interval => 'FREQ=MINUTELY;INTERVAL=1'
, end_date     => NULL
, job_class    => 'DEFAULT_JOB_CLASS'
, job_type     => 'STORED_PROCEDURE'
, job_action   => 'INSERT_PRO'
, comments     => NULL
);
END;
/

=====
Menjalankan JOB
=====
BEGIN
SYS.DBMS_SCHEDULER.ENABLE
(name => 'SYSTEST2');
END;
/

```

Gambar 3.3.3.1 Latihan Pembuatan Procedure dan Job

3.3.4 Pengenalan Jenis – Jenis Partisi di Oracle

Bapak Eric meminta kepada penulis untuk dibuatkan sebuah *script* yang dapat menghapus data H-2 sejak hari ini dari table FLIGHT_CLASS_FARE secara berkala dengan tujuan *server* tidak menjadi berat. Setelah berdiskusi cukup panjang, pada akhirnya kita semua memutuskan untuk tidak menghapus satu data pun yang ada di database Oracle karena dapat membuatnya menjadi tidak stabil.

Jalan keluar yang kita pilih adalah dengan melakukan partisi pada *table* FLIGHT_CLASS_FARE. Partisi ini bertujuan untuk memisahkan

secara otomatis data yang sudah tidak digunakan agar tidak muncul ketika terjadi proses *select* di *database*.

Setelah penulis melakukan pembelajaran terhadap beberapa jenis partisi yang disediakan oleh Oracle, pada akhirnya penulis dan tim database memilih menggunakan *Partitioning Interval* berdasarkan waktu, sangat memenuhi kebutuhan Bapak Eric.

3.3.5 Pembuatan Partitioning Interval Table Master

Gambar 3.3.5.1 di bawah ini adalah sebuah potongan sebuah potongan *query* yang membuat sebuah *table* menjadi terpartisi secara otomatis berdasarkan waktu yang telah ditentukan. Penulis membuat satu buah *table* dapat melakukan partisi secara otomatis selama lima tahun terhitung sejak tahun ini.

Seluruh data yang masih berada di tahun 2015 akan secara otomatis masuk ke dalam sebuah *tablespace* yang bernama FLIGH_CLASS_FARE_2015, untuk data di tahun 2016 akan masuk ke dalam *tablespace* FLIGHT_CLASS_FARE_2016 dan begitu seterusnya hingga di tahun ke lima yaitu tahun 2020.

Dengan adanya partisi ini data yang tampil ketika kita melakukan fungsi *select* akan tersusun secara rapi berdasarkan tahunnya. Contohnya jika kita ingin melihat data – data penerbangan di tahun 2015, maka kita tinggal memilih data di *tablespace* yang tahun 2015 dan secara otomatis data di tahun – tahun yang lain tidak akan tampil. Dengan begini harapan

untuk membuat beban server menjadi lebih ringan berhasil dan data semuanya tersimpan.

```
PARTITION BY RANGE (DEP_TIME)
(
  PARTITION FCF_AIRASIA_P2015 VALUES LESS THAN (TO_DATE('01-01-2016', 'DD-MM-YYYY'))
  NOLOGGING
  NOCOMPRESS
  TABLESPACE FLIGHT_CLASS_FARE_2015
  PCTUSED 40
  PCTFREE 10
  INITRANS 1
  MAXTRANS 255
  STORAGE
  (
    INITIAL 8M
    NEXT 1M
    MAXSIZE UNLIMITED
    MINEXTENTS 1
    MAXEXTENTS UNLIMITED
    FREELISTS 1
    FREELIST GROUPS 1
    BUFFER_POOL DEFAULT
    FLASH_CACHE DEFAULT
    CELL_FLASH_CACHE DEFAULT
  ),
  PARTITION FCF_AIRASIA_P2016 VALUES LESS THAN (TO_DATE('01-01-2017', 'DD-MM-YYYY'))
  NOLOGGING
  NOCOMPRESS
  TABLESPACE FLIGHT_CLASS_FARE_2016
  PCTUSED 40
  PCTFREE 10
  INITRANS 1
)
```

Gambar 3.3.5.1 Script Pembuatan Partisi Interval

Dapat kita lihat pada potongan *query* tersebut setiap data yang berada di bawah tahun yang telah kita tentukan akan masuk ke *tablespace* yang telah kita buat sebelumnya. Jumlah ingin seberapa banyak tahun yang kita buat tergantung dari permintaan (tidak dibatasi oleh Oracle).

3.3.6 Pembuatan *Procedure* dan *Job Update Status Expired*

Setelah *table* menggunakan partisi secara tahunan, Bapak Eric merasa masih kurang sempit dalam melakukan seleksi waktunya. Setelah dirundingkan beberapa saat, solusi yang diambil adalah adanya sebuah

kolom baru yaitu STATUS_PARTITION dimana di dalamnya mengandung satu buah karakter yaitu O (*expired*) atau N (aktif).

Penulis diminta untuk membuat sebuah *procedure* untuk melakukan *update* dari *status* tersebut berdasarkan hari maupun penerbangan yang sudah dilaksanakan dan *procedure* tersebut dijalankan secara otomatis secara harian dengan ketentuan seluruh penerbangan yang telah berhasil tetapi telah lewat dua hari sejak hari ini dan masih berstatus N, secara otomatis statusnya akan berubah menjadi O.

