

Structured Query Language (SQL) Untuk Purchase Order (PO) Menggunakan SQL Server 2008

Didik Setiyadi ^{1,*}, Herlawati ²

¹ Teknik Informatika; STMIK Bina Insani; Jl. Siliwangi No.6 Rawa Panjang Bekasi Bekasi Timur 17114 Indonesia; Telp. (021) 824 36 886 / (021) 824 36 996. Fax. (021) 824 009 24; e-mail: didiksetiyadi@binainsani.ac.id

² Sistem Informasi; STMIK Bina Insani; Jl. Siliwangi No.6 Rawa Panjang Bekasi Bekasi Timur 17114 Indonesia; Telp. (021) 824 36 886 / (021) 824 36 996. Fax. (021) 824 009 24; e-mail: herlawati@binainsani.ac.id

* Korespondensi: e-mail: didiksetiyadi@binainsani.ac.id

Diterima: 25 Maret 2019; Review: 24 Mei 2019; Disetujui: 14 Juni 2019

Cara sitasi: Setiyadi D, Herlawati. 2019. *Structured Query Language (SQL) untuk Purchase Order (PO) menggunakan SQL Server 2008*. Bina Insani ICT Journal. 6 (1): 75 – 88.

Abstrak: *Structured Query Language (SQL)* merupakan bahasa pemrograman khusus yang dimanfaatkan dalam pengelolaan data pada *Relational Database Management System (RDBMS)*. SQL terdiri dari sintaks sederhana dalam bentuk instruksi-instruksi dalam melakukan manipulasi data, instruksi tersebut sering disebut dengan *query*. Bahasa *query* tersebut digunakan untuk mengimplementasikan basis data *Purchase Order (PO)* dengan menggunakan basis data *SQL Server 2008*. Langkah awal yang dilakukan sebelum mengimplementasi dengan *query* ke *SQL Server 2008* adalah dengan mengetahui bagaimana bentuk *Entity Relationship Diagram (ERD)* dari *database Purchase Order (PO)* beserta struktur dari tabel-tabel yang ada dalam *database* tersebut. Implementasi awal yang dilakukan adalah dengan membuat (*create*) *database* sebagai lokasi untuk menyimpan tabel-tabel yang ada dengan menggunakan *sintaks query*. Setelah *database* terbentuk, langkah selanjutnya adalah membuat (*create*) tabel-tabel yang telah ada, melakukan perekaman data (*insert data*), membuat relasi (*relation*) dan langkah terakhir adalah membuat *database diagram*. Perangkat lunak yang digunakan dalam mengimplementasikan *database Purchase Order (PO)* menggunakan *RDMS SQL Server 2008*.

Kata Kunci: *SQL, Query, Sql Server 2008, Entity Relationship Diagram, Purchase Order.*

Abstract: *Structured Query Language (SQL)* is a special programming language that is used in data management in the *Relational Database Management System (RDBMS)*. SQL consists of simple syntax in the form of instructions for manipulating data, these instructions are often called queries. The query language is used to implement a *Purchase Order (PO)* database using *SQL Server 2008* database. The first step before implementing a query to *SQL Server 2008* is to find out how the *Entity Relationship Diagram (ERD)* forms from a *database Purchase Order (PO)* along with the structure of the tables in the database. The initial implementation is done by creating (*create*) a database as a location to store existing tables using query syntax. After the database is formed, the next step is to create (*create*) existing tables, do data recording (*insert data*), create relations (*relations*) and the last step is to create a *database diagram*. Software used in implementing *database Purchase Orders (PO)* using *RDMS SQL Server 2008*.

Keywords: *SQL, Query, Sql Server 2008, Entity Relationship Diagram, Purchase Order.*

1. Pendahuluan

Perkembangan teknologi informasi dan komunikasi (TIK) saat ini begitu cepat, semua bidang kegiatan bisnis pada dunia usaha dan industri (DUDI) membutuhkan sistem informasi

yang cepat dan akurat yang mampu untuk bersaing dalam kegiatan bisnis tersebut. Sistem informasi dibangun pada setiap DUDI pasti berbasis komputer dan terintegrasi dalam setiap proses kegiatannya, dimana aplikasi yang dijalankan baik berbasis *web* maupun *mobile* dengan basis data yang interaktif tersimpan dalam *database server*. Pada saat ini umumnya perusahaan menggunakan *database* dalam penyimpanan data terdiri dari *Sql Server, MySql, Oracle, SqlLite* dan *database* lainnya sesuai dengan *requirement* dari perusahaan yang menggunakannya.

Basis data dapat dipahami suatu kumpulan data terhubung (*interrelated data*) yang disimpan secara bersama-sama pada suatu media, tanpa mengatap satu sama lain atau tidak perlu suatu kerangkapan data (kalaupun ada maka kerangkapan data tersebut harus seminimal mungkin dan terkontrol). Data disimpan dengan cara-cara tertentu sehingga mudah digunakan atau ditampilkan kembali. Data bisa digunakan satu atau lebih program-program aplikasi secara optimal dan disimpan tanpa mengalami ketergantungan dengan program yang akan menggunakannya. Data disimpan sedemikian rupa sehingga proses penambahan, pengambilan, dan modifikasi data dapat dilakukan dengan mudah dan terkontrol [Sutanta, 2011].

Entity Relationship Model (ERM) merupakan suatu model data yang dikembangkan berdasarkan objek dan digunakan untuk menjelaskan hubungan antar data dalam basis kepada pengguna secara logik. ERM digambarkan dalam bentuk diagram yang disebut *Entity Relationship Diagram* (ERD) ERD digunakan dalam pemodelan sistem yang nantinya basis datanya akan dikembangkan serta membantu perancang/analisis sistem pada saat melakukan analisis dan perancangan basis data karena model ini dapat menunjukkan macam data yang dibutuhkan dan kerelasian antar data didalamnya [Sutanta, 2011].

Untuk menjalankan database relasional, maka kita harus mengenal perintah-perintah SQL (*Structured Query Language*). Perintah SQL merupakan perintah untuk mengolah, menampilkan dan memanipulasi data yang tersimpan dalam sebuah RDBMS. *Data Definistion Language* (DDL) adalah merupakan perintah untuk membuat atau mengubah struktur objek *database* seperti *create, alter, drop, truncate* dan *rename*. *Data Control Language* (DCL) merupakan perintah untuk mengatur hak akses *user* dalam *database*, seperti *grant* dan *revoke*. Sedangkan untuk *Data Manipulation Language* (DML) merupakan perintah untuk melakukan manipulasi data seperti menambah, menghapus, dan menubah data dalam *table*, misalkan *insert, delete, update* dan *merge* [Sari and Milwadhari, 2016].

