

TB DATABASE (DATA STATISTIK RS DR.HARYOTO)



Disusun oleh :

Dinda Anik Masruro

D4 TI 2C

1.18.4.003

PROGRAM DIPLOMA IV POLITEKNIK POS INDONESIA

POLITEKNIK POS INDONESIA

BANDUNG

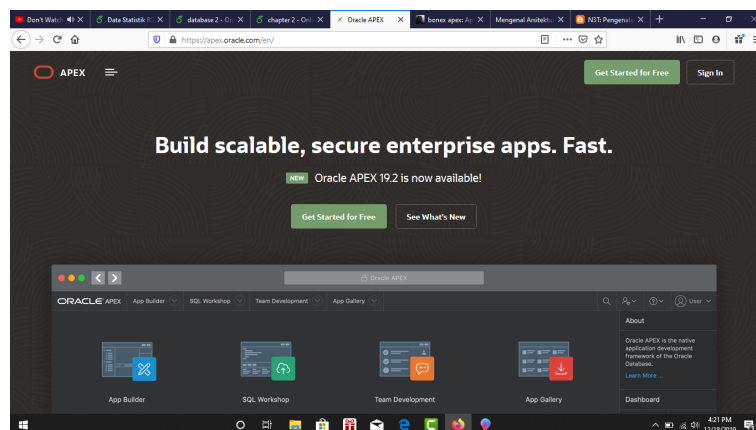
2019

1 Membuat Aplikasi dengan Oracle APEX

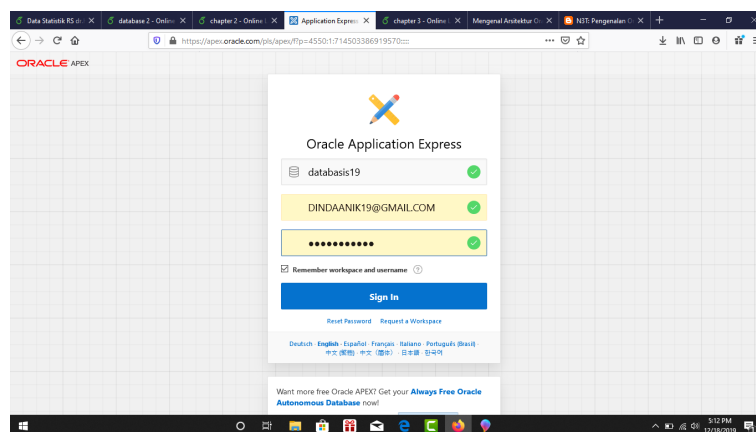
Oracle Application Express (Oracle APEX) yang dulu disebut HTML-DB adalah sebuah framework yang berbasis pada sebuah database dedicated (sementara ini sampai versi terbaru masih dedicated untuk Oracle Db saja dan lisensi include dalam lisensi database), ini artinya apa bahwa engine aplikasi dibangun sepenuhnya didalam sebuah database. Bahkan untuk arsitektur Embedded PL/SQL Gateway seperti yang dipakai dalam Oracle XE dan Oracle 11G file image (library,css,theme,dll) disimpan didalam database metadata juga. Inilah hal yang berbeda dibandingkan framework yang lain. Berikut ini merupakan tata cara membuat suatu aplikasi di Oracle APEX:

1.1 Tutorial masuk dan login Oracle APEX

1. Pertama, kita pilih menggunakan APEX Online atau APEX offline. Disini saya akan menggunakan APEX online dengan cara kunjungi website <https://apex.oracle.com/en>
2. Kedua, setelah masuk pada link akan muncul seperti gambar dibawah ini kemudian silahkan sign in.

