

TUGAS BESAR DATABASE II

18 DESEMBER 2019



Disusun oleh :

JOHN KEVIN GIRALDI

D4 TI 2C

1.18.4.049

PROGRAM DIPLOMA IV POLITEKNIK POS INDONESIA

POLITEKNIK POS INDONESIA

BANDUNG

2019

1 PROSES PEMBUATAN TABEL

1. Pertama kita harus membuka website <https://apex.oracle.com/pls/apex/f?p=4550:1:712758388074636:::> untuk melakukan *login* disini saya menggunakan akun dan *workspace* yang telah saya buat :

Workspace : APEX_KEVIN

Username : johnkevin1305@gmail.com

Password : johnkev12

Link aplikasi : <https://apex.oracle.com/pls/apex/f?p=70162:7:110993318567207::NO::>

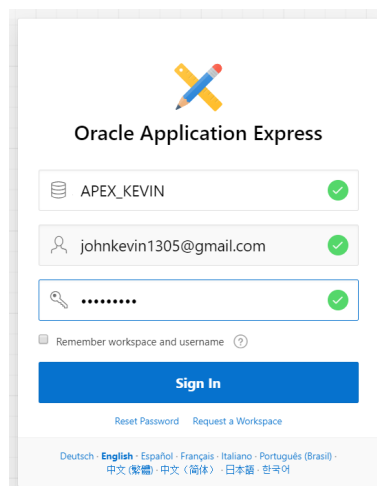


Figure 1: Login

2. Setelah melakukan *login* maka akan muncul tampilan yang terdiri dari berubah ke halaman utama *Oracle Apex* seperti ini:

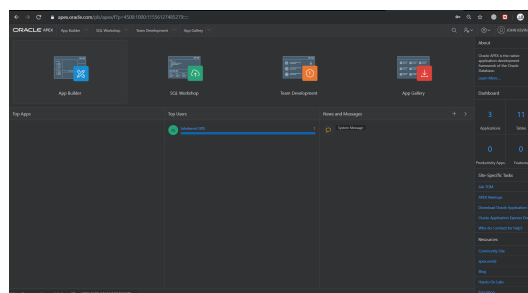


Figure 2: HALAMAN UTAMA ORACLE APEX

3. Lalu pilih *SQL Command* pada *SQL Workshop* untuk menuju ke halaman pengkodean atau *query*.

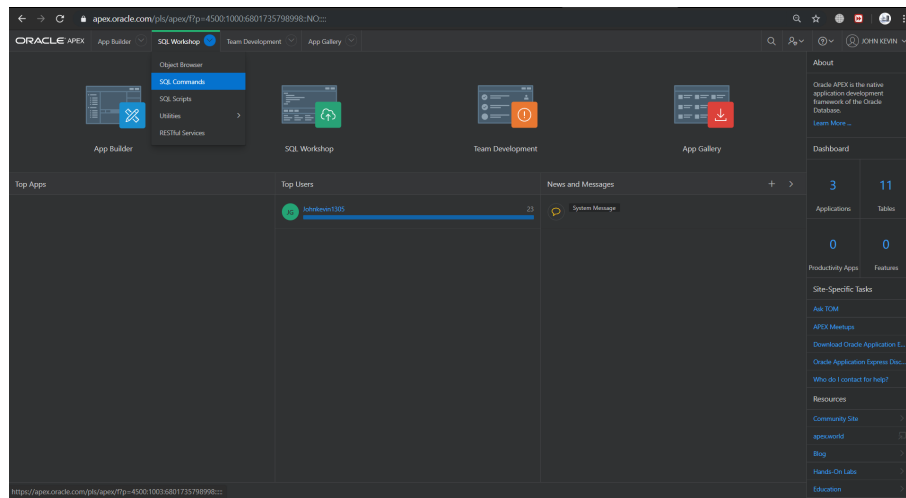


Figure 3: SQL WORKSHOP

4. Setelah muncul halaman pengkodean, sekarang kita akan mulai membuat *query* untuk membentuk tabel-tabel yang kita inginkan. Disini saya akan membuat 6 tabel yang terdiri dari:

- PASIEN dengan *Primary Key* (ID_PASIEN)
- OBAT dengan *Primary Key* (ID_OBAT)
- DOKTER dengan *Primary Key* (ID_DOKTER)
- DIAGNOSA dengan *Primary Key* (ID_DIAGNOSA) yang berelasi dengan tabel PASIEN, DOKTER dan RESEP yang membentuk beberapa FK/*Foreign Key* dari setiap relasi yang di hasilkan. *Foreign Key*(ID_PASIEN _FK), *Foreign Key* (ID_DOKTER _FK), *Foreign Key* (ID_RESEP _FK).
- RESEP dengan *Primary Key* (ID_RESEP) yang berelasi dengan tabel OBAT, PASIEN DAN DOKTER yang membentuk beberapa FK/*Foreign Key* dari setiap relasi yang di hasilkan. *Foreign Key*(ID_OBAT _FK), *Foreign Key* (ID_PASIEN2 _FK), *Foreign Key* (ID_DOKTER2 _FK).
- LOG_PASIEN tanpa *Primary Key* ataupun *Foreign Key*

```

1 CREATE TABLE PASIEN
2 (ID_PASIEN VARCHAR(15) CONSTRAINT ID_PASIEN_PK PRIMARY KEY,
3 NAMA_PASIEN VARCHAR(20) NOT NULL,
4 TANGGAL_LAHIR DATE NOT NULL,
5 ALAMAT VARCHAR(60) NOT NULL,
6 NO_HANDPHONE VARCHAR(15) NOT NULL,
7 JUMLAH_PERIKSA INT DEFAULT 0);
8
9 CREATE TABLE OBAT
10 (ID_OBAT VARCHAR(15) CONSTRAINT ID_OBAT_PK PRIMARY KEY,
11 NAMA_OBAT VARCHAR(20) NOT NULL,
12 KEGUNAAN VARCHAR(100) NOT NULL,
13 HARGA_OBAT VARCHAR(15) NOT NULL);
14
15 CREATE TABLE DOKTER
16 (ID_DOKTER VARCHAR(15) CONSTRAINT ID_DOKTER_PK PRIMARY KEY,
17 NAMA_DOKTER VARCHAR(30) NOT NULL,
18 ALAMAT VARCHAR(20) NOT NULL,
19 NO_TELP VARCHAR(15) NOT NULL,
20 KEAHLIAN VARCHAR(25) NOT NULL,
21 TARIF VARCHAR(15) NOT NULL);

```

Figure 4: PEMBUATAN TABEL

```

22
23 CREATE TABLE RESEP
24 (ID_RESEP INT CONSTRAINT ID_RESEP_PK PRIMARY KEY,
25 ISI_RESEP VARCHAR(50) NOT NULL,
26 ID_OBAT VARCHAR(15) NOT NULL CONSTRAINT ID_OBAT_FK REFERENCES OBAT(ID_OBAT),
27 ID_PASIEN VARCHAR(15) NOT NULL CONSTRAINT ID_PASIEN_FK REFERENCES PASIEN(ID_PASIEN),
28 ID_DOKTER VARCHAR(15) NOT NULL CONSTRAINT ID_DOKTER_FK REFERENCES DOKTER(ID_DOKTER));
29
30 CREATE TABLE DIAGNOSA
31 (ID_DIAGNOSA INT CONSTRAINT ID_DIAGNOSA_PK PRIMARY KEY,
32 TANGGAL_PERIKSA DATE NOT NULL,
33 HASIL_DIAGNOSA VARCHAR(60) NOT NULL,
34 ID_PASIEN VARCHAR(15) NOT NULL CONSTRAINT ID_PASIEN_FK REFERENCES PASIEN(ID_PASIEN),
35 ID_DOKTER VARCHAR(15) NOT NULL CONSTRAINT ID_DOKTER_FK REFERENCES DOKTER(ID_DOKTER),
36 ID_RESEP INT NOT NULL CONSTRAINT ID_RESEP_FK REFERENCES RESEP(ID_RESEP));
37
38 CREATE TABLE LOG_PASIEN
39 (ID_PASIEN VARCHAR(15) NOT NULL,
40 NAMA_PASIEN VARCHAR(20) NOT NULL,
41 TANGGAL DATE NOT NULL,
42 KETERANGAN VARCHAR(60) NOT NULL);