SQL Server adalah *software* (perangkat lunak) RDBMS kelas *enterprise* yang banyak digunakan dalam dunia korporat. Dengan menggunakan *SQL Server*, *user* dapat menyimpan banyak data dan mengimplementasikannya untuk kepentingan bisnis dalam perusahaan [Wahana, 2013]. *Microsoft SQL Server* merupakan produk RDBMS (*Relational Database Management System*) yang dibuat oleh *Microsoft*. Orang sering menyebutnya dengan *SQL Server* saja. *Microsoft SQL Server* juga mendukung SQL sebagai bahasa untuk memproses *query* ke dalam *database*. *Microsoft SQL Server* banyak digunakan pada dunia bisnis, pendidikan atau juga pemerintahan sebagai solusi *database* atau penyimpanan data [Nuryana and Sulistiyono, 2014].

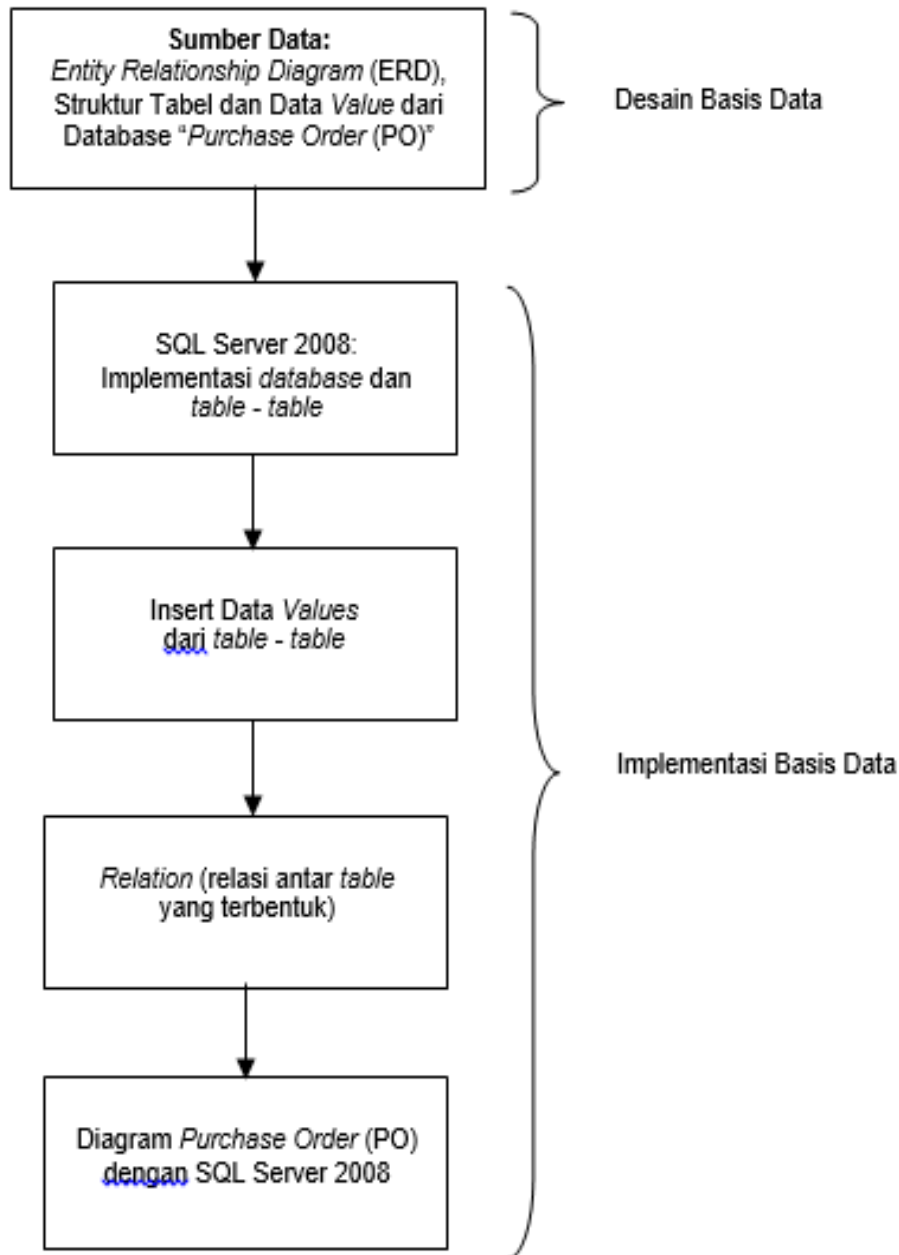
Berdasarkan penjelasan diatas, dalam penelitian ini akan dibahas tentang bagaimana menjalankan *Structured Query Language* (SQL) untuk *database Purchase Order* (PO) dengan menggunakan *SQL Server 2008*. Implementasi didasarkan pada ERD, struktur *table* dan data yang telah dibentuk pada penelitian sebelumnya. Implementasi dengan *query* pada *SQL Server 2008* yang dimulai dari *create database, create table, insert data*, relasi sampai dengan terbentuknya diagram PO.

2. Metode Penelitian

Metode penelitian yang digunakan adalah dengan melakukan teknik pengumpulan data dengan studi pustaka. Pada tahapan studi pustaka dilakukan dengan mengumpulkan data dari buku-buku dan jurnal yang berkaitan dengan masalah yang akan dibahas.

Metode yang digunakan dalam penelitian ini dengan menggunakan *Database System Development Life Cycle* (DSDLC) yang meliputi tahapan perencanaan basis data, definisi sistem, analisis dan pengumpulan kebutuhan, desain basis data, seleksi basis data, desain aplikasi, prototipe, implementasi, konversi data, pengujian dan pemeliharaan operasional [Indrajani, 2018]. Pada penelitian ini hanya dibatasi pada tahapan implementasi basis data dengan menggunakan *SQL Server 2008* dari tahapan pembuatan basis data, tabel - tabel, *insert data*, relasi sampai terbentuk diagram basis data.

