pISSN: 2442-3386 eISSN: 2442-4293
Vol 6 No 1 Januari 2020, 55 - 62
Melek IT
INFORMATION TECHNOLOGY JOURNAL

PERANCANGAN ACTIVE DATABASE DALAM SISTEM DATABASE TOKO *ONLINE* PROGRAM STUDI TEKNIK INFORMATIKA UWKS

Alffit Nadhifatulloh Ichtiarto[1], Muhammad Ilham Akbar[2],Yudha Dwi Cahyo[2]
Fakultas Teknik Program Studi Informatika, Universitas Wijaya Kusuma Surabaya
Jalan Dukuh Kupang nomer 52 Surabaya 60225
eMail : info@uwks.ac.id

Abstrak

Toko *online* merupakam sarana dimana pembeli dan penjual melakukan transaksi, dalam toko *online* tidak lepas dari yang namanya database sebagai tempat menyimpan data dari trasaksi antara penjual dan pembeli. Sistem database merupakan sistem perangkat lunak yang dirancang untuk mengelola dan menjalankan operasi terhadap data yang diminta oleh pengguna, untuk dapat mengelola data dalam database sistem memerlukan adanya active database sehingga database dapat berjalan sebagaimana yang di perlukan untuk pengguna. Dalam sistem database toko *online* diperlukan adanya active database untuk menggabungkan antara tabel satu dengan lainnya sehingga jika pelanggan melakukan transaksi maka sistem dapat mengelola permintaan pelanggan dalam memilih barang sampai dengan mengirimkan barang jke alamat pelanggan.

Kata Kunci: Toko *online* , database dalam toko *online* ,active database.

Abstract

Online Shop is a facility where buyers and sellers make transactions, in an online store that is not separated from the name of the database as a place to store data from transactions between sellers and buyers. A database system is a software system designed to manage and carry out operations on data requested by users, to be able to manage data in a database system requires an active database so that the database can run as needed for the user. In an online store database system, an active database is needed to combine one table with another so that if a customer makes a transaction, the system can manage customer requests in selecting goods to send goods to the customer's address.
Keywords: *Online shop, database in online shop, active database.*

I. PENDAHULUAN

Semakin berkembangnya zaman, semakin berkembang pula teknologi yang ada. Teknologi mengalami perkembangan dari waktu ke waktu dan tingkat kecanggihannya pun semakin tinggi. Tidak heran mengapa saat ini sangat banyak teknologi-teknologi yang canggih bermunculan. Teknologi sering dikaitkan dengan internet. Internet merupakan bagian dari teknologi informasi dan komunikasi. Teknologi diciptakan bertujuan untuk membantu dan memberikan kemudahan dalam berbagai aspek kehidupan, baik pada saat manusia bekerja, berkomunikasi, bahkan untuk mengatasi berbagai permasalahan dan persoalan yang ada di masyarakat.

Toko *online* merupakan salah satu dampak dari berkembangnya teknologi, dimana dalam toko *online* masyarakat dapat berineraksi sebagai penjual dan pembeli tanpa harus bertatap muka secara langsung.

Menurut Didit Agus Irwantoko, belanja *online* (*online shop*) merupakan proses pembelian barang/jasa oleh konsumen ke penjual realtime, tanpa pelayan, dan melalui internet. Toko virtual ini mengubah paradigma proses membeli barang/jasa dibatasi oleh tembok pengecer, atau mall[5].Maksutnya, tak perlu harus bertemu penjual/pembeli secara langsung, tak perlu menemukan wujud ‘pasar’ secara fisik, namun hanya dengan menghadap layar monitor computer, dengan koneksi internet tersambung, kita dapat melakukan transaksi jual/beli secara cepat dan nyaman. Untuk dapat menciptakan toko *online* tersebut diperlukan sistem untuk mengelola data data dari penjualan *owner* atau pemilik toko. Sistem adalah sekelompok unsur yang erat hubungannya satu dengan yang lain, yang berfungsi bersama-sama untuk mencapai tujuan tertentu.[10]. Sistem yang dimaksud adalah sistem database yaitu sekumpulan dari data data yang disusun dan digunakan sebagai acuan untuk mendapatkan hasil yang sesuai dengan pengguna, menurut Kristanto pada tahun 1994 database merupakan kumpulan file-file yang saling berelasi, relasi tersebut ditunjukkan dengan kunci dari tiap

file yang ada untuk digunakan dalam satu lingkup perusahaan, instansi.[6]