```

Figure 5: PEMBUATAN TABEL

- Setelah kita berhasil membuat tabel-tabel yang diinginkan, selanjutnya kita akan menyisipkan data pada tabel PASIEN, DOKTER, OBAT, RESEP, DIAGNOSA, LOG_PASIEN

```

43 INSERT ALL
44 INTO PASIEN VALUES('PSN001','SUPRPTIO','01/21/1945','CINAHU','087514562',0)
45 INTO PASIEN VALUES('PSN002','SUTARNO','02/23/1965','CILILIN','083414361',0)
46 INTO PASIEN VALUES('PSN003','PALJO','04/26/1955','CIKAMPEK','082567562',0)
47 INTO PASIEN VALUES('PSN004','TUKIEM','07/27/1947','CININDI','086417862',0)
48 INTO PASIEN VALUES('PSN005','SARTONO','08/29/1959','CIKOKOL','08634562',0)
49 SELECT * FROM DUAL;
50
51 INSERT ALL
52 INTO DOKTER VALUES('IDI001','dr. MULYOKO','JAKARTA','0986754532','DOKTER MATA',500000)
53 INTO DOKTER VALUES('IDI002','dr. SUPANTO','BANDUNG','0676564372','DOKTER KULIT',250000)
54 INTO DOKTER VALUES('IDI003','dr. BUMBANG','SERANG','0896789583','DOKTER GIGI',130000)
55 INTO DOKTER VALUES('IDI004','dr. SUHERMAN','DENPASAR','0888767654','DOKTER ANAK',180000)
56 INTO DOKTER VALUES('IDI005','dr. MIZAYA','BOGOR','0892514352','DOKTER UMUM',85000)
57 SELECT * FROM DUAL;
58
59 INSERT ALL
60 INTO OBAT VALUES('OB1001','PARACETAMOL','MENGEHLANGKAN RASA SAKIT',15000)
61 INTO OBAT VALUES('OB1002','DEKAMETHASONE','ALERGI DAN PERADANGAN',100000)
62 INTO OBAT VALUES('OB1003','LYTACUR CYRUP','MENAMBAH NAFSU MAKAN ANAK',50000)
63 INTO OBAT VALUES('OB1004','HYDROCORTISONE','MENGEHLANGKAN GATAL DAN ALERGI',100000)
64 INTO OBAT VALUES('OB1005','AMOKSILIN','ANTIBIOTIK',20000)
65 SELECT * FROM DUAL;

```

Figure 6: MENYISIPKAN DATA

Sebelum menyisipkan data ke tabel berikutnya, kita harus membuat *sequence* terlebih dahulu, untuk mengaktifkan kode yang dirancang dan membuat penomoran secara otomatis atau *auto increment*.

```

68 CREATE SEQUENCE ID_DIAGNOSA_SEQ INCREMENT BY 1 START WITH 770 MAXVALUE 5000 NOCACHE NOCYCLE;
69 CREATE SEQUENCE ID_RESEP_SEQ INCREMENT BY 1 START WITH 880 MAXVALUE 5000 NOCACHE NOCYCLE;

```

Figure 7: MEMBUAT SEQUENCE

Setelah *sequence* dibuat maka *query* untuk menyisipkan data pada tabel RESEP dan DIAGNOSA dapat digunakan.

```

66 INSERT INTO RESEP VALUES (ID_RESEP_SEQ.NEXTVAL, 'DITIMBUN 3X SEHARI', '081001', 'PSN005', 'ID1004');
67 INSERT INTO RESEP VALUES (ID_RESEP_SEQ.NEXTVAL, 'DITIMBUN 3X SEHARI', '081001', 'PSN002', 'ID1002');
68 INSERT INTO RESEP VALUES (ID_RESEP_SEQ.NEXTVAL, 'DITIMBUN 3X SEHARI', '081001', 'PSN001', 'ID1001');
69 INSERT INTO RESEP VALUES (ID_RESEP_SEQ.NEXTVAL, 'DITIMBUN 3X SEHARI', '081002', 'PSN003', 'ID1001');
70 INSERT INTO RESEP VALUES (ID_RESEP_SEQ.NEXTVAL, 'DITIMBUN 3X SEHARI', '081005', 'PSN004', 'ID1005');
71
72 INSERT INTO DIAGNOSA VALUES (ID_DIAGNOSA_SEQ.NEXTVAL, '01/25/1999', 'KURANG NAFSU MAKAN', 'PSN005', 'ID1004', 880);
73 INSERT INTO DIAGNOSA VALUES (ID_DIAGNOSA_SEQ.NEXTVAL, '02/25/1999', 'GATAL-GATAL', 'PSN002', 'ID1002', 881);
74 INSERT INTO DIAGNOSA VALUES (ID_DIAGNOSA_SEQ.NEXTVAL, '03/25/1999', 'Sakit PADA GIGI', 'PSN001', 'ID1001', 882);
75 INSERT INTO DIAGNOSA VALUES (ID_DIAGNOSA_SEQ.NEXTVAL, '04/25/1999', 'MATA MERAH', 'PSN003', 'ID1001', 883);
76 INSERT INTO DIAGNOSA VALUES (ID_DIAGNOSA_SEQ.NEXTVAL, '05/25/1999', 'DAYA TAHAN TURUN MENURUN', 'PSN004', 'ID1005', 884);