Berikut ini adalah kerangka pemikiran yang merupakan langkah-langkah yang dilakukan dalam mengimplementasikan *database Purchase Order (PO)* yang dilakukan dengan menggunakan RDBMS *Sql Server 2008*.

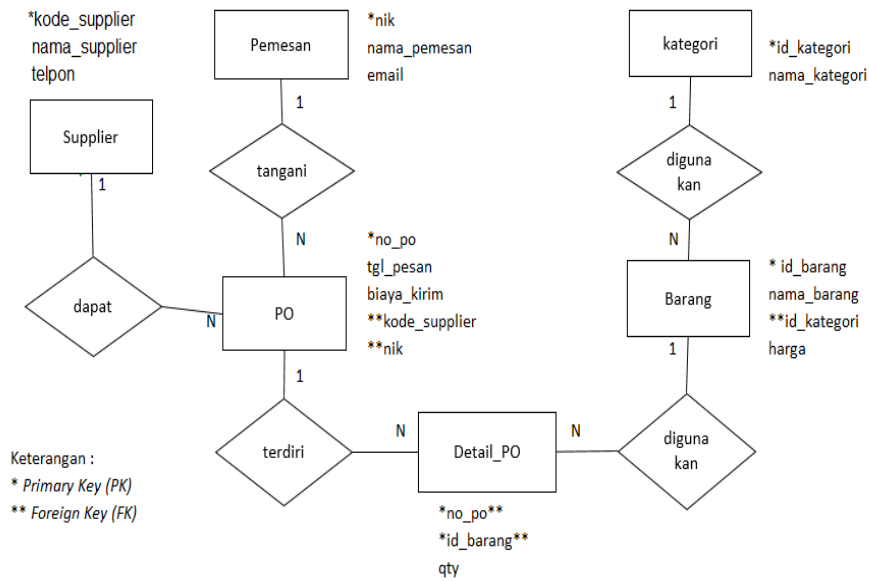


Sumber: Hasil Penelitian (2019)

Gambar 1. Kerangka Pemikiran

Penjelasan dari Gambar 1 adalah bahwa untuk melakukan implementasi *database ke SQL Server 2008*, tahapan awal adalah dengan mendapatkan sumber data ERD (*Entity Relationship Diagram*) dan struktur tabel-2 dari *database Purchase Order (PO)*. Selanjutnya dilakukan implementasi dengan menggunakan RDBMS *SQL Server 2008* dengan editor *query* dari pembuatan basis data, pembuatan tabel – tabel, insert data, pembuatan relasi antar tabel dan diagram dari *database Purchase Order (PO)* tersebut.

Sumber data diperoleh dari *Entity Relationship Diagram (ERD)* dari basis data *Purchase Order (PO)* yang terdiri dari 6 (enam) tabel yaitu Pemesan, Supplier, PO, Detail_PO, Barang dan Kategori [Setiyadi, 2018]



Sumber: Hasil Penelitian (2019)

Gambar 2. Entity Relationship Diagram (ERD) Purchase Order (PO)

Adapun integrasi data dari hasil ERD tersebut diatas adalah sebagai berikut:

Tabel 1. Integrasi data antar tabel pada basis data Purchase Order (PO)

Supplier			Pemesan		
kode_supplier	nama_supplier	telpon	nik	nama_pemesan	email
S120	PT. Global Jaya Solusi	021-8888777	K16001	Rini Hapsari	rini@lintangjaya.com
S210	PT. Revanda Jaya	021-6567787	K16007	Bagas Chandradewa	bagas@lintangjaya.com
PK (Primary Key)			PK (Primary Key)		

PO				
no_po	tgl_pesan	biaya_kirim	kode_supplier	nik
PO132	21/3/2018	150000	S120	K16001
PO173	23/3/2018	100000	S210	K16007
PK (Primary Key)			FK (Foreign Key)	

Detail_PO		
no_po	id_barang	qty
PO132	BR10211	10
PO132	BR23111	2
PO132	BR23112	1
PO173	BR10211	20
PO173	BR23118	10
PK (Primary Key) & FK (Foreign Key)		

Barang			
id_barang	nama_barang	id_kategori	harga
BR10211	Spidol	Kt001	5000
BR23111	Kulkas 2 Pintu	Kt010	4500000
BR23112	Meja Tamu	Kt011	7000000
BR23118	Meja Kantor	Kt011	200000
PK (Primary Key)		FK (Foreign Key)	

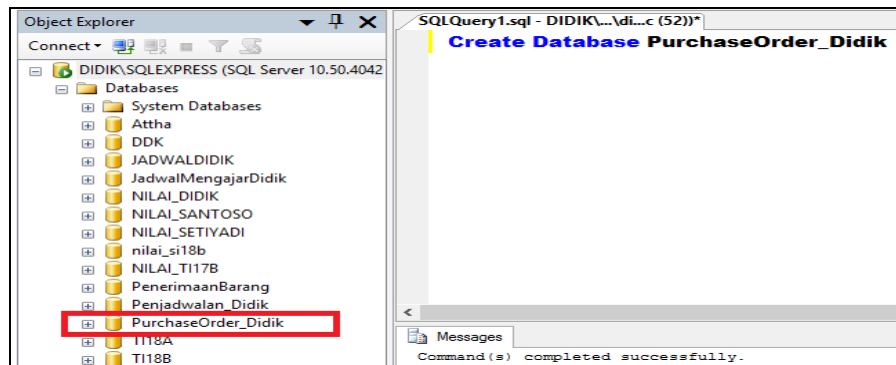
Kategori	
id_kategori	nama_kategori
Kt001	ATK
Kt010	Elektronik
Kt011	Inventaris Kantor
PK (Primary Key)	

Sumber: Hasil Penelitian (2019)

3. Hasil dan Pembahasan

3.1. Implementasi Basis Data

Berdasarkan hasil ERD dan integrasi data dari tabel – tabel dari basis data *Purchase Order* (PO), langkah selanjutnya adalah mengimplementasikan kedalam RDBMS *Sql Server 2008* yang dimulai dari pembuatan basis data. Berikut ini adalah sintaks query dalam pembuatan basis data, basis data yang akan dibuat adalah ‘PurchaseOrder_Didik’, seperti berikut ini.