```
CREATE OR REPLACE PROCEDURE UPDATE_SPAR_SRIWIJAYA
IS code int;
BEGIN
update FLIGHT_CLASS_FARE_SRIWIJAYA set STATUS_PARTITION = 'O' where DEP_TIME < (SYSDATE-2) and STATUS_PARTITION
='N';
commit;
END UPDATE_SPAR_SRIWIJAYA;
/

BEGIN
SYS.DBMS_SCHEDULER.CREATE_JOB
(
  job_name      => 'UPDATE_STATUS'
  ,start_date   => TO_TIMESTAMP_TZ('2015/07/07 00:01:00.000000 +07:00','yyyy/mm/dd hh24:mi:ss.ff tzz')
  ,repeat_interval => 'FREQ=DAILY;INTERVAL=1'
  ,end_date     => NULL
  ,job_class    => 'DEFAULT_JOB_CLASS'
  ,job_type     => 'STORED_PROCEDURE'
  ,job_action   => 'UPDATE_SPAR_SRIWIJAYA'
  ,comments    => NULL
);
END;
/

BEGIN
SYS.DBMS_SCHEDULER.ENABLE
(name => 'UPDATE_STATUS');
END;
/
```

Gambar 3.3.6.1 Script Procedure Update Status Partisi

Setelah semua persyaratan tersebut dibuat, akhirnya penulis buat dan dijalankan di *server*. Hasil dari script tersebut akan menjadi seperti gambar di bawah ini.

FLIGHT_CLASS_FARE_SRIWIJAYA: Created: 7/28/2015 1:11:47 PM Last DDL: 7/28/2015 3:46:41 PM
Primary Key: CLASS_FARE_ID

CLASS_SEQUENCE	LASTUPDATE	CHD_FARE	PSC	INF_FARE	DEP_TIME	STATUS_PARTITION
VVWNUZXUT09	11 4/23/2015 6:52:01 PM	877	40000	76	8/1/2015 2:20:00 PM	O
VVWNUZXUT09	12 4/23/2015 6:52:01 PM	910	40000	79	8/1/2015 2:20:00 PM	O
VVWNUZXUT09	13 4/23/2015 6:52:01 PM	943	40000	82	8/1/2015 2:20:00 PM	O
VVWNUZXUT09	14 4/23/2015 6:52:01 PM	972	40000	84	8/1/2015 2:20:00 PM	O
	15 4/23/2015 6:52:01 PM	0	40000	84	8/1/2015 2:20:00 PM	O
	16 4/23/2015 6:52:01 PM	0	40000	84	8/1/2015 2:20:00 PM	O
	1 4/23/2015 6:52:01 PM	0	40000	84	8/1/2015 2:20:00 PM	O
	0 5/12/2015 2:16:56 PM	0	40000	86	8/1/2015 2:30:00 PM	N