II. METODE

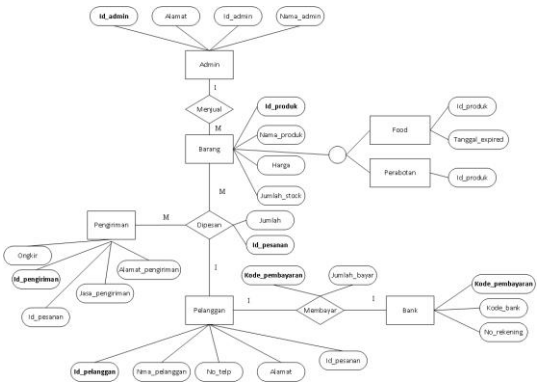
Metode penelitian yang digunakan dalam perancangan active database adalah parameter *Driven Approach*. Dalam pendekatan ini aturan-aturan disimpan dalam sebuah perangkat lunak basisdata. Aturan-aturan tersebut dikenali dengan sejumlah atribut atau nilai berupa parameter sehingga mampu berkomunikasi dengan aplikasi front-end yang menggunakan aturan. Aplikasi akan mengirimkan nilai-nilai berupa parameter yang telah didefinisikan di dalam aturan yang tersimpan di dalam database. Sehingga melalui parameter-parameter inilah aplikasi dan basisdata saling berkomunikasi untuk bertukar data dan informasi, menurut Dayal (1997, h. 1) pada.[1]

Paradigma aturan produksi pada *Active Database Management System (ADBMS)* mengikuti pola aturan produksi pada *Artifficial Intelligence* (AI) dengan aturan menyerupai aturan produksi pada sistem pakar. Salah satu fitur yang terdapat dalam *Active Database System* adalah adanya mekanisme pendefinsian:

Event – Condition – Action (ECA) Rule. Event – Condition – Action (ECA) adalah sebuah cara yang digunakan untuk menangkap perilaku dinamis dalam sebuah sistem informasi. Paradigma ECA telah memberikan dampak yang siginifikan di bidang Sistem Informasi dan telah digunakan dalam Active Database baik secara konseptual maupun dalam implementasinya.[1]

1. *EERD (Enhanced entity Relationship Diagram)*

   EERD*(Enhanced Entity Relationship Diagram)* merupakan model seluruh konsep model ER ditambah konsep-konsep dari subclass dan superclass, dan konsep-konsep yang berhubungan yaitu *specialization* dan *generalization.* Superclass adalah sebuah istilah yang merujuk pada kelas yang dijadikan acuan pewarisan/penurunan data.`Oleh karena itu, superclass bersifat umum.Sementara itu, subclass adalah sebuah istilah yang merujuk pada kelas yang diturunkan atau diwariskan dari superclass. Oleh karena itu, subclass bersifat lebih khusus atau spesifik dalam penyajian informasi dari pada *superclass*.[9]

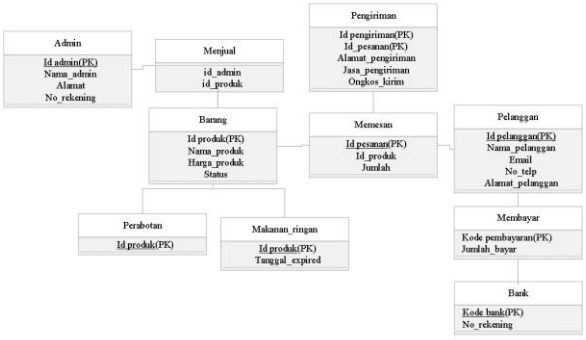


Gambar 1.1 *EERD* dari toko *online*

Gambar 1.1 merupakan EERD berisi seluruh konsep ERD ditambah dengan konsep konsep yang berhubungan yaitu *specialization* dan *generalization*, model EERD menekankan pada superclass/subclass *relationship* yang merupakan hubungan antara superclass dan subclassnya.Dari gambar di atas diketahui Barang sebagai super class dan perabotan, makanan sebagai sub class.