Sumber: Hasil Penelitian (2019)

Gambar 3. Pembuatan (*Create*) basis data

3.2. Pembuatan Tabel.

Tabel – tabel yang akan dibuat meliputi tabel Supplier, Pemesan, PO, Detail_PO, Barang dan Kategori. Sebelum kita mengimplementasikan tabel – tabel tersebut dasarnya adalah dari ERD yang telah terbentuk yang diterjemahkan kedalam struktur tabel dari semua entitas/tabel tersebut. Untuk nama tabel, kolom dan tipe data harus menyesuaikan tipe data yang ada pada *SQL Server 2008*, tipe data dan lebar disesuaikan dengan data dari hasil 3 NF pada waktu kita melakukan normalisasi data.

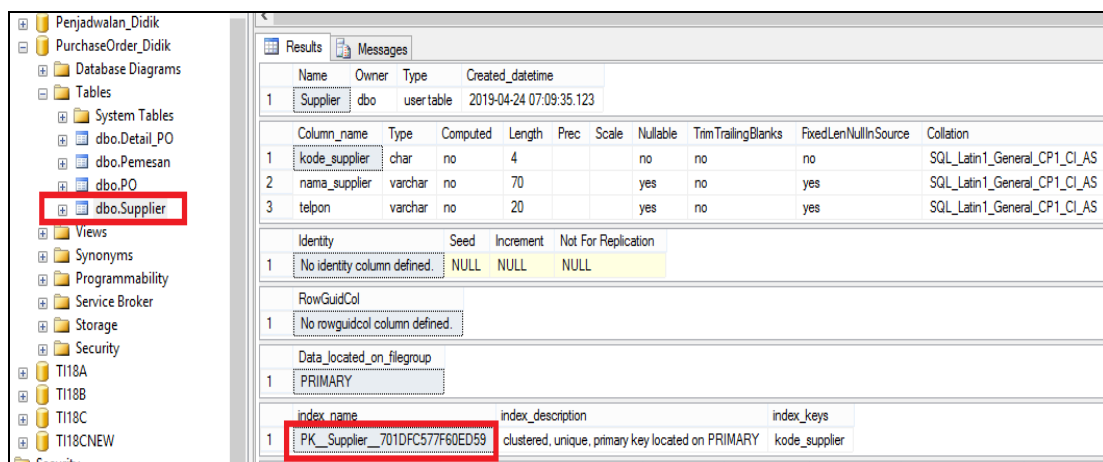
1. *Create table* Supplier

Berikut ini adalah sintaks *query* dari table Supplier, dimana pembentukan *Primary Key* (PK) tabel tersebut langsung dibuat pada saat membuat tabel.

Create Table Supplier

```
(
kode_supplier char(4) Primary Key,
nama_supplier varchar(70),
telpon varchar(20),
)
```

Index name : PK__Supplier__701DFC577F60ED59 merupakan *constraint Primary Key* (PK) dari tabel Supplier hasil *generate* dari *Sql Server 2008*. Untuk melihat struktur tabel Supplier dengan *query* ketik “*SP_Help Supplier*” pada editor, sehingga hasilnya seperti berikut ini.



Sumber: Hasil Penelitian (2019)

Gambar 4. Pembuatan *table* Supplier

2. Create table Pemesan

Berikut ini adalah sintaks *query* dari table Pemesan, dimana pembentukan *Primary Key* (PK) tabel tersebut dengan *alter table* pada saat membuat tabel.

Create Table Pemesan

```
(
  nik char(6) Not Null,
  nama_karyawan varchar(70),
  email varchar(40),
)
```

Jika kita membuat *table* Pemesan dengan sintaks dalam pembentukan PK dengan *alter table*, maka kolom yang menjadi PK pada *table* tersebut harus didefinisikan dengan *Not Null*, setelah ini baru dibuat PK nya dengan *sintaks* berikut ini.

Alter Table Pemesan

Add Constraint PKPemesan Primary Key(nik)

Sintaks PKPemesan merupakan *constraint Primary Key* (PK) pada *table* Pemesan tersebut. Untuk melihat struktur tabel Pemesan dengan *query* ketik “*SP_Help Pemesan*” pada editor, sehingga hasilnya seperti berikut ini.

Name	Owner	Type	Created_datetime
1 Pemesan	dbo	usertable	2019-04-24 07:31:05.027

Column_name	Type	Computed	Length	Prec	Scale	Nullable	TrimTrailingBlanks	FixedLenNullInSource	Collation
1 nik	char	no	6			no	no	no	SQL_Latin1_General_CP1_CI_AS
2 nama_karyawan	varchar	no	70			yes	no	yes	SQL_Latin1_General_CP1_CI_AS
3 email	varchar	no	40			yes	no	yes	SQL_Latin1_General_CP1_CI_AS

Identity	Seed	Increment	Not For Replication
1 No identity column defined.	NULL	NULL	NULL

RowGuidCol
1 No rowguidcol column defined.

Data_located_on_filegroup
1 PRIMARY

index name	index_description	index_keys
1 PKPemesan	clustered, unique, primary key located on PRIMARY	nik

Sumber: Hasil Penelitian (2019)

Gambar 5. Pembuatan *table* Pemesan

3. Create table PO

Berikut ini adalah sintaks *query* dari table PO, dimana pembentukan *Primary Key* (PK) tabel tersebut dengan *constraint* pada saat membuat tabel.

Create Table PO

```
(
  no_po char(6),
  tgl_pesan date,
  biaya_kirim decimal,
  kode_supplier char(4),
  nik char(6),
  Constraint PKPO Primary Key (no_po),
)
```

Sintaks PKPO merupakan *constraint Primary Key* (PK) pada *table* PO tersebut. Untuk melihat struktur tabel PO dengan *query* ketik “*SP_Help PO*” pada editor, sehingga hasilnya seperti berikut ini.

Name	Owner	Type	Created_datetime
1	PO	dbo	user table 2019-04-24 07:49:55.180

Column_name	Type	Computed	Length	Prec	Scale	Nullable	Trim TrailingBlanks	FixedLenNullInSource	Collation
1	no_po	char	no	6		no	no	no	SQL_Latin1_General_CP1_CI_AS
2	tgl_pesan	date	no	3	10	0	yes	(n/a)	NULL
3	biaya_kirim	decimal	no	9	18	0	yes	(n/a)	NULL
4	kode_supplier	char	no	4		yes	no	yes	SQL_Latin1_General_CP1_CI_AS
5	nik	char	no	6		yes	no	yes	SQL_Latin1_General_CP1_CI_AS

Identity	Seed	Increment	Not For Replication	
1	No identity column defined.	NULL	NULL	NULL

RowGuidCol	
1	No rowguidcol column defined.