	17 5/12/2015 2:16:56 PM	0	40000	86	8/1/2015 2:30:00 PM	N
VVWNUZXUT09	2 5/12/2015 2:16:56 PM	518	40000	49	8/1/2015 2:30:00 PM	N
VVWNUZXUT09	3 5/12/2015 2:16:56 PM	562	40000	47	8/1/2015 2:30:00 PM	N
VVWNUZXUT09	4 5/12/2015 2:16:56 PM	606	40000	51	8/1/2015 2:30:00 PM	N
VVWNUZXUT09	5 5/12/2015 2:16:56 PM	650	40000	55	8/1/2015 2:30:00 PM	N
VVWNUZXUT09	6 5/12/2015 2:16:56 PM	694	40000	59	8/1/2015 2:30:00 PM	N
VVWNUZXUT09	7 5/12/2015 2:16:56 PM	738	40000	63	8/1/2015 2:30:00 PM	N
VVWNUZXUT09	8 5/12/2015 2:16:56 PM	782	40000	67	8/1/2015 2:30:00 PM	N

Row 5988 of 6500 fetched so far (more rows exist)

Gambar 3.3.6.2 Hasil Query Update Status Partisi

Dapat kita lihat perbedaan antara data yang masih berstatus O dan N. Semua data yang berstatus O akan secara otomatis tidak muncul saat proses *select* berjalan kecuali dibutuhkan penarikan data khusus jika terjadi suatu masalah dikemudian harinya.

3.3.7 Pembuatan Query Melakukan Check Terhadap Double Booking

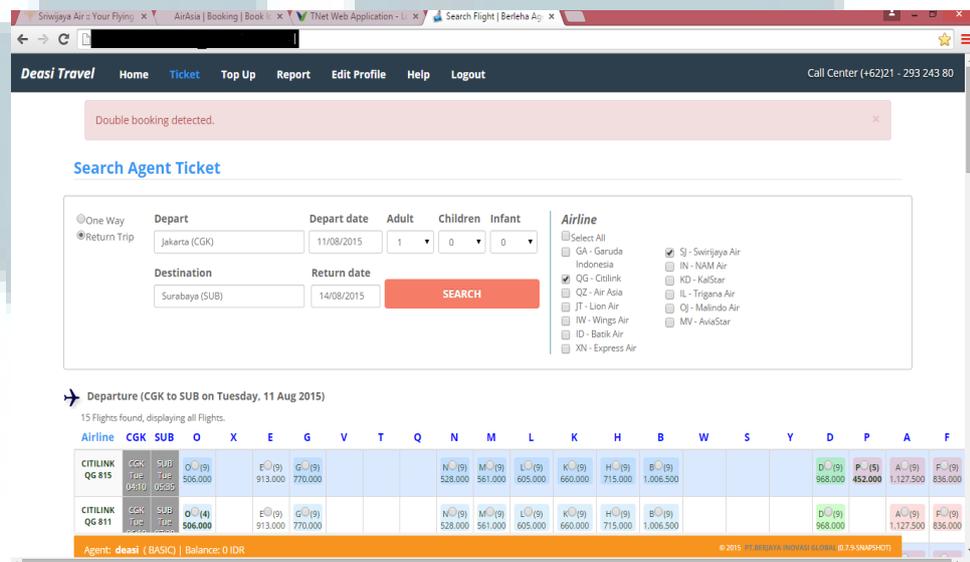
Terdapat dua buah masalah utama dalam bisnis travel seperti ini. Yang pertama adalah adanya kesalahan harga yang tampil ke publik baik terlalu rendah atau terlampau tinggi. Dan masalah yang kedua adalah terjadinya *double booking*. *Double booking* sendiri adalah suatu keadaan dimana seseorang memesan sebuah tiket dengan data diri yang sama untuk jalur penerbangan yang berbeda di waktu yang bersamaan.

Untuk mencegah hal itu terjadi, penulis membuat sebuah *script* yang dijalankan secara otomatis oleh aplikasi yang selalu melakukan pencegahan ketika seseorang secara tidak sengaja maupun disengaja melakukan *double booking* tersebut.