2. Relational DataBase Management System (*RDBMS)*

   RDBMS adalah program yang melayani sistem basisdata yang entitas utamanya terdiri dari tabel-tabel yang mempunyai relasi dari satu tabel ketabel yang lain.[7]., *Foreign Key* adalah satu set atribut atau set atribut sebagai key penghubung kedua tabel dan melengkapi satu relationship (hubungan) terhadap primary key yang menunjukan keinduknya.[8]



Gambar 1.2 Rancangan *RDBMS* Toko *Online*

Dari gambar *RDBMS* 1.2 yaitu merupakan hubungan relasi antar tabel pada sistem toko *online*. Dalam RDBMS di atas membentuk relasi yaitu pertama tabel admin berelasi dengan tabel barang dengan relasi menjual dalam tabel admin terdapat id_admin sebagai *primary key* serta dalam tabel barang id_produk sebagai *primary key*, tabel barang dipecah menjadi 2 subclass yaitu perabotan

pISSN: 2442-3386 eISSN: 2442-4293
Vol 6 No 1 Januari 2020, 57 - 62
Melek IT
INFORMATION TECHNOLOGY JOURNAL

dan makanan_ringan, selanjutnya relasi tabel barang dengan pelanggan, pelaggan bertindak sebagai pemesan sehingga terbentuk tabel relasi memesan dengan *primary key* id_produk dari tabel barang serta planggan memiliki id_pelanggan sebagai *primary key*, selanjutnya adalah relasi pelanggan dengan bank denagan relasi membayar dan memiliki kode_pembayaran sebagai *primary key*, selanjutnya adalah pengiriman dengan barang dengan relasi mengirim yang memiliki *primary key* id_pesanan.

3. Active Database

Dalam *active database* terdapat 3 macam perintah yang digunakan yaitu:

1. *Procedure* adalah suatu program yang terpisah dalam blok sendiri yang berfungsi sebagai seubprogram (program bagian). Prosedur diawali dengan kata cadangan *PROCEDURE* di dalam bagian deklarasi prosedur. Prosedur dipanggil dan digunakan di dalam blok program lainnya dengan menyebutkan judul prosedurnya.[3]
2. *Function* adalah suatu blok PL/SQL yang memiliki konsep sama dengan *procedure*, hanya saja pada function terdapat pengembalian nilai (return value).Karena *function* dapat mengembalikan sebuah nilai, *function* dapat diakses seperti layaknya sebuah variabel biasa.[3]
3. *Trigger* adalah sebuah mekanisme kerja yang di panggil ketika ada sebuah aksi yang terjadi pada sebuah table. Penamaan trigger tidak boleh melebihi 128 karakter. Aksi yang di kenali oleh trigger dapat berupa statement DML biasa seperti *INSERT*, *UPDATE*, dan *DELETE* atau statement DDL. Biasanya yang di eksekusi oleh trigger adalah *stored procedure* atau *batch*.[2]

Untuk dapat membuat database maka diperlukan alat atau aplikasi seperti SQL atau oracle. Dalam membuat sistem database toko *online* kami mengunakan aplikasi oracle Oracle adalah relational database management system (RDBMS) untuk mengelola informasi secara terbuka, komprehensif dan terintegrasi. Oracle Server menyediakan solusi yang efisien dan efektif [6].Alasan kami menggunakan oracle adalah:

1. Dapat bekerja di lingkungan client/server
2. Menangani manajemen space dan basis data

   yang besar oleh karena itu oracle cocok untuk mendata barang yang masuk dan keluar.
3. Mendukung akses data secara simultan
4. Performansi pemrosesan transaksi yang tinggi
5. Menjamin ketersediaan yang terkontrol
6. Lingkungan yang terreplikasi

III. HASIL DAN PEMBAHASAN

Dalam bab ini disimpulkan hasil dari pembahasan yaitu procedure untuk mencari data, trigger untuk men*delete* dan *insert* data ,dan function untuk mendapatkan data dari tabel yang bersangkutan, sehingga didapat kesimpulan *planing* atau rencana yaitu:

1. Procedure Planing

1.1. Proceedure pesanan

Yang berfunsi untuk menampilkan data tentang pesanan pelanggan untuk dapat melakukan procedure pada tabel pelanggan dibutuhkan input yaitu id_pelanggan ,id_pesanan , alamat_pelanggan.Pada procedure ini bertujuan untuk menampilkan data pesanan berdasar id_pelanggan