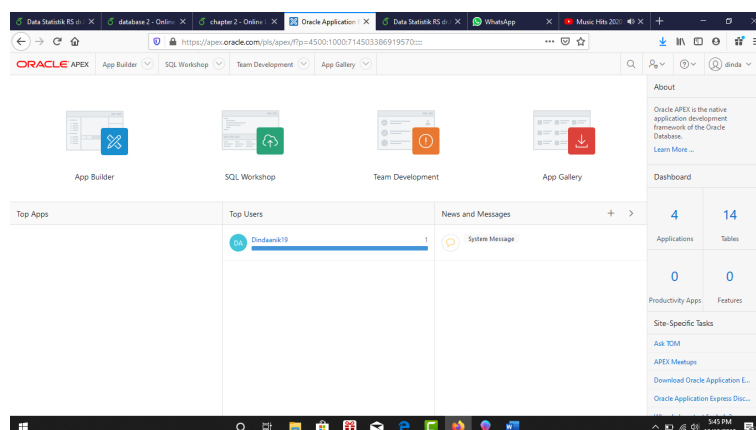


3. ketiga, pilih sign in jika kita sudah memiliki workspace. Lalu masukkan nama database yang telah dibuat, nama username dan password tetapi jika kita belum memiliki akun request workspace. Maka ikuti langkah-langkah pada form pendaftaran. Ketika sudah kita telah selesai memasukkan nama database, username dan password yang telah dibuat. Maka kita bisa masuk dengan memasukkan data dengan benar sesuai data yang telah kita miliki.



1.1.1 Langkah-langkah Membuat Aplikasi Di Oracle APEX

1. Pertama, jika kita sudah menginstall aplikasi oracle apex atau sudah mendaftar di oracle apex online maka setelah itu akan masuk ke dalam aplikasi oracle apex.



2. Kemudian, pilih sql workshop dan pilih sql command. Sql command digunakan untuk mengetikkan query yang akan digunakan dan dibuat.

3. Langkah selanjutnya adalah pembuatan database yang dimulai dengan membuat table. Disini saya akan membuat aplikasi data statistik dari RS yang terdapat lima tabel yaitu pasien, dokter, obat, resep, dan diagnosa. Query yang diinputkan di sql command seperti berikut :

(a) Table Pasien

Table pasien yang mempunyai 6 kolom yaitu id pasien, nama pasien, tanggal lahir, alamat, no telp, dan jumlah periksa. Pada table ini yang menjadi primary keynya adalah id pasien.

```
1 CREATE TABLE PASIEN
2 (ID_PASIEN VARCHAR(15) CONSTRAINT ID_PASIEN_PK PRIMARY KEY,
3 NAMA_PASIEN VARCHAR(50) NOT NULL,
4 TANGGAL_LAHIR DATE NOT NULL,
5 ALAMAT VARCHAR(50) NOT NULL,
6 NO_TELP VARCHAR(15) NOT NULL,
7 JUMLAH_PERIKSA INT DEFAULT 0);
```

(b) Table Dokter

Table dokter yang mempunyai 6 kolom yaitu id dokter, nama dokter, alamat, no telp, keahlian, dan tarif. Pada table ini yang menjadi primary keynya adalah id dokter.

```
17 CREATE TABLE DOKTER
18 (ID_DOKTER VARCHAR(15) CONSTRAINT ID_DOKTER_PK PRIMARY KEY,
19 NAMA_DOKTER VARCHAR(50) NOT NULL,
20 ALAMAT VARCHAR(50) NOT NULL,
21 NO_TELP VARCHAR(15) NOT NULL,
22 KEAHLIAN VARCHAR(25) NOT NULL,
23 TARIF VARCHAR(15) NOT NULL);
```

(c) Table Obat

Table obat yang mempunyai 4 kolom yaitu id-obat, nama-obat, kegunaan, dan harga-obat. Pada table ini yang menjadi primary keynya adalah id-obat.

```
10 CREATE TABLE OBAT
11 (ID_OBAT VARCHAR(15) CONSTRAINT ID_OBAT_PK PRIMARY KEY,
12 NAMA_OBAT VARCHAR(50) NOT NULL,
13 KEKURANGAN VARCHAR(100) NOT NULL,
14 HARGA_OBAT VARCHAR(15) NOT NULL);
```