```
ALTER TABLE nama_table ADD CONSTRAINT nama_constraint UNIQUE (nama_column1, nama_column2, nama_column3);
```

Gambar 3.3.7.1 Script Constraint Table

Hasil dari *script* tersebut yang telah diimplementasikan ke dalam aplikasi adalah sebagai berikut.



Gambar 3.3.7.2 Double Booking Detected

Seperti yang dapat kita lihat pada gambar 3.3.7.2 tersebut ketika seseorang ingin melakukan *double booking*, aplikasi akan secara otomatis mendeteksi dan tidak dapat melakukan *booking*.

3.3.8 Pembuatan *Table Master* Komisi dan Melakukan Kontrol Inputan

Pekerjaan tambahan yang penulis dapatkan adalah membuat *table master* untuk besaran komisi yang nantinya akan dimasukkan datanya secara otomatis oleh aplikasi Berleha. Tujuan awal dibuatnya *table* ini adalah untuk menghitung besaran komisi yang nantinya akan kita terima dari setiap maskapai.

Setelah penulis menyelesaikan pembuatan *table* ini, ternyata aplikasi Berleha masih belum siap dalam menginput data besaran komisi ini. Berikut adalah *Script* untuk *Table Master* Komisi

```
CREATE TABLE FLIGHT_COMMISSION
(
  ID VARCHAR2(40),
  AIRLINE_ID NUMBER,
  ORIGIN VARCHAR2(5 BYTE),
  DESTINATION VARCHAR2(5 BYTE),
  RETURN INT DEFAULT 0,
  NO_PENERBANGAN VARCHAR2(30 BYTE),
  A NUMBER,
  B NUMBER,
  C NUMBER,
  D NUMBER,
  E NUMBER,
  F NUMBER,
  G NUMBER,
  H NUMBER,
  I NUMBER,
  J NUMBER,
  K NUMBER,
  L NUMBER,
  M NUMBER,
  N NUMBER,
  O NUMBER,
  P NUMBER,
  Q NUMBER,
  R NUMBER,
  S NUMBER,
  T NUMBER,
  U NUMBER,
  V NUMBER,
  W NUMBER,
  X NUMBER,
  Y NUMBER,
  Z NUMBER
)
TABLESPACE VWR_TABLE1
PCTUSED 40
PCTFREE 10
INITRANS 1
MAXTRANS 255
STORAGE (
  INITIAL 64K
  NEXT 1M
  MAXSIZE UNLIMITED
  MINEXTENTS 1
  MAXEXTENTS UNLIMITED
  PCTINCREASE 0
  FREELISTS 1
  FREELIST_GROUPS 1
  BUFFER_POOL DEFAULT
)
```

Gambar 3.3.8.1 Script Pembuatan *Table Master* Komisi

3.3.9 Menghitung Besaran Komisi dan Menginput ke Dalam Table Komisi

Sebuah permasalahan yang sempat timbul dalam proyek ini adalah berapa besaran komisi yang diterima oleh kita dari setiap maskapai penerbangan dan berapa persen yang akan kita berikan terhadap sub-agent yang kita miliki. Masalah lain yang terjadi adalah aplikasi Berleha belum siap untuk melakukan inputan data besaran komisi secara otomatis.

Berdasarkan masalah tersebut akhirnya penulis diberikan sebuah dokumen dari sebuah maskapai mengenai besaran persentase komisi yang diperoleh dan penulis diminta untuk mengubahnya ke dalam bentuk data di table komisi.

Rute	No Penerbangan	Discount Agent Berdasarkan Kelas		
		X,V	T,Q	N s/d C
BTH-PGK-BTH	JT128, JT129	15%	10%	5%
KNO-PDG-KNO	JT131, JT130	15%	10%	5%
KNO-BDO-KNO	JT960 / JT961	15%	10%	5%
BTH-BKS-BTH	JT963, JT964	15%	10%	5%
DPS-CGK	JT031, JT029, JT023, JT037, JT015, JT033, JT025, JT027, JT019	15%	10%	5%
CGK-PLM	JT1340, JT348	15%	10%	5%
KNO-CGK	JT207, JT395, JT303, JT201, JT383, JT385, JT203, JT309, JT205, JT219, JT209	15%	10%	5%
CGK-KNO	JT210, 380, JT300, JT200, JT204, JT398, JT382, JT202, JT206, JT208, JT212, JT218	15%	10%	5%
TKG-CGK	JT243	15%	10%	5%
CGK-PDG	JT252, JT350, JT254	15%	10%	5%
PKU-CGK	JT291, JT295, JT297	15%	10%	5%
SUB-BDJ	JT312, JT222	15%	10%	5%
BDJ-SUB	JT313, JT223	15%	10%	5%
SUB-BDJ	JT314	15%	10%	5%
CGK-BDJ	JT320, JT528, JT524, JT328	15%	10%	5%
BDJ-CGK	JT321, JT327	15%	10%	5%
PLM-CGK	JT331, JT1341	15%	10%	5%
CGK-PLM	JT340	15%	10%	5%
PDG-CGK	JT353, JT255, JT257	15%	10%	5%
CGK-PDG	JT356	15%	10%	5%
SUB-BPN	JT360, JT266, JT366	15%	10%	5%

Gambar 3.3.9.1 Data Besaran Komisi

ID	AIRLINE_CODE	ORIGIN	DESTINATION	RETURN	FLIGHTNUM	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	