1.2. Procedure jumlah_dipesan
Yang berfunsi untuk menampilkan data tentang jumlah pesanan pelanggan Input data yang diperlukan yaitu id_pesanan ,id_produk ,jumlah .Sehingga pada procedure jumlah_dipesan akan menampilkan data dari jumlah pesanan menurut id_pesanan

1.3. Procedure dta_admin
Berfunsi untuk menampilkan data tentang admin dan input yang digunakan data adalah id_admin **,**nama_admin, alamat, no_rekening. Sehingga pada procedure “dta_admin” mencari admin berdasar id_admin

1.4. Prosedure kirim
Kegunaan berfunsi untuk menampilkan data tentang pengiriaman pesanan pelanggan dan input yang digunakan adalah id_pengiriman ,id_pesanan , jasa_pengiriman, ongkos_kirim

Sehingga pada procedure “kirim” mencari pengiriman berdasar id_pengiriman

1.5. Procedure jumlah_bayar
Berfunsi untuk menampilkan data tentang jumlah yang harus di bayar pelanggan dan input yang digunakan adalah kode_pembayaran**,**jumlah_bayar, pelanggan Sehingga pada procedure “jumlah_bayar” mencari pembayaran berdasar id_pelanggan

1.6. Procedure Stock
Berfunsi untuk menampilkan data tentang kesediaan barang procedure ini menggunakan input idproduk , nama_produk , jumlah_stock. Sehingga pada procedure “stock” mencari barang berdasar id_barang

2. Function Planing

2.1. Function jumlah perabotan
Berfungsi untuk menampilkan jumlah perabotan. Input yang digunakan adalah id_produk sehingga jumlah perabotan akan ditampilkan

2.2. Function jumlah pesanan perabotan
Berfungsi untuk menampilkan jumlah perabotan. Dengan input yang di gunakan yaitu id_produk . Sehingga *output* akan menampilkan jumlah pesanan perabotan

2.3. Function jumlah pesanan berdasar alamat
Berfungsi untuk menampilkan jumlah pesanan. Pada function ini diperlukan input yaitu id_pesanan, alamat_pelanggan**.** Sehingga jumlah dari pesanan ditampilkan.

2.4. Function jumlah pengiriman berdasar alamat
Berfungsi untuk menampilkan jumlah pengiriman.Dalam sistem ini diperlukan input yaitu id dari pengiriman dan alamat_pelangga. Seingga jumlah pengriman dapat di tampilkan

2.5. Function kurir
Berfungsi untuk menampilkan jumlah kurir. Input yang di gunakan yaitu id_pengiriman, jasa_pengiriman Sehingga jumlah dari satu jasa kurir dapat di tampilkan

1.7. Function kode bank
Berfungsi untuk menampilkan jumlah dari kode bank. Input yang digunakan untuk function ini adalah kode_bank**.** Sehinggajumlah kode bank

3. Trigger Planing

3.1. Trigger delete pesanan.
Berguna untuk mendelete data dari pesanan pelanggan dengan input id_pesanan dan id_pelanggan .Sehingga output data dari pesanan yang terhapus.

3.2. Trigger delete data pengriman pelanggan.

Berguna menghapus data dari pengiriman Dengan input id_pengiriman**,** id_pesanan Sehingga tabel pengiriman yang kosong karena terhapus.

4. Implementasi Oracle dari active database

1. Procedure

1.1. Procedure pesanan

```
CREATE OR REPLACE PROCEDURE pemesan
is
nama_pelanggan varchar(20);
id_pesanan varchar(10);
alamat_pelanggan varchar(50);
id_pelanggan varchar(10);
BEGIN
select pelanggan.id_pelanggan,pelanggan.nama_pelanggan,
pelanggan.id_pesanan, pelanggan.alamat_pelanggan
into id_pelanggan,nama_pelanggan, id_pesanan,alamat_pelanggan
from pelanggan where pelanggan.id_pesanan='303';
DBMS_OUTPUT.PUT_LINE('Nama = ' || nama_pelanggan);
DBMS_OUTPUT.PUT_LINE('id_pesanan= ' || id_pesanan);
DBMS_OUTPUT.PUT_LINE('Alamat = '||alamat_pelanggan);
DBMS_OUTPUT.PUT_LINE('id_pelanggan = ' || id_pelanggan);
end pemesan;
```