(d) Table Resep

Table resep yang mempunyai 5 kolom yaitu id resep, isi resep, id obat, id pasien, dan id dokter. Pada table ini yang menjadi primary keynya adalah id-resep dengan id pasien, id obat, dan id dokter sebagai foreign keynya.

```
34 CREATE TABLE RESEP
35 (ID_RESEP INT CONSTRAINT ID_RESEP_PK PRIMARY KEY,
36 ISI_RESEP VARCHAR(50) NOT NULL,
37 ID_OBAT VARCHAR(15) NOT NULL CONSTRAINT ID_OBAT_FK REFERENCES OBAT(ID_OBAT),
38 ID_PASIEN VARCHAR(15) NOT NULL CONSTRAINT ID_PASIEN_FK REFERENCES PASIEN(ID_PASIEN),
39 ID_DOKTER VARCHAR(15) NOT NULL CONSTRAINT ID_DOKTER_FK REFERENCES DOKTER(ID_DOKTER));
```

(e) Table Diagnosa

Table resep yang mempunyai 6 kolom yaitu id diagnosa, tanggal periksa, hasil diagnosa, id resep, id pasien, dan id-dokter. Pada table ini yang menjadi primary keynya adalah id diagnosa dengan id pasien, id resep, dan id dokter sebagai foreign keynya.

```
26 CREATE TABLE DIAGNOSA
27 (ID_DIAGNOSA INT CONSTRAINT ID_DIAGNOSA_PK PRIMARY KEY,
28 TANGGAL_PERIKSA DATE NOT NULL,
29 HASIL_DIAGNOSA VARCHAR(40) NOT NULL,
30 ID_PASIENT VARCHAR(15) NOT NULL CONSTRAINT ID_PASIENT_FK REFERENCES PASIENT(ID_PASIENT),
31 ID_DOKTER VARCHAR(15) NOT NULL CONSTRAINT ID_DOKTER_FK REFERENCES DOKTER(ID_DOKTER),
32 ID_RESEP INT NOT NULL CONSTRAINT ID_RESEP_FK REFERENCES RESEP(ID_RESEP));
33
```

4. Setelah membuat table maka selanjutnya adalah mengisi baris pada kolom table yang telah dibuat. Berikut adalah cara menambahkan baris pada kolom:

(a) Table Pasien

Pada table pasien mempunyai 6 kolom maka baris yang akan dimasukkan juga sebanyak 6 baris sesuai dengan jumlah kolom.

```
40 INSERT ALL
41 INTO PASIENT VALUES('PS001', 'RUMIHA LESTARI', '81/14/2001', 'LAMPUNG', '082108449823', 0)
42 INTO PASIENT VALUES('PS002', 'DINDEN ANDEK HUGURNO', '85/11/1999', 'LUPACJANG', '083847410088', 0)
43 SELECT * FROM DUAL;
44
```

(b) Table Dokter

Pada table dokter mempunyai 6 kolom maka baris yang akan dimasukkan juga sebanyak 6 baris sesuai dengan jumlah kolom.

```
40 INSERT ALL
41 INTO DOKTER VALUES('DD0001', 'Dr. UTOMO', 'SURABAYA', '081122677100', 'SPECIALIS KULIT', 'Rp. 50000')
42 INTO DOKTER VALUES('DD0002', 'Dr. SRI WARMUTI', 'JOMBANG', '087698256799', 'DOKTER UPJAY', 'Rp. 25000')
43 SELECT * FROM DUAL;
44
```

(c) Table Obat

Pada table obat mempunyai 4 kolom maka baris yang akan dimasukkan juga sebanyak 4 baris sesuai dengan jumlah kolom.

```
53 INSERT ALL
54 INTO OBAT VALUES('OB0001', 'HYDROCORTISONE CREAM', 'HENGKILANGKAN BATAL JAY ALENET', 'Rp. 10000')
55 INTO OBAT VALUES('OB0002', 'PARACETAMOL', 'HENGKILANGKAN RASA SAKIT DAN PERUBAH PANAS', 'Rp. 10000')
56 SELECT * FROM DUAL;
57
```

(d) Table Resep

Pada table resep mempunyai 5 kolom maka baris yang akan dimasukkan juga sebanyak 5 baris sesuai dengan jumlah kolom. Akan tetapi perhatikan foreign key yang diambil dari table obat, pasien dan juga dokter harus sesuai dengan yang sudah dimasukkan dalam table sebelumnya.