1018795F9E22202E05344030A0A0865	JT	BTH	PGK	1	JT128	0,05	0,05	0,05	0,05	0	0	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0	0	0,1	0	0,05	0,1	0
1087292DC7102305E053444680CBA705	JT	DPS	CGK	0	JT031	0,05	0,05	0,05	0,05	0	0	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0	0	0,1	0	0,05	0,1	0
1018795F9E24202E05344030A0A0865	JT	KNO	PDG	1	JT131	0,05	0,05	0,05	0,05	0	0	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0	0	0,1	0	0,05	0,1	0
1018795F9E25202E05344030A0A0865	JT	KNO	BDO	1	JT960	0,05	0,05	0,05	0,05	0	0	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0	0	0,1	0	0,05	0,1	0
1087292DC7112305E053444680CBA705	JT	DPS	CGK	0	JT029	0,05	0,05	0,05	0,05	0	0	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0	0	0,1	0	0,05	0,1	0
1087292DC7002305E053444680CBA705	JT	KNO	BDO	1	JT961	0,05	0,05	0,05	0,05	0	0	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0	0	0,1	0	0,05	0,1	0
1087292DC7082305E053444680CBA705	JT	BTH	PGK	1	JT129	0,05	0,05	0,05	0,05	0	0	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0	0	0,1	0	0,05	0,1	0
1087292DC70C2305E053444680CBA705	JT	KNO	PDG	1	JT130	0,05	0,05	0,05	0,05	0	0	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0	0	0,1	0	0,05	0,1	0
1087292DC70E2305E053444680CBA705	JT	BTH	BKS	1	JT964	0,05	0,05	0,05	0,05	0	0	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0	0	0,1	0	0,05	0,1	0
1087292DC70F2305E053444680CBA705	JT	BTH	BKS	1	JT963	0,05	0,05	0,05	0,05	0	0	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0	0	0,1	0	0,05	0,1	0
1087292DC7122305E053444680CBA705	JT	DPS	CGK	0	JT023	0,05	0,05	0,05	0,05	0	0	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0	0	0,1	0	0,05	0,1	0
1087292DC7132305E053444680CBA705	JT	DPS	CGK	0	JT037	0,05	0,05	0,05	0,05	0	0	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0	0	0,1	0	0,05	0,1	0
1087292DC7142305E053444680CBA705	JT	DPS	CGK	0	JT015	0,05	0,05	0,05	0,05	0	0	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0	0	0,1	0	0,05	0,1	0
1087292DC7152305E053444680CBA705	JT	DPS	CGK	0	JT033	0,05	0,05	0,05	0,05	0	0	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0	0	0,1	0	0,05	0,1	0
1087292DC7162305E053444680CBA705	JT	DPS	CGK	0	JT025	0,05	0,05	0,05	0,05	0	0	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0	0	0,1	0	0,05	0,1	0
1087292DC7172305E053444680CBA705	JT	DPS	CGK	0	JT027	0,05	0,05	0,05	0,05	0	0	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0	0	0,1	0	0,05	0,1	0
1087292DC7182305E053444680CBA705	JT	DPS	CGK	0	JT019	0,05	0,05	0,05	0,05	0	0	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0	0	0,1	0	0,05	0,1	0

Gambar 3.3.9.2 Isi Table Komisi Hasil Konversi

3.3.10 Pembuatan Query Insert Seluruh Data

Sebelum melakukan peluncuran aplikasi, setiap aplikasi harus menjalani sebuah tahap yang dinamakan *testing*. *Testing* tahap terakhir adalah UAT (*User Acceptance Test*) dimana kita melakukan demo kepada pihak pengelola seolah – olah aplikasi kita sedang digunakan secara sungguhan.

Dalam *test* ini tidak menutup kemungkinan terjadinya *crash* data baik karena kesalahan dari user maupun aplikasinya yang masih memiliki celah dan dapat ditembus oleh user. Oleh karena itu Bapak Eric meminta kepada penulis untuk dibuatkan *insert query* untuk *table – table* yang penting seperti *FLIGHT_CLASS_FARE*.

Penulis menjawab tugas tersebut dengan melakukan *export* data menjadi *.sql* sehingga secara otomatis akan ter-*convert* menjadi *insert*

query di dalamnya. Pembuatan *insert query* ini sendiri bertujuan untuk mengembalikan data yang *corrupt* secara instan pada saat dilakukannya UAT tersebut. Berikut adalah contoh *insert query* yang telah penulis siapkan.