Gambar 4.1.1 Merupakan query dari procedure pesanan

```
Task completed in 0,698 seconds
1 row inserted.


Procedure PEMESAN compiled

Nama = Sulis Berta
id_pesanan= 303
Alamat = Magetan
id_pelanggan = 1103


PL/SQL procedure successfully completed.
```

Gambar 4.1.2 Merupakan *Output* dari query procedure pesanan

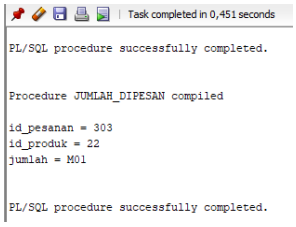
1.2. Procedure jumlah dipesan

```
CREATE OR REPLACE PROCEDURE jumlah_dipesan
is
id_pesanan varchar(10);
id_produk varchar(10);
jumlah varchar(50);
BEGIN
select memesan.id_pesanan,
memesan.jumlah,memesan.id_produk
into id_pesanan,id_produk,jumlah
from memesan where memesan.id_pesanan='303';
DBMS_OUTPUT.PUT_LINE('id_pesanan = ' || id_pesanan);
DBMS_OUTPUT.PUT_LINE('id_produk = ' || id_produk);
DBMS_OUTPUT.PUT_LINE('jumlah = ' || jumlah);
end jumlah_dipesan;

set SERVEROUTPUT on
EXECUTE jumlah_dipesan;
```

Gambar 4.2.1 Merupakan query dari jumlah dipesan

pISSN: 2442-3386 eISSN: 2442-4293
Vol 6 No 1 Januari 2020, 59 - 62
Melek IT
INFORMATION TECHNOLOGY JOURNAL



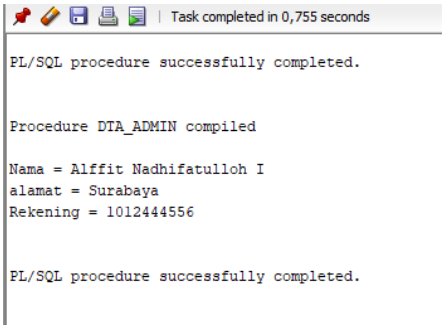
Gambar 4.2.2 *Output* dari *query* jumlah

1.3. Procedure dta admin

```
CREATE OR REPLACE PROCEDURE dta_admin
is
id_admin varchar(10);
nama_admin varchar(50);
alamat varchar(50);
no_rekening varchar(50);
BEGIN
select admin.id_admin, admin.nama_admin, admin.alamat,
admin.no_rekeninginto id_admin, nama_admin,alamat,no_rekening
from admin where admin.id_admin='101';
DBMS_OUTPUT.PUT_LINE('Nama = ' || nama_admin);
DBMS_OUTPUT.PUT_LINE('alamat = ' || alamat);
DBMS_OUTPUT.PUT_LINE('Rekening = '||  no_rekening);
end dta_admin;

set SERVEROUTPUT on
EXECUTE dta_admin;
```

Gambar 4.3.1 merupakan *query* dari data admin



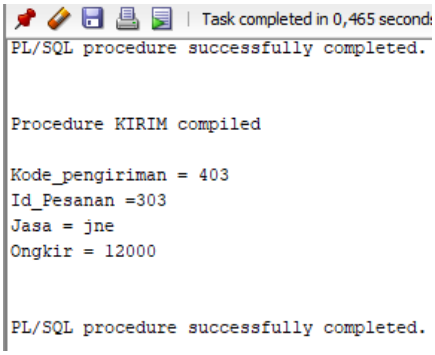
Gambar 4.3.2 merupakan *Output* dari data admin