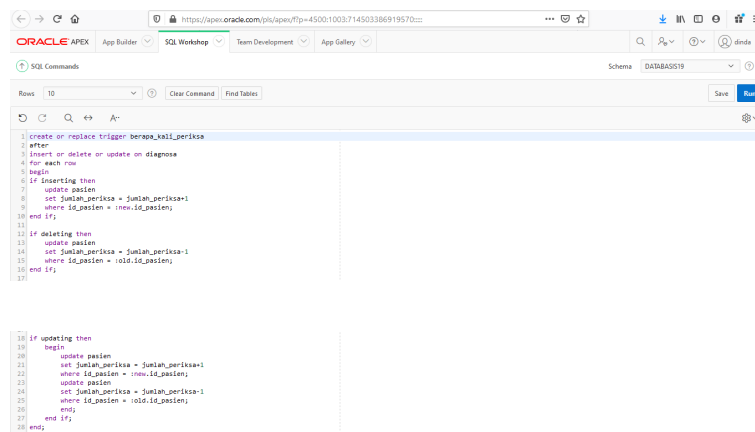
```
57 INSERT INTO RESEP VALUES (ID_RESEP_SEQ.NEXTVAL, 'DINAMIA 30 SEMESTER', '08T001', 'P0001', 'D00001');
58 INSERT INTO RESEP VALUES (ID_RESEP_SEQ.NEXTVAL, 'DINAMIA 30 SEMESTER', '08T002', 'P0002', 'D00002');
```

(e) Table Diagnosa

Pada table resep mempunyai 6 kolom maka baris yang akan dimasukkan juga sebanyak baris sesuai dengan jumlah kolom. Akan tetapi perhatikan foreign key yang diambil dari table resep, pasien dan juga dokter harus sesuai dengan yang sudah dimasukkan dalam table sebelumnya.

```
59 INSERT INTO DIAGNOSA VALUES (ID_DIAGNOSA_SEQ.NEXTVAL, '12-12-2019', 'ALERGI', 'P0001', 'D00001', '1');
60 INSERT INTO DIAGNOSA VALUES (ID_DIAGNOSA_SEQ.NEXTVAL, '12-12-2019', 'DEMAM', 'P0002', 'D00002', '2');
```

5. Setelah kolo dan baris telah diisi maka selanjutnya adalah pembuatan fungsi trigger. Fungsi trigger adalah fungsi yang digunakan untuk menampilkan suatu data secara otomatis pada saat insert, delete, dan update pada table. Pada aplikasi ini saya membuat trigger yang bernama berapa kali periksa, yang mana ketika kita menginsert, mendelete, dan mengupdate pada table Pasien maka yang akan mengalami perubahan secara otomatis pada table Diagnosa tepatnya di colum tanggal-periksa. Berikut merupakan trigger yang dibuat:

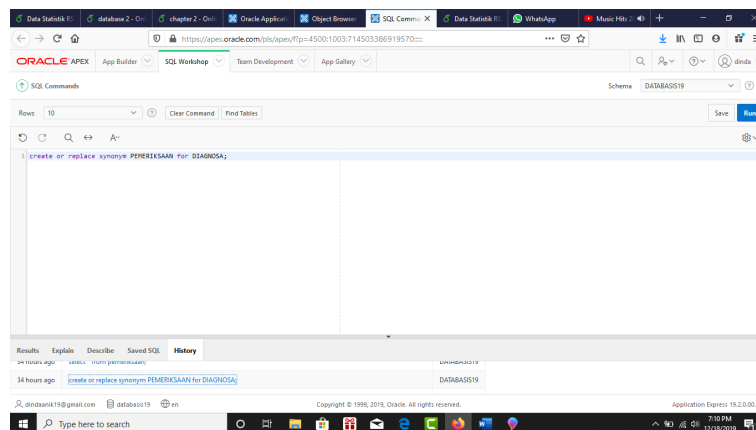


```
1 create or replace trigger berapa_kali_periksa
2 after
3 insert or delete or update on diagnosa
4 for each row
5 begin
6 if inserting then
7 update pasien
8 set jumlah_periksa = jumlah_periksa+1
9 where id_pasien = :new.id_pasien;
10 end if;
11
12 if deleting then
13 update pasien
14 set jumlah_periksa = jumlah_periksa-1
15 where id_pasien = :old.id_pasien;
16 end if;
17
18 if updating then
19 begin
20 update pasien
21 set jumlah_periksa = jumlah_periksa+1
22 where id_pasien = :new.id_pasien;
23 update pasien
24 set jumlah_periksa = jumlah_periksa-1
25 where id_pasien = :old.id_pasien;
26 end;
27 end if;
28 end;
```

6. Setelah itu adalah tahapan membuat view pada aplikasi. View merupakan perintah untuk menggabungkan query join, innerjoin table dengan lebih sederhana. Jadi contoh pada view tersebut akan menampilkan nama pasien, tanggal periksa, dan nama dokter dimana ke-3nya merupakan dari table yang berbeda. Jadi view lebih menyederhanakan query yang dimasukkan.

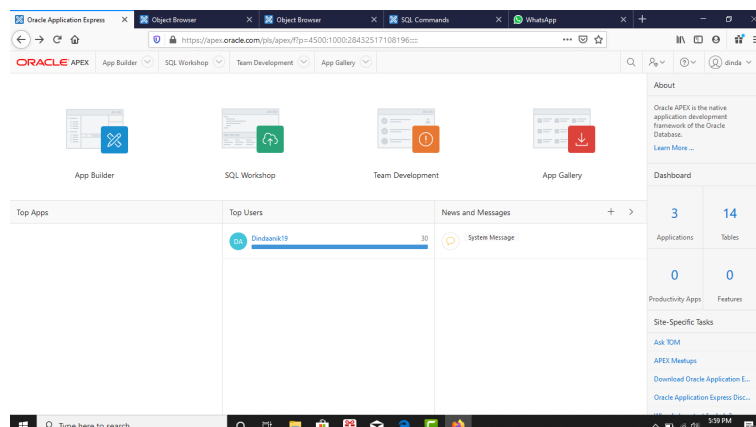
```
1 create view data_pemeriksaan as select
2 b.nama_pasien,c.tanggal_periksa ,d.nama_dokter
3 from pasien b, diagnosa c, dokter d
4 where b.id_pasien=c.id_pasien
5 and c.id_dokter=d.id_dokter;
```

7. Selanjutnya adalah pembuatan sinonim. Sinonim digunakan untuk memanggil nama table dengan nama yang berbeda. Contohnya table Diagnosa dapat dipanggil dengan nama Pemeriksaan.



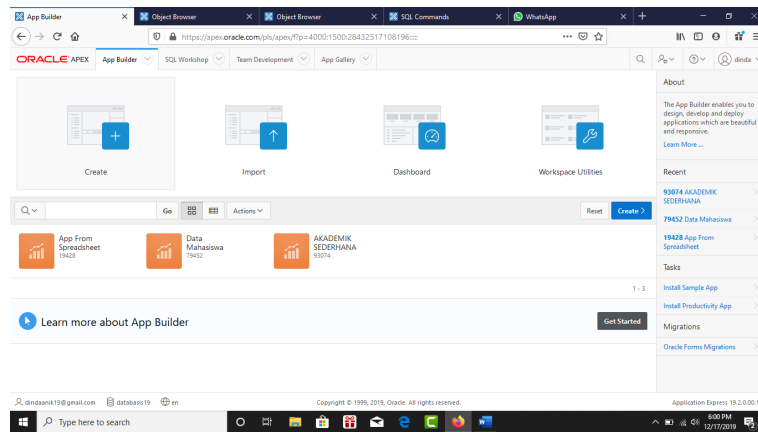
The screenshot shows the Oracle APEX SQL Workshop interface. The SQL Command window contains the following code: `1 create or replace synonym PEMERIKSAAN for DIAGNOSA;`. The interface includes a top navigation bar with 'ORACLE APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'App Gallery'. Below the command window, there are tabs for 'Results', 'Explain', 'Describe', 'Saved SQL', and 'History'. The 'History' tab is active, showing a list of recent queries. The bottom status bar indicates the user is 'dinda' and the schema is 'DATABASE19'.