```

SET DEFINE OFF;
--SQL Statement which produced this data:
--
--  SELECT * FROM VWR.FLIGHT_CLASS_FARE_AIRASIA;
--
Insert into VWR.FLIGHT_CLASS_FARE_AIRASIA
(CLASS_FARE_ID, FLIGHTAVAILID, CLASS_NAME, CLASS_AVAIL_SEAT, CLASS_RATE,
BASIC_RATE, TAX, IWJR, AGENT_DISCOUNT, EXTRA_COVER,
NTA, SURCHARGE_FEE, INSENTIF, TOTAL, RADIO_ID,
CLASS_SEQUENCE, LASTUPDATE, PSC, CHD_FARE, INF_FARE,
UPD_FLAG, DEP_TIME, STATUS_PARTITION)
Values
(28588590, 'f8866d52-b7a8-4a99-8d5f-e6471c3a3e09', 'G', 7, 450900,
369000, 81900, NULL, NULL, NULL,
NULL, NULL, NULL, NULL, '0~Z~Z01H00~AAB1~2~X|QZ~7556~ ~CGK~08/25/2015
05:50~JOG~08/25/2015 06:55~@08/24/2015 22:50:00',
1, TO_DATE('08/24/2015 16:48:33', 'MM/DD/YYYY HH24:MI:SS'), NULL, NULL, 150000,
NULL, TO_DATE('08/25/2015 05:50:00', 'MM/DD/YYYY HH24:MI:SS'), 'N');
Insert into VWR.FLIGHT_CLASS_FARE_AIRASIA
(CLASS_FARE_ID, FLIGHTAVAILID, CLASS_NAME, CLASS_AVAIL_SEAT, CLASS_RATE,
BASIC_RATE, TAX, IWJR, AGENT_DISCOUNT, EXTRA_COVER,
NTA, SURCHARGE_FEE, INSENTIF, TOTAL, RADIO_ID,
CLASS_SEQUENCE, LASTUPDATE, PSC, CHD_FARE, INF_FARE,
UPD_FLAG, DEP_TIME, STATUS_PARTITION)
Values
(28588591, 'f8866d52-b7a8-4a99-8d5f-e6471c3a3e09', 'I', 7, 670900,
569000, 101900, NULL, NULL, NULL,
NULL, NULL, NULL, NULL, '0~ZF~ZF1F00~AAB1~5~X|QZ~7556~ ~CGK~08/25/2015
05:50~JOG~08/25/2015 06:55~@08/24/2015 22:50:00',
2, TO_DATE('08/24/2015 16:48:33', 'MM/DD/YYYY HH24:MI:SS'), NULL, NULL, 150000,
NULL, TO_DATE('08/25/2015 05:50:00', 'MM/DD/YYYY HH24:MI:SS'), 'N');
Insert into VWR.FLIGHT_CLASS_FARE_AIRASIA
(CLASS_FARE_ID, FLIGHTAVAILID, CLASS_NAME, CLASS_AVAIL_SEAT, CLASS_RATE,
BASIC_RATE, TAX, IWJR, AGENT_DISCOUNT, EXTRA_COVER,
NTA, SURCHARGE_FEE, INSENTIF, TOTAL, RADIO_ID,
CLASS_SEQUENCE, LASTUPDATE, PSC, CHD_FARE, INF_FARE,
UPD_FLAG, DEP_TIME, STATUS_PARTITION)
Values
(28588592, 'c2d78108-ab94-40af-8f77-5094f6d59f7c', 'O', 7, 442936,
361760, 81176, NULL, NULL, NULL,
NULL, NULL, NULL, NULL, '0~K~K01H06~AAB1~1~X|QZ~7552~ ~CGK~08/25/2015
10:35~JOG~08/25/2015 11:40~@08/25/2015 03:35:00',
0, TO_DATE('08/24/2015 16:48:33', 'MM/DD/YYYY HH24:MI:SS'), NULL, NULL, 150000,
NULL, TO_DATE('08/25/2015 10:35:00', 'MM/DD/YYYY HH24:MI:SS'), 'N');
Insert into VWR.FLIGHT_CLASS_FARE_AIRASIA
(CLASS_FARE_ID, FLIGHTAVAILID, CLASS_NAME, CLASS_AVAIL_SEAT, CLASS_RATE,