1.4. Procedure kirim

```
CREATE OR REPLACE PROCEDURE kirim
is
id_pengiriman varchar(10);
id_pesanan varchar(10);
jasa_pengiriman varchar(10);
ongkos_kirim varchar(50);
BEGIN
select pengiriman.id_pengiriman, pengiriman.id_pesanan,
pengiriman.jasa_pengiriman, pengiriman.ongkos_kirim
into id_pengiriman,id_pesanan, jasa_pengiriman, ongkos_kirim
from pengiriman where pengiriman.id_pesanan='303';
DBMS_OUTPUT.PUT_LINE('Kode_pengiriman = ' || id_pengiriman);
DBMS_OUTPUT.PUT_LINE('Id_Pesanan ='||id_pesanan);
DBMS_OUTPUT.PUT_LINE('Jasa = ' || jasa_pengiriman);
DBMS_OUTPUT.PUT_LINE('Ongkir = ' || ongkos_kirim);
end kirim;

set SERVEROUTPUT on
EXECUTE kirim;
```

Gambar 4.4.1 merupakan *query* dari kirim



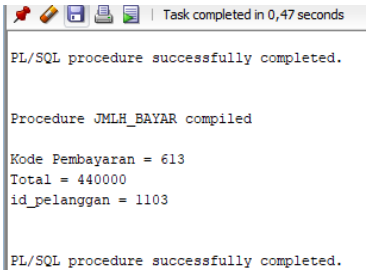
Gambar 4.4.2 merupakan *output* dari *query* data admin

1.5. Procedure jumlah_bayar

```
CREATE OR REPLACE PROCEDURE jmlh_bayar
is
kode_pembayaran varchar(10);
id_pelanggan varchar(10);
jumlah_bayar varchar(20);
BEGIN
select membayar.kode_pembayaran ,
membayar.id_pelanggan, membayar.jumlah_bayar
into kode_pembayaran,id_pelanggan, jumlah_bayar
from membayar where membayar.id_pelanggan='1103';
DBMS_OUTPUT.PUT_LINE('Kode Pembayaran = ' || kode_pembayaran);
DBMS_OUTPUT.PUT_LINE('Total = ' || jumlah_bayar);
DBMS_OUTPUT.PUT_LINE('id_pelanggan = ' || id_pelanggan);
end jmlh_bayar;

set SERVEROUTPUT on
EXECUTE jmlh_bayar;
```

Gambar 4.5.1 merupakan *query* dari jumlah_bayar



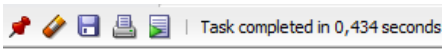
Gambar 4.5.2 merupakan *output query* dari jumlah_bayar

1.6. Procedure Stock

```
CREATE OR REPLACE PROCEDURE stock
is
id_produk varchar(10);
nama_produk varchar(50);
jumlah_stock varchar(50);
BEGIN
select barang.id_produk ,
barang.nama_produk, barang.Jumlah_stock
into id_produk,nama_produk,Jumlah_stock
from barang where barang.id_produk='M05';
DBMS_OUTPUT.PUT_LINE('Nama Barang = ' || nama_produk);
DBMS_OUTPUT.PUT_LINE('id_barang = ' || id_produk);
DBMS_OUTPUT.PUT_LINE('Stock = ' ||Jumlah_stock);
end stock;

set SERVEROUTPUT on
EXECUTE stock;
```

Gambar 4.6.1 merupakan *query* dari stock




```
Procedure STOCK compiled

Nama Barang = Susu kental per sachet
id_barang = M05
Stock = 110


PL/SQL procedure successfully completed.
```

Gambar 4.6.2 merupakan *output query* dari stock

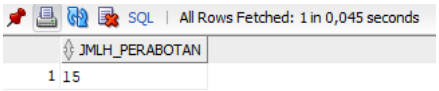
2. Function

2.1. Function jumlah perabotan

```
create or replace function jmlh_perabotan
return varchar
is
id_produk varchar (10);
begin
select count(id_produk)into id_produk
from barang where id_produk Like 'P%';
return id_produk;
end jmlh_perabotan;

select jmlh_perabotan from dual
```

Gambar 4.7.1 merupakan *query* dari function jumlah_perabotan



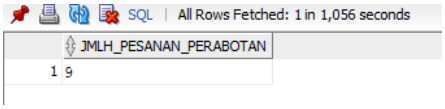
Gambar 4.7.2 merupakan *Output query* dari function jumlah_perabotan