8. Setelah semuanya selesai, maka selanjutnya adalah pembuatan aplikasi dengan memilih app builder.

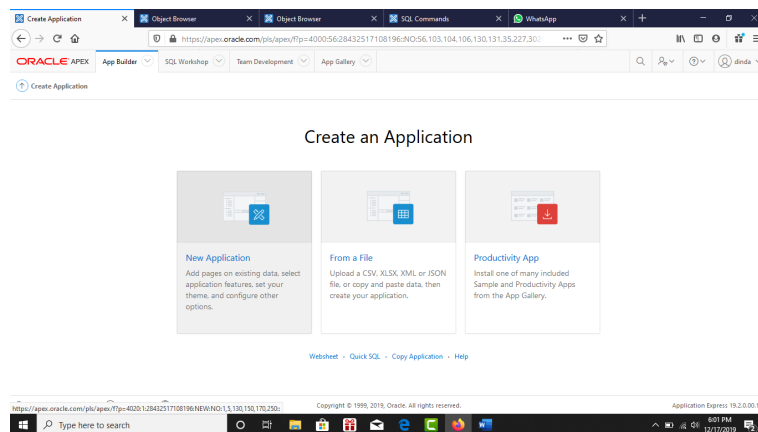


The screenshot shows the Oracle APEX dashboard. The top navigation bar includes 'ORACLE APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'App Gallery'. The main area features a grid of icons for 'App Builder', 'SQL Workshop', 'Team Development', and 'App Gallery'. Below this, there are sections for 'Top Apps', 'Top Users', and 'News and Messages'. The 'Top Users' section shows a bar chart with the user 'Dinda19' at the top. The 'News and Messages' section shows a 'System Message'. On the right side, there is a sidebar with 'About', 'Dashboard', and various links like 'Productivity Apps', 'Site-Specific Tasks', 'Ask TOM', 'APEX Meetups', 'Download Oracle Application E...', and 'Oracle Application Express Disc...'. The bottom status bar shows the time as 5:19 PM on 12/17/2019.

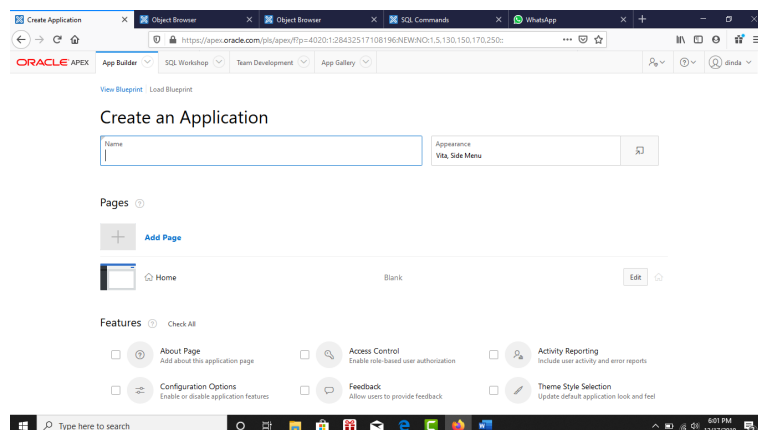
9. Setelah itu pilih create untuk membuat aplikasi yang baru dengan menggunakan data-data yang telah dibuat di oracle apex.