Data_located_on_filegroup	
1	PRIMARY

index_name	index_description	index_keys
1	PKPO	clustered, unique, primary key located on PRIMARY no_po

Sumber: Hasil Penelitian (2019)

Gambar 6. Pembuatan *table* PO

4. Create *table* Detail_PO

Berikut ini adalah sintaks *query* dari *table* Detail_PO, dimana pembentukan *Primary Key* (PK) tabel tersebut dengan *alter table* pada saat membuat tabel.

Create *Table* Detail_PO

```
(
no_po char(6) Not Null,
id_barang char(8) Not Null,
Qty int,
)
```

Jika kita membuat *table* Detail_PO dengan sintaks dalam pembentukan PK dengan *alter table*, maka kolom yang menjadi PK pada *table* tersebut harus didefinisikan dengan *Not Null*, setelah ini baru dibuat PK nya dengan *sintaks* berikut ini:

Alter *Table* Detail_PO

Add *Constraint* PK_DPO *Primary Key* (no_po,id_barang)

Sintaks PK_DPO merupakan *constraint Primary Key* (PK) pada *table* Detail_PO tersebut. Untuk melihat struktur tabel Detail_PO dengan *query* ketik “SP_Help Detail_PO” pada editor, sehingga hasilnya seperti berikut ini.

Name	Owner	Type	Created_datetime
1	Detail_PO	dbo	usertable 2019-04-24 20:45:59.583

Column_name	Type	Computed	Length	Prec	Scale	Nullable	Trim TrailingBlanks	FixedLenNullInSource	Collation
1	no_po	char	no	6		no	no	no	SQL_Latin1_General_CP1_CI_AS
2	id_barang	char	no	8		no	no	no	SQL_Latin1_General_CP1_CI_AS
3	Qty	int	no	4	10	0	yes	(n/a)	NULL

Identity	Seed	Increment	Not For Replication	
1	No identity column defined.	NULL	NULL	NULL

RowGuidCol	
1	No rowguidcol column defined.

Data_located_on_filegroup	
1	PRIMARY

index_name	index_description	index_keys
1	PK_DPO	clustered, unique, primary key located on PRIMARY no_po, id_barang

Sumber: Hasil Penelitian (2019)

Gambar 7. Pembuatan *table* Detail_PO

5. Create *table* Barang

Berikut ini adalah sintaks *query* dari *table* Barang, dimana pembentukan *Primary Key* (PK) tabel tersebut langsung dibuat pada saat membuat tabel.

Create Table Barang

```
(
id_barang char(8) Primary Key,
nama_barang varchar(70),
id_kategori Char(6),
harga decimal,
)
```

Idex name : PK__Barang__13792DC608EA5793 merupakan *constraint Primary Key* (PK) dari tabel Barang hasil *generate* dari Sql Server 2008. Untuk melihat struktur tabel Barang dengan *query* ketik “SP_Help Barang” pada editor, sehingga hasilnya seperti berikut ini.

Name	Owner	Type	Created_datetime	
1	Barang	dbo	user table	2019-04-24 20:57:13.430

Column_name	Type	Computed	Length	Prec	Scale	Nullable	TrimTrailingBlanks	FixedLenNullInSource	Collation
1	id_barang	char	no	8		no	no	no	SQL_Latin1_General_CP1_CI_AS
2	nama_barang	varchar	no	70		yes	no	yes	SQL_Latin1_General_CP1_CI_AS
3	id_kategori	char	no	6		yes	no	yes	SQL_Latin1_General_CP1_CI_AS
4	harga	decimal	no	9	18	0	yes	(n/a)	NULL

Identity	Seed	Increment	Not For Replication	
1	No identity column defined.	NULL	NULL	NULL

RowGuidCol	
1	No rowguidcol column defined.

Data_located_on_filegroup	
1	PRIMARY

index_name	index_description	index_keys
1	PK__Barang__13792DC608EA5793	clustered, unique, primary key located on PRIMARY id_barang

Sumber: Hasil Penelitian (2019)

Gambar 8. Pembuatan *table* Barang

6. Create table Kategori

Berikut ini adalah sintaks *query* dari table Kategori, dimana pembentukan *Primary Key* (PK) tabel tersebut dengan *constraint* pada saat membuat tabel.

```
Create Table Kategori
(
id_kategori char(6),
Nama_kategori varchar(60),
Constraint PK_Kategori Primary Key(id_kategori),
)
```

Sintaks PK_Kategori merupakan *constraint Primary Key* (PK) pada *table* Kategori tersebut. Untuk melihat struktur tabel Kategori dengan *query* ketik “SP_Help Kategori” pada editor, sehingga hasilnya seperti berikut ini.

Name	Owner	Type	Created_datetime	
1	Kategori	dbo	user table	2019-04-24 21:13:25.747

Column_name	Type	Computed	Length	Prec	Scale	Nullable	TrimTrailingBlanks	FixedLenNullInSource	Collation
1	id_kategori	char	no	6		no	no	no	SQL_Latin1_General_CP1_CI_AS
2	Nama_kategori	varchar	no	60		yes	no	yes	SQL_Latin1_General_CP1_CI_AS

Identity	Seed	Increment	Not For Replication	
1	No identity column defined.	NULL	NULL	NULL

RowGuidCol	
1	No rowguidcol column defined.

Data_located_on_filegroup	
1	PRIMARY

index_name	index_description	index_keys
1	PK_Kategori	clustered, unique, primary key located on PRIMARY id_kategori

Sumber: Hasil Penelitian (2019)

Gambar 9. Pembuatan *table* Kategori

3.3. Pengisian data pada tabel (*insert data*).

Setelah mengimplementasikan pembuatan tabel Supplier, Pemesan, PO, Detail_PO, Barang dan Kategori. Langkah selanjutnya adalah melakukan pengisian data pada masing – masing tabel (*insert data*).