```

Figure 8: MENYISIPKAN DATA

Mengapa saya tidak menyisipkan data pada tabel LOG_PASIEN? Karena tabel ini saya gunakan untuk penyimpanan data secara otomatis jika proses *trigger* atau *delete, update, insert* dilakukan langsung tercatat pada tabel LOG_PASIEN.

- Setelah semua data telah disisipkan pada setiap tabel, selanjutnya kita akan membuat *trigger* untuk membuat sebuah skema agar kita mengetahui setiap data yang di *insert, update* maupun *delete* pada tabel yang sudah ditentukan atau ditandai dengan *trigger* itu sendiri.

```

83 CREATE OR REPLACE TRIGGER tg_delete_pasien
84 AFTER DELETE ON PASIEN
85 FOR EACH ROW
86 BEGIN
87     INSERT INTO LOG_PASIEN VALUES (
88         :OLD.ID_PASIEN,
89         :OLD.NAMA_PASIEN,
90         CURRENT_TIMESTAMP,
91         'DELETED'
92     );
93 END;
94
95 CREATE OR REPLACE TRIGGER tg_insert_pasien
96 AFTER INSERT ON PASIEN
97 FOR EACH ROW
98 BEGIN
99     INSERT INTO LOG_PASIEN VALUES (
100         :NEW.ID_PASIEN,
101         :NEW.NAMA_PASIEN,
102         CURRENT_TIMESTAMP,
103         'ADDED'
104     );
105 END;

```

Figure 9: MEMBUAT TRIGGER

```

107 CREATE OR REPLACE TRIGGER tg_update_pasien
108 AFTER UPDATE ON PASIEN
109 FOR EACH ROW
110 BEGIN
111     INSERT INTO LOG_PASIEN VALUES (
112         :NEW.ID_PASIEN,
113         :NEW.NAMA_PASIEN,
114         CURRENT_TIMESTAMP,
115         'UPDATED'
116     );
117 END;

```

Figure 10: MEMBUAT TRIGGER

Cara mengecek *trigger* yang telah dibuat:

Pilih *Object Browser*

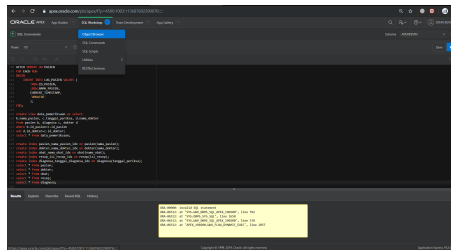


Figure 11: OBJECT BROWSER

Lalu pilih *trigger*

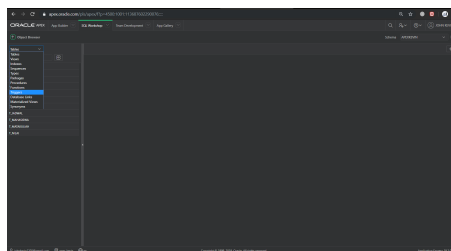


Figure 12: TRIGGER

7. Selanjutnya kita akan membuat *view* seperti yang kita tahu, *view* digunakan untuk membuat sebuah tabel *virtual* (bukan tabel sebenarnya) yang dibuat dari beberapa tabel lain. *SQL View* tidak memiliki data sendiri, tetapi data-datanya berasal dari tabel-tabel atau yang telah berkorelasi. *View* digunakan untuk memudahkan atau menyederhanakan data yang ditampilkan.

```

119 create view data_pemeriksaan as select
120 b.nama_pasien, c.tanggal_periksa, d.nama_dokter
121 from pasien b, diagnosa c, dokter d
122 where b.id_pasien=c.id_pasien
123 and d.id_dokter=c.id_dokter;
124 select * from data_pemeriksaan;
125