2.2. Function jumlah pesanan perabotan

```
create or replace function jmlh_pesanan_perabotan
return varchar
is
id_pesanan varchar (10);
begin
select count(id_produk)into id_pesanan
from memesan where id_produk like 'P%';
return id_pesanan;
end jmlh_pesanan_perabotan;

select jmlh_pesanan_perabotan from dual
```

Gambar 4.8.1 merupakan *query* dari *function* jumlah_pesanan_perabotan

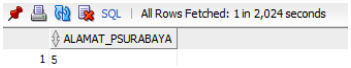


2.3. Function jumlah pesanan berdasar alamat

```
create or replace function alamat_psurabaya
return varchar
is
id_pelanggan varchar (10);
begin
select count(id_pelanggan)into id_pelanggan
from pelanggan where alamat_pelanggan = 'Surabaya';
return id_pelanggan;
end alamat_psurabaya;

select alamat_psurabaya from dual
```

Gambar 4.8.1 merupakan *query* dari *function* alamat_pesanan surabaya



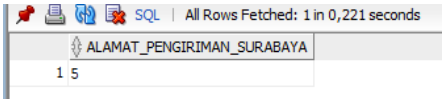
Gambar 4.8.2 merupakan *Output query* dari function jumlah_perabotan

2.4. Function jumlah pengiriman berdasar alamat

```
create or replace function alamat_pengiriman_surabaya
return varchar
is
id_pengiriman varchar (50);
begin
select count(alamat_pelanggan)
into id_pengiriman from pengiriman
where alamat_pelanggan = 'Surabaya';
return id_pengiriman;
end alamat_pengiriman_surabaya;

select alamat_pengiriman_surabaya from dual
```

Gambar 4.9.1 merupakan *query* dari function alamat_pengiriman_surabaya



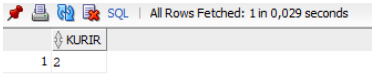
Gambar 4.9.2 merupakan *Output query* dari function alamat pengiriman Surabaya

2.5. Function kurir

```
create or replace function kurir
return varchar
is
id_pengiriman varchar (10);
begin
select count(jasa_pengiriman)
into id_pengiriman from pengiriman
where jasa_pengiriman = 'pos indonesia';
return id_pengiriman;
end kurir;

select kurir from dual
```

Gambar 4.10.2 merupakan *query* dari function kurir.



Gambar 4.10.2 merupakan *Output query* dari function kurir

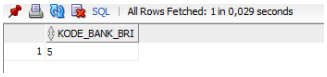
2.6. Function kode bank

pISSN: 2442-3386 eISSN: 2442-4293
Vol 6 No 1 Januari 2020, 61 - 62
Melek IT
INFORMATION TECHNOLOGY JOURNAL

```
create or replace function kode_bank_BRI
return varchar
is
id_pembayaran varchar (10);
begin
select count(kode_bank)into id_pembayaran
from bank where kode_bank = '002';
return id_pembayaran;
end kode_bank_BRI;

select kode_bank_BRI from dual
```

Gambar 4.11.1 merupakan *Output query* dari function jumlah_perabotan



Gambar 4.11.2 merupakan *Output query* dari function kode bank

3. Trigger

3.1. Trigger *delete* pesanan

```
create or replace trigger delete_pesanan
before delete on memesan
for each row
begin
dbms_output.put_line('DATA TERHAPUS');
delete from pelanggan where id_pesanan = '301';
end;
set serveroutput on
delete from memesan where id_pesanan = '301';
```

Gambar 4.12.1 merupakan *query* dari *trigger delete_pesanan*

```
Task completed in 0,752 seconds

Trigger DELETE_PESANAN compiled

DATA TERHAPUS


1 row deleted.
```

Gambar 4.12.2 merupakan *Output query* dari *trigger delete_pesanan*

3.2. Trigger *delete* pengiriman

```
select*from memesan

create or replace trigger delete_pengiriman
before delete on pengiriman
for each row
begin
dbms_output.put_line('DATA TERHAPUS');
delete from memesan where id_pesanan = '301';
end;
set serveroutput on
delete from pengiriman where id_pesanan = '301';

SELECT * FROM pengiriman
```

Gambar 4.12.1 merupakan *Output query* dari *trigger delete_pengiriman*

```
Task completed in 0,492 seconds

Trigger DELETE_PENGIRIMAN compiled

DATA TERHAPUS


1 row deleted.
```