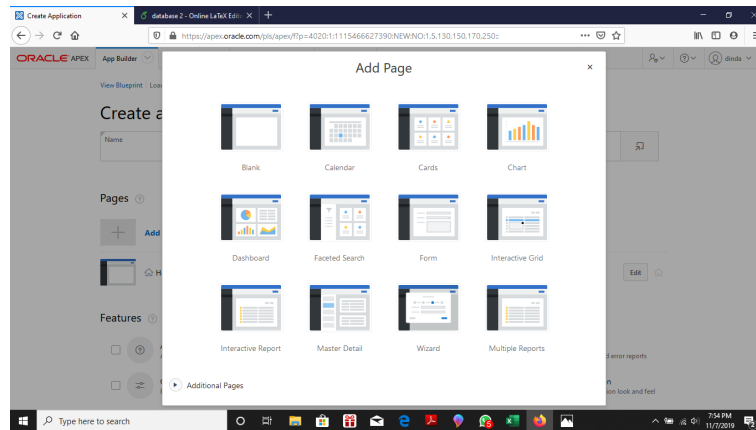
10. Selanjutnya pilih new application yang digunakan untuk membuat aplikasi.



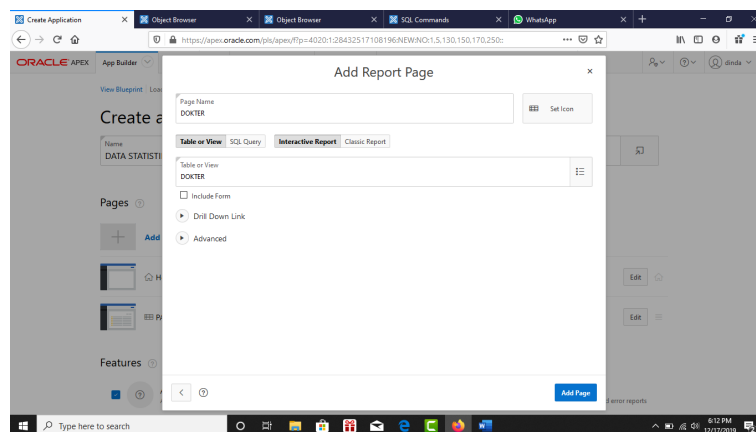
11. Kemudian buatlah nama aplikasi yang akan dibuat, lalu pilih add page untuk memilih page yang akan digunakan untuk membuat aplikasi.



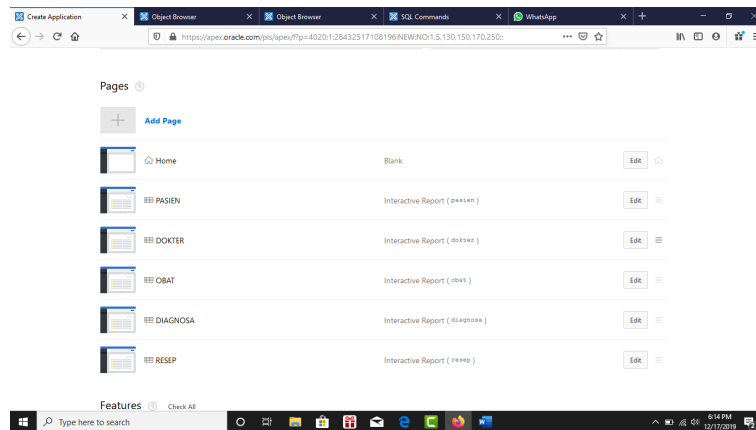
12. Lalu, pilih interactive report untuk page aplikasi yang dibuat.