```

Figure 13: MEMBUAT VIEW

Cara melihat menggunakan *query* pada *create view* yang telah dibuat: //

```

124 select * from data_pemeriksaan;
125

```

Figure 14: MELIHAT VIEW

2 CREATE APLICATION

Setelah proses query dalam pembuatan tabel dan penyisipan data selesai kita akan menuju ke pembuatan aplikasi. Langkah-langkah pembuatan aplikasi adalah sebagai berikut:

1. Pertama kita pilih *Oracle Apex* (pada lingkaran_biru) untuk menuju ke halaman utama

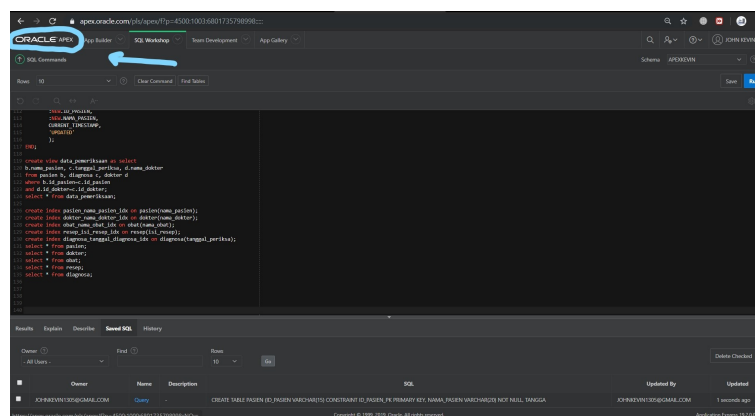


Figure 15: MENUJU KE HALAMAN UTAMA

2. Lalu pilih *App Builder*

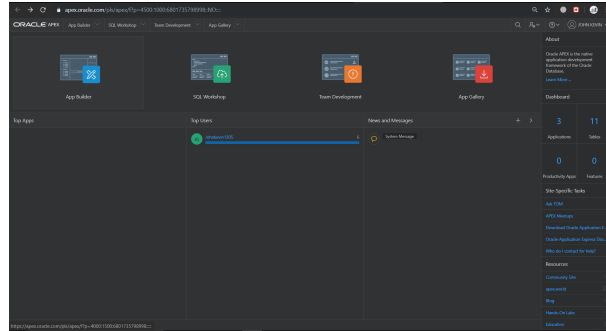


Figure 16: APP BUILDER

3. Selanjutnya pilih *Create*

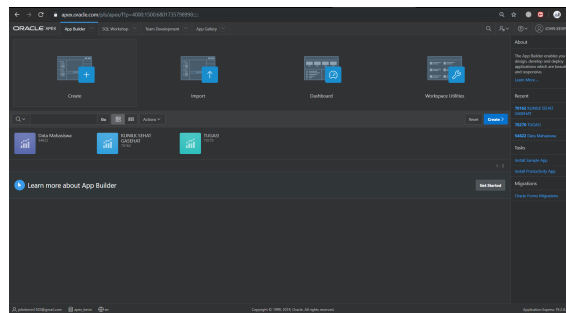


Figure 17: CREATE

4. Setelah itu pilih *New Application*

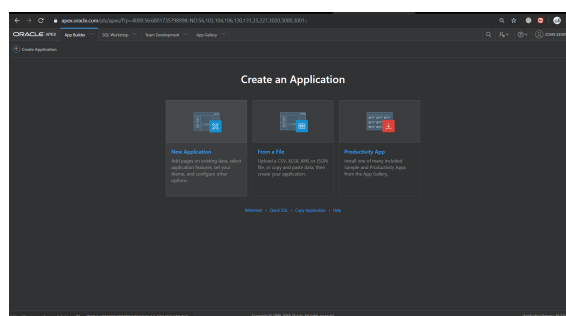


Figure 18: NEW APPLICATION

5. Silahkan isi nama aplikasi yang diinginkan dan tampilan pada halaman tabel ingin seperti apa