1. *Insert data* Supplier

Berikut ini adalah sintaks *query* untuk mengisi data tabel Supplier.

```
Insert Into Supplier Values ('S120','PT. Gloab Jaya Solusi','021-8888777')
```

```
Insert Into Supplier Values ('S210','PT. Revanda Jaya','021-6567787')
```

Untuk menampilkan data seluruh *record* pada tabel Supplier tersebut adalah dengan *sintaks sql* berikut ini.



The screenshot shows a SQL query editor with the following text:

```
Select *  
From Supplier  
--atau--  
Select kode_supplier,nama_supplier,telpon  
From Supplier
```

Below the query, the results are displayed in a table with two columns: 'Results' and 'Messages'. The 'Results' column shows two tables of data:

	kode_supplier	nama_supplier	telpon
1	S120	PT. Gloab Jaya Solusi	021-8888777
2	S210	PT. Revanda Jaya	021-6567787

	kode_supplier	nama_supplier	telpon
1	S120	PT. Gloab Jaya Solusi	021-8888777
2	S210	PT. Revanda Jaya	021-6567787

Sumber: Hasil Penelitian (2019)

Gambar 10. Menampilkan seluruh *record* data pada tabel Supplier

2. *Insert data* Pemesan

Berikut ini adalah sintaks *query* untuk mengisi data tabel Pemesan.

```
Insert Into Pemesan Values ('K16001','Rini Hapsari','rini@lintangjaya.com'),
```

```
('K16007','Bagas Chandradewa','bagas@lintangjaya.com')
```

Untuk menampilkan data seluruh *record* pada tabel Pemesan tersebut adalah dengan *sintaks sql* berikut ini (pada sintaks Select '**NIK**', huruf tidak sesuai pada struktur table yang seharusnya '**nik**' tidak menjadi masalah).



The screenshot shows a SQL query editor with the following text:

```
Select *  
From Pemesan  
--atau--  
Select NIK,nama_karyawan,email  
From Pemesan
```

Below the query, the results are displayed in a table with two columns: 'Results' and 'Messages'. The 'Results' column shows two tables of data:

	nik	nama_karyawan	email
1	K16001	Rini Hapsari	rini@lintangjaya.com
2	K16007	Bagas Chandradewa	bagas@lintangjaya.com

	NIK	nama_karyawan	email
1	K16001	Rini Hapsari	rini@lintangjaya.com
2	K16007	Bagas Chandradewa	bagas@lintangjaya.com

Sumber: Hasil Penelitian (2019)

Gambar 11. Menampilkan seluruh *record* data pada tabel Pemesan

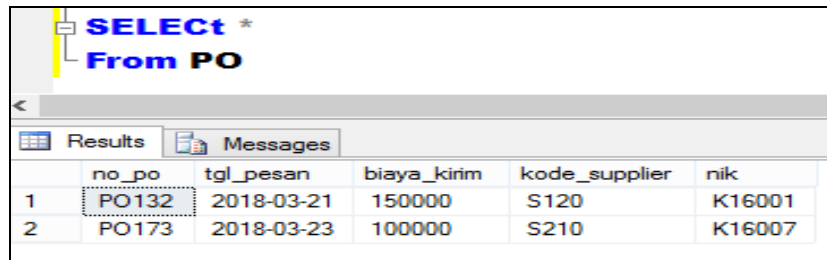
3. *Insert data* PO

Berikut ini adalah sintaks *query* untuk mengisi data tabel PO.

```
Insert Into PO Values ('PO132','03/21/2018',150000,'S120','K16001')
```

```
Insert Into PO Values ('PO173','03/23/2018',100000,'S210','K16007')
```

Untuk menampilkan data seluruh *record* pada tabel PO tersebut adalah dengan *sintaks sql* berikut ini.



	no_po	tgl_pesan	biaya_kirim	kode_supplier	nik
1	PO132	2018-03-21	150000	S120	K16001
2	PO173	2018-03-23	100000	S210	K16007

Sumber: Hasil Penelitian (2019)

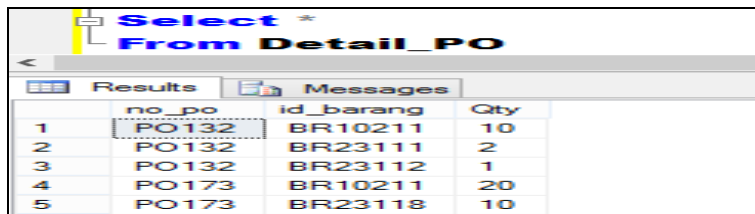
Gambar 12. Menampilkan seluruh *record* data pada tabel PO

4. *Insert data* Detail_PO

Berikut ini adalah *sintaks query* untuk mengisi data tabel Detail_PO.

```
Insert Into Detail_PO Values ('PO132','BR10211',10),
('PO132','BR23111',2),
('PO132','BR23112',1),
('PO173','BR10211',20),
('PO173','BR23118',10)
```

Untuk menampilkan data seluruh *record* pada tabel Detail_PO tersebut adalah dengan *sintaks sql* berikut ini.



	no_po	id_barang	Qty
1	PO132	BR10211	10
2	PO132	BR23111	2
3	PO132	BR23112	1
4	PO173	BR10211	20
5	PO173	BR23118	10

Sumber: Hasil Penelitian (2019)


Gambar 13. Menampilkan seluruh *record* data pada tabel PO

5. *Insert data* Barang

Berikut ini adalah *sintaks query* untuk mengisi data tabel Barang.

```
Insert Into Barang Values ('BR10211','Spidol','Kt001',5000),
('BR23111','Kulkas 2 Pintu','Kt010',4500000),
('BR23112','Meja Tamu','Kt011',7000000),
('BR23118','Meja Kantor','Kt011',2000000)
```

Untuk menampilkan data seluruh *record* pada tabel Barang tersebut adalah dengan *sintaks sql* berikut ini:



	id_barang	nama_barang	id_kategori	harga
1	BR10211	Spidol	Kt001	5000
2	BR23111	Kulkas 2 Pintu	Kt010	4500000
3	BR23112	Meja Tamu	Kt011	7000000
4	BR23118	Meja Kantor	Kt011	2000000

Sumber: Hasil Penelitian (2019)

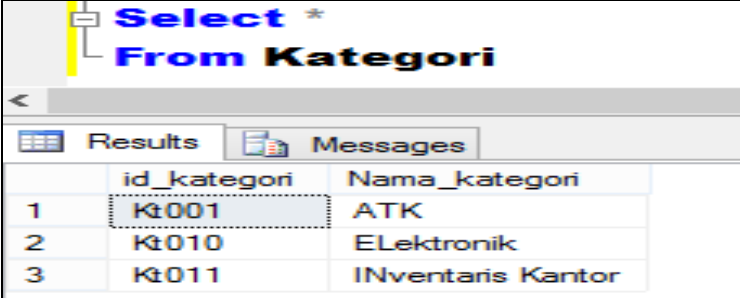
Gambar 14. Menampilkan seluruh *record* data pada tabel Barang

6. *Insert data* Kategori

Berikut ini adalah *sintaks query* untuk mengisi data tabel Kategori.