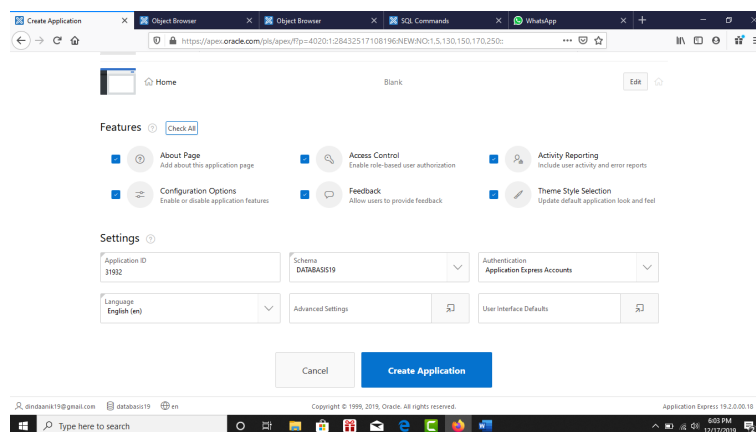
13. Setelah itu page isi page namanya dengan yang merupakan table yang ingin dibuat pada aplikasi. Pilih table dokter pada table or view lalu add page.



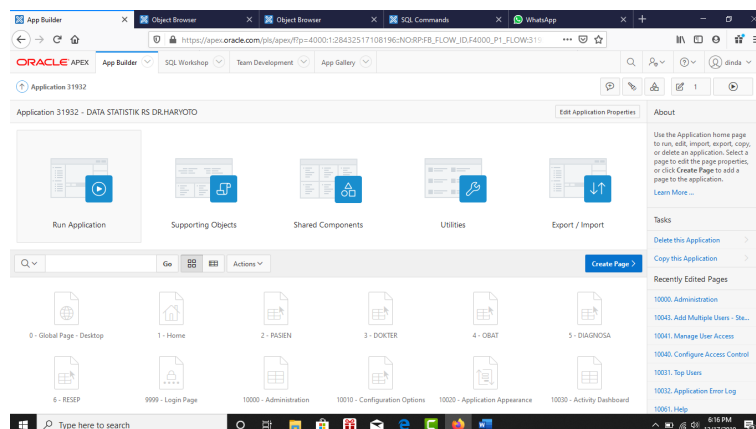
14. lakukan hal yang sama pilih add page dan pilih page intercative report lalu isi name page dengan pasien, dokter, obat, resep, dan diagnosa secara bergantian. Lalu pilih table atau view table yang yang telah dibuat yaitu pasien, dokter, obat, resep, dan diagnosa juga secara bergantian. Jangan lupa jika ingin menambahkan view yang telah dibuat dengan nama hasil panen pada page name lalu pilih view yang telah dibuat yaitu data pemeriksaan. Lalu add page.



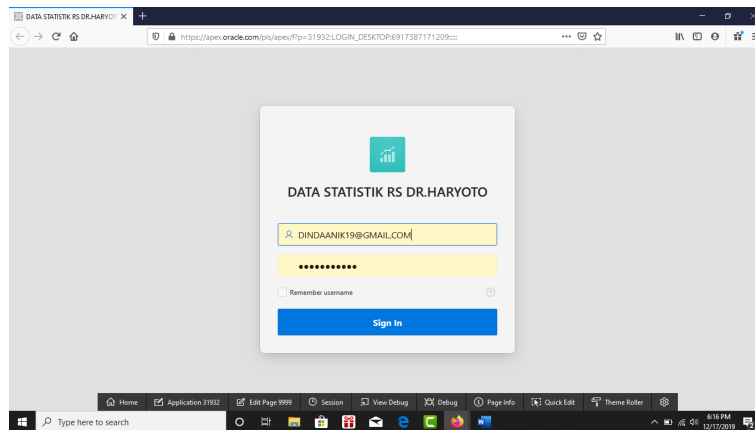
15. Setelah semuanya di add page maka pada features di cek all. Maka setelah itu create application. Tunggu hingga selesai.



16. Setelah itu pilih run application untuk melihat aplikasi yang sudah dibuat.



17. Lalu masukkan username dan password untuk melihat aplikasi yang telah dibuat. Lalu sign in



18. Jadi aplikasi data statistik RS dr. Haryoto sudah selesai dibuat