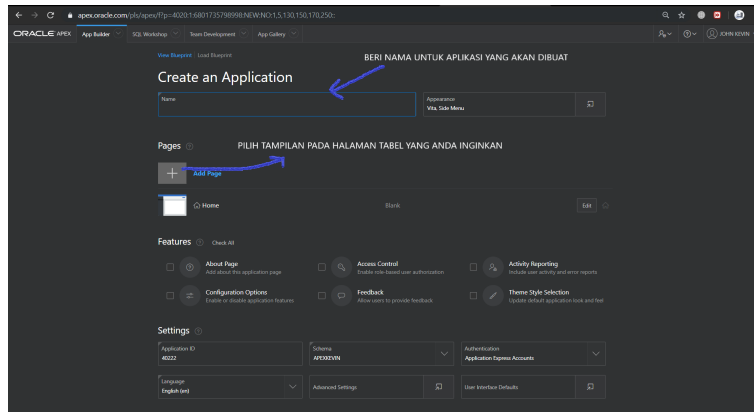


Figure 19: ISI NAMA DAN BENTUK TAMPILAN

- Setelah nama aplikasi dan tampilan tabel telah anda tentukan selanjutnya silahkan pilih *Create Application*

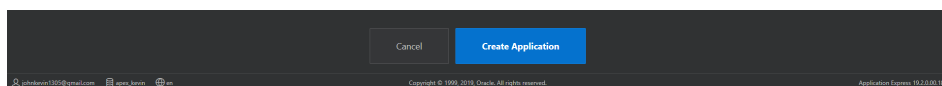


Figure 20: CREATE APPLICATION

- Selanjutnya setelah aplikasi berhasil dibuat, anda dapat menjalankannya dengan cara memilih *Run Application*

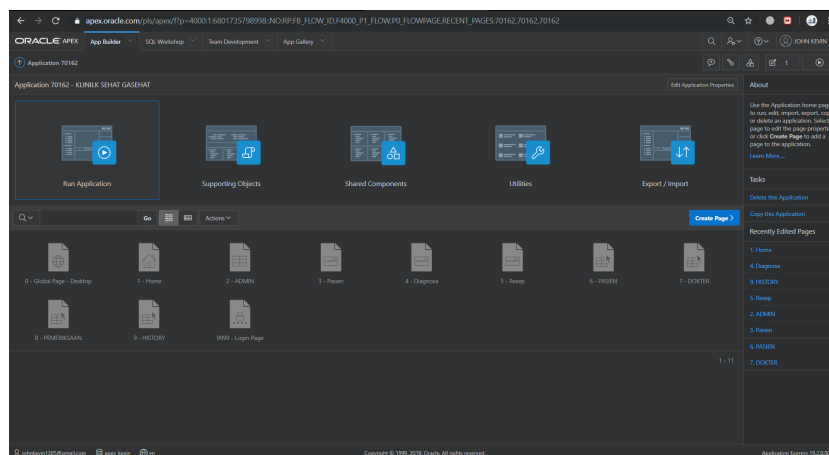


Figure 21: MENUJU KE HALAMAN UTAMA

- Setelah anda memilih *Run Application*, akan muncul halaman *login* untuk mengakses

aplikasi yang telah dibuat. Anda dapat menggunakan *Username* dan *Password* pada saat anda masuk pada aplikasi *Oracle Apex* anda.

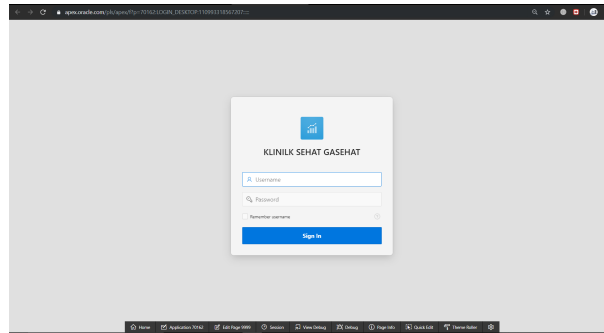


Figure 22: LOGIN APLIKASI

9. Setelah anda berhasil *Login*, anda akan berada pada halaman utama aplikasi yang telah anda buat

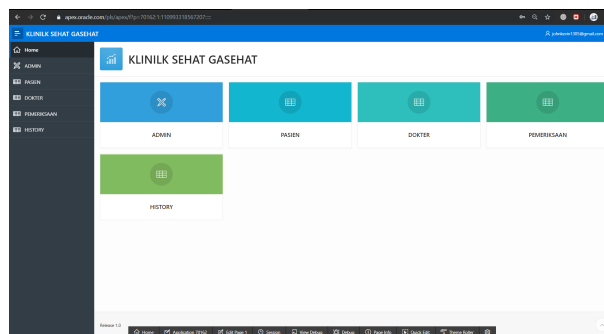


Figure 23: HALAMAN UTAMA APLIKASI