Insert Kategori Values ('Kt001','ATK'),
('Kt010','ELEktronik'),
('Kt011','INventaris Kantor')

Untuk menampilkan data seluruh *record* pada tabel Kategori tersebut adalah dengan *sintaks sql* berikut ini.



	id_kategori	Nama_kategori
1	Kt001	ATK
2	Kt010	ELEktronik
3	Kt011	INventaris Kantor

Sumber: Hasil Penelitian (2019)

Gambar 15. Menampilkan seluruh *record* data pada tabel Kategori

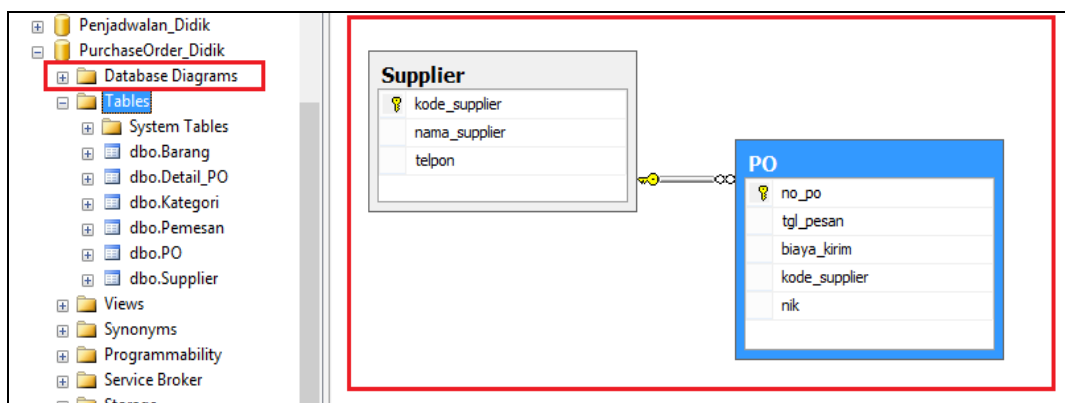
3.4. Membuat relasi antar *table*

Setelah mengimplementasikan pembuatan tabel dan pengisian data untuk Supplier, Pemesan, PO, Detail_PO, Barang dan Kategori. Langkah selanjutnya adalah melakukan pembuatan relasi antar tabel.

1. Relasi tabel Supplier ke PO
Berikut ini adalah *sintaks query* untuk merelasikan tabel Supplier ke PO.

Alter Table PO
Add Constraint FKSupplier Foreign Key(kode_supplier)
References Supplier(kode_supplier)

Untuk menampilkan relasi antara tabel Supplier ke PO, dapat dilihat pada *databases diagrams* dengan hasil sebagai berikut.



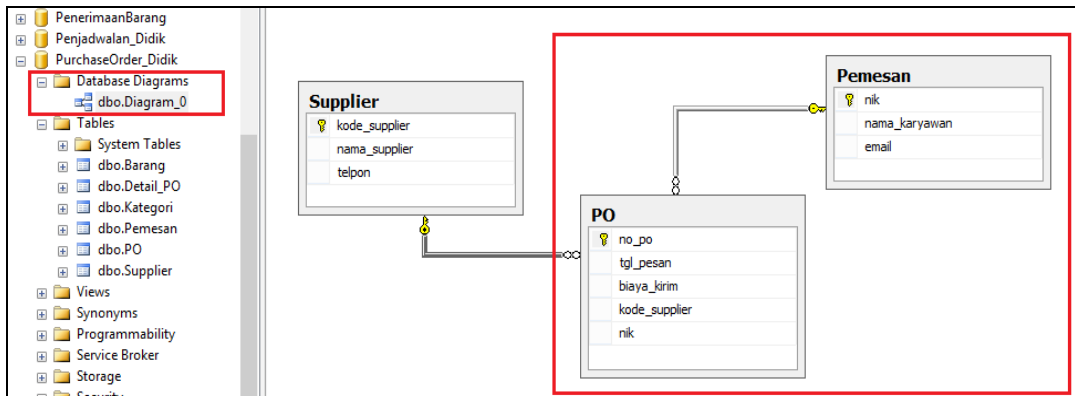
Sumber: Hasil Penelitian (2019)

Gambar 16. Relasi tabel Supplier ke PO

2. Relasi tabel Pemesan ke PO
Berikut ini adalah *sintaks query* untuk merelasikan tabel Pemesan ke PO.

Alter Table PO
Add Constraint FKPemesan Foreign Key(nik)
References Pemesan(nik)

Untuk menampilkan relasi antara tabel Pemesan ke PO, dapat dilihat pada *databases diagrams* dengan hasil sebagai berikut.



Sumber: Hasil Penelitian (2019)