Gambar 4.12.1 merupakan *Output query* dari *trigger delete_pengiriman*

4. Trigger

4.1. Trigger *delete* pesanan

```
create or replace trigger delete_pesanan
before delete on memesan
for each row
begin
dbms_output.put_line('DATA TERHAPUS');
delete from pelanggan where id_pesanan = '301';
end;
set serveroutput on
delete from memesan where id_pesanan = '301';
```

Gambar 4.12.1 merupakan *query* dari *trigger delete_pesanan*

```
Task completed in 0,752 seconds

Trigger DELETE_PESANAN compiled

DATA TERHAPUS


1 row deleted.
```

Gambar 4.12.2 merupakan *Output query* dari *trigger delete_pesanan*

4.2. Trigger *delete* pengiriman

```
select*from memesan

create or replace trigger delete_pengiriman
before delete on pengiriman
for each row
begin
dbms_output.put_line('DATA TERHAPUS');
delete from memesan where id_pesanan = '301';
end;
set serveroutput on
delete from pengiriman where id_pesanan = '301';

SELECT * FROM pengiriman
```

Gambar 4.12.1 merupakan *Output query* dari *trigger delete_pengiriman*

```
Task completed in 0,492 seconds

Trigger DELETE_PENGIRIMAN compiled

DATA TERHAPUS


1 row deleted.
```

Gambar 4.12.1 merupakan *Output query* dari *trigger delete_pengiriman*

5. Singkatan dan Akronim

EERD (Enhanced entity Relationship Diagram) merupakan model seluruh konsep model ER ditambah konsep-konsep dari subclass dan

6. Singkatan dan Akronim

1. *EERD (Enhanced entity Relationship Diagram)* merupakan model seluruh konsep model ER ditambah konsep-konsep dari subclass dan superclass, dan konsep-konsep yang berhubungan yaitu specialization dan generalization.[4].
2. RDBMS (*relational database managemnent system*) di gunakan pada sistem perancangan hijab ukhti special sebelum eerd yang melayani sistem basis data yang entitas utamanya terdiri dari tabel-tabel yang mempunyai relasi dari satu tabel ke tabel yang lain.[6]

IV. PENUTUP

4.1. Kesimpulan

Pada jurnal sistem database toko *online* yang kmmi kerjakan dapat kami simpulkan bahwa active database merpakan komponen sistem basis data yang dapat melakukan pemrosesan data dan memiliki kemampuan untuk mendeteksi ketika data di ubah, dihapus atau ditambahkan dapat secara otomatis dalam tabel. Sehingga pencarian informasi pada database lebih mudah.

DAFTAR PUSTAKA

[1] Amin, M. M. (2013). Pendekatan Active Database System Dan Business Rule Dalam Pengembangan Sistem Informasi. *Jurnal Informatika Darmajaya*, *13*(1), 52–62. https://doi.org/10.30873/ji.v13i1.125

[2] Haryanto,Rudy.Pengertian View, Stored Procedure, Triger dan Function..2016. Http://rudilengkong.blogspot.com/2016/08/pengertian-view-stored-procedure-triger.html

[3] Mauritz, I. (n.d.). FUNCTION. *Gunadarma*. Retrieved from http://ivan_maurits.staff.gunadarma.ac.id/Downloads/files/5636/4+-+Prosedure+dan+Function.pdf

[4] TutorialRide.com. (2017). *Enhanced Entity Relationship Model (EER Model)*.1–4.

[5] Irwantoko, Didit.A. (n.d.). *Online* shopping, belanja *online*, amankah? Retrieved July 2, 2012, from http://nevafarrell.blogspot.com/2011/07/*online*-shopping-belanja-*online*-amankah.html

[6] Irawan, Muhammad. R. (2015).Materi lengap *database from* http://learningrplit.blogspot.com /2015/09/materi-lengkap-tentang-database.html

[7] Pratama,Wahyu (n.d.) MateriPerkuliahan-RDBMS.pdf

[8] Agus T.(2015). Belajar C perbedaan candidate key, primary key, dan foreign key

[9] TutorialRide.com. (2017). *Enhanced Entity Relationship Model (EER Model)*.1–4.

[10] Fauzi, R. A. (2017). *Sistem Informasi.*