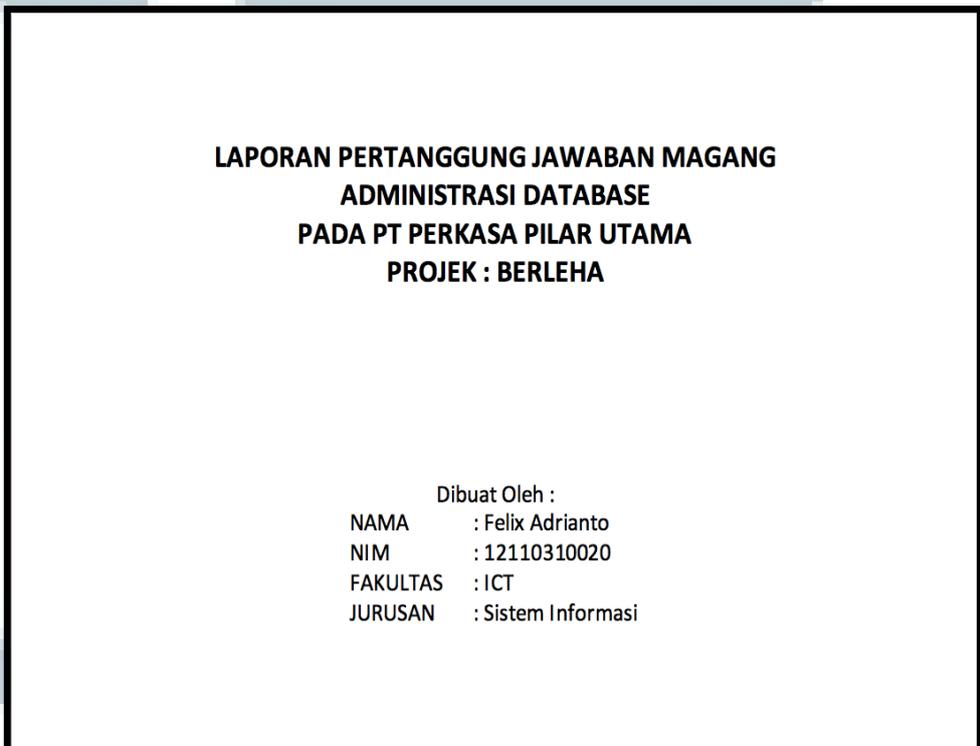
```

Gambar 3.3.10.1 Contoh Insert Query

3.3.11 Dokumentasi Database

Setelah penulis menyelesaikan acara kerja magang di PT Perkasa Pilar Utama, penulis diberikan sebuah tugas akhir untuk melakukan dokumentasi dari *database / schema* yang digunakan terutama yang telah penulis kerjakan selama ini. Hasil dokumentasi yang penulis berikan kepada pihak kantor adalah sebuah dokumen yang berisikan *codingan – codingan* yang telah penulis kerjakan selama penulis melakukan kerja magang.

Fungsi dari dokumentasi ini sendiri adalah untuk melakukan *review* kembali *script – script* yang ada dan dilihat apakah semua *script* yang ada dan digunakan saat ini sudah benar. Berikut adalah contoh dokumentasi yang penulis berikan kepada pihak PT Perkasa Pilar Utama.



Gambar 3.3.11.1 Cover Dokumentasi Database

```

BUAT CONSTRAIN DI TABLE (1 CONSTRAINT LANGSUNG 3 KOLOM)
ALTER TABLE nama_table ADD CONSTRAINT nama_constraint UNIQUE (nama_column1, nama_column2,
nama_column3);

CHECK DURASI SEJAK START MASUKSAMPAI ~ PER HARI INI (JIKA WAKTU EXPIRED TIDAK ADA)
CREATE OR REPLACE PROCEDURE CHECK_LAMA
IS code int;
BEGIN
update trb a set (a.lama) = (select (SUBSTR((systimestamp-b.jam),12,2)*60+ SUBSTR((systimestamp-
b.jam),15,2)) from trb b where a.jam = b.jam and a.waktu = trunc(sysdate)) where a.waktu =
trunc(sysdate) and a.status='P';
commit;
END CHECK_LAMA;
/

BEGIN
SYS.DBMS_SCHEDULER.CREATE_JOB
(
  job_name          => 'UPDATE_DURASI',
  start_date        => TO_TIMESTAMP_TZ('2015/07/07 23:40:00.000000 +07:00','yyyy/mm/dd
hh24:mi:ss.ff tzz'),
  repeat_interval   => 'FREQ=SECONDLY;INTERVAL=1',
  end_date          => NULL,
  job_class         => 'DEFAULT_JOB_CLASS',
  job_type          => 'STORED_PROCEDURE',
  job_action        => 'CHECK_LAMA',
  comments          => NULL
);
END;
/

MENYALAKAN SCHEDULER UPDATE_DURASI
BEGIN

```

Gambar 3.3.11.2 Contoh Hasil Dokumentasi Database

3.4 Pekerjaan Tambahan

Selain menjadi *Database Administrator*, penulis juga mendapatkan pekerjaan sampingan di bidang *networking*. Karena dalam proyek ini semuanya berhubungan dengan internet maka sebuah kestabilan dan kecepatan internet sangat berpengaruh ke dalam kinerja tim Berleha ini.

Solusi yang kita miliki adalah dengan menggandeng dua buah provider penyedia layanan internet. Tetapi kendala yang didapati adalah sebagian besar dari tim tidak ada yang mengerti mengenai *networking* seperti ini.

Pada akhirnya penulis dan rekan kerja penulis yaitu Bapak Verry yang menangani masalah ini. Dari tahap pemantauan pemasangan, peletakan router *Wi-*

Fi yang baik agar satu lantai kantor dapat menikmati layanan internetnya hingga pengaturan dari router itu sendiri.

3.5 Kendala yang Dihadapi

Penulis pada awalnya kesulitan dalam mempelajari *schema database* yang ada dikarenakan banyak sekali istilah yang belum pernah penulis lihat dan pelajari sebelumnya. Selain itu ilmu yang dibutuhkan dalam proyek ini seperti partisi, backup, job, trigger dan lainnya belum pernah dipelajari sebelumnya pada saat kegiatan belajar mengajar di Universitas Multimedia Nusantara. Aplikasi yang digunakan untuk mengakses sebuah *database* pun cukup banyak sehingga penulis terkadang salah dalam menggunakan aplikasi yang tepat.

3.6 Solusi

Penulis perlu lebih banyak membaca dan mempelajari istilah – istilah yang digunakan dalam pembuatan database dan fungsi dari setiap istilah – istilah tersebut. Penulis juga membutuhkan buku / info tambahan mengenai materi apa saja yang digunakan dalam dunia pekerjaan yang sesungguhnya, sehingga penulis dapat lebih mempersiapkan diri ketika kerja. Penulis juga banyak bertanya dan berdiskusi kepada Bapak Eric Elkana maupun Bapak Helmi mengenai fungsi – fungsi dari *database* yang diinginkan. Selain itu penulis juga bertanya kepada rekan – rekan kantor mengenai aplikasi yang tepat agar pengerjaannya dapat lebih cepat dan lebih baik hasilnya. Diluar itu semua, penulis juga mempelajari secara

online melalui website dari Oracle langsung mengenai fitur – fitur apa saja yang dimiliki oleh oracle dan bagaimana cara pengaplikasiannya.

Dengan adanya solusi – solusi tersebut akhirnya penulis mampu untuk memenuhi kebutuhan dari PT Perkasa Pilar Utama dan memenuhi tujuan dari kegiatan kerja magang ini. Penulis mampu mengembangkan ilmu pengetahuan dan pengalaman dalam bidang *database* secara nyata dan mengetahui resiko – resiko apa saja ketika berkerja menjadi seorang *Database Administrator*.