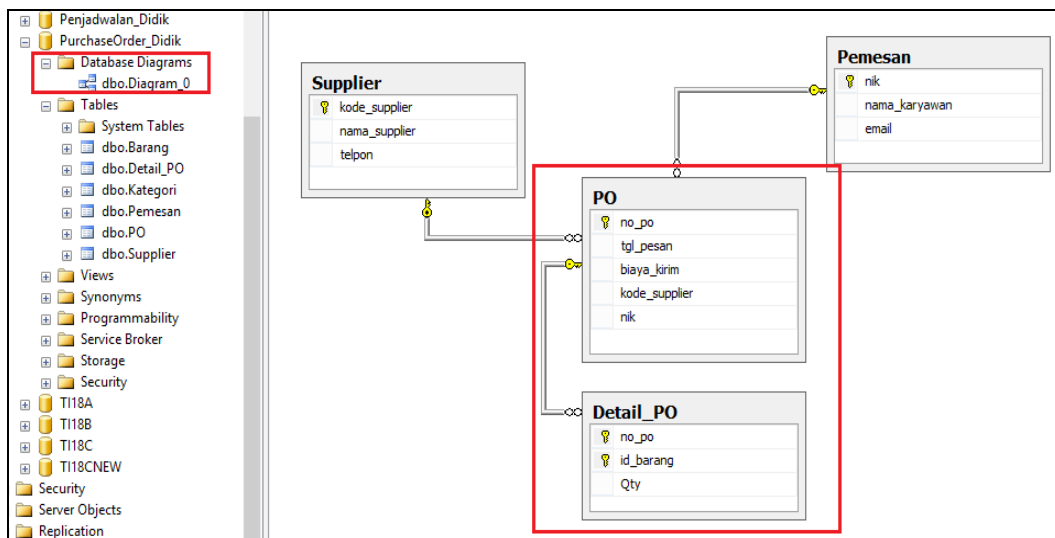
Gambar 17. Relasi tabel Pemesan ke PO

3. Relasi tabel PO Ke Detail_PO

Berikut ini adalah *sintaks query* untuk merelasikan tabel PO ke Detail_PO.

```
Alter Table Detail_PO
Add Constraint FKPO Foreign Key(no_po)
References PO(no_po)
```

Untuk menampilkan relasi antara tabel PO ke Detail_PO, dapat dilihat pada *databases diagrams* dengan hasil sebagai berikut.



Sumber: Hasil Penelitian (2019)

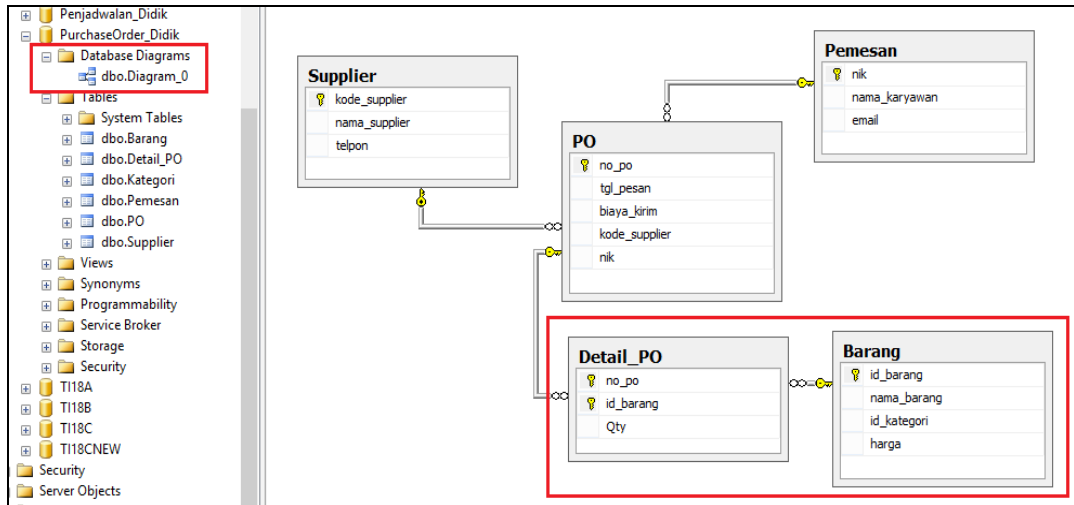
Gambar 18. Relasi tabel PO ke Detail_PO

4. Relasi tabel Barang Ke Detail_PO

Berikut ini adalah *sintaks query* untuk merelasikan tabel Barang ke Detail_PO.

```
Alter Table Detail_PO
Add Constraint FKBarang Foreign Key(id_barang)
References Barang(id_barang)
```

Untuk menampilkan relasi antara tabel Barang ke Detail_PO, dapat dilihat pada *databases diagrams* dengan hasil sebagai berikut.



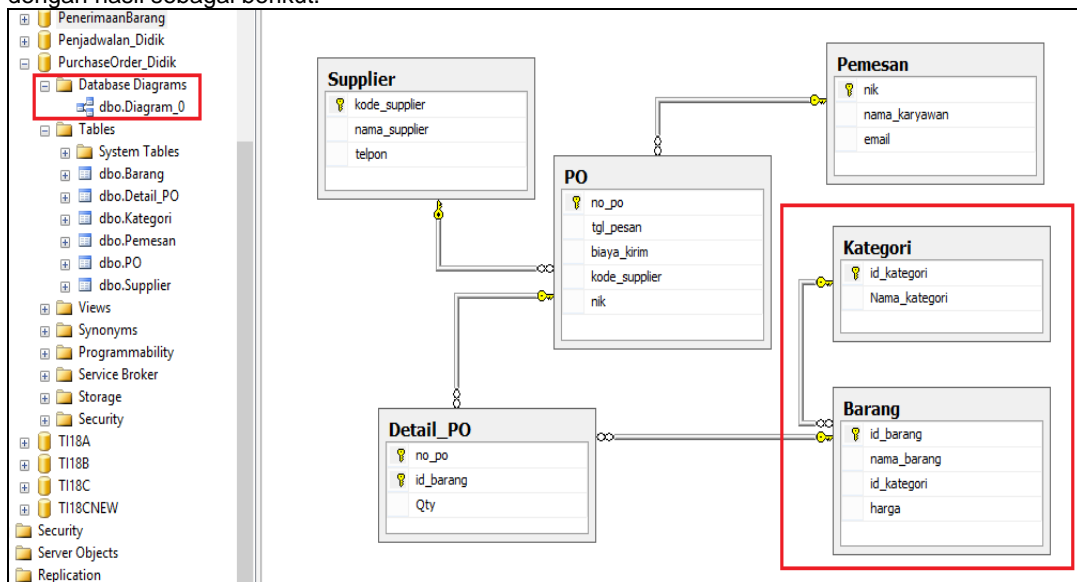
Sumber: Hasil Penelitian (2019)

Gambar 19. Relasi tabel Barang ke Detail_PO

5. Relasi tabel Kategori Ke Barang
Berikut ini adalah *sintaks query* untuk merelasikan tabel Kategori ke Barang.

Alter Table Barang
Add Constraint FKKategori Foreign Key(id_kategori)
References Kategori(id_kategori)

Untuk menampilkan relasi antara tabel Kategori ke Barang, dapat dilihat pada *databases diagrams* dengan hasil sebagai berikut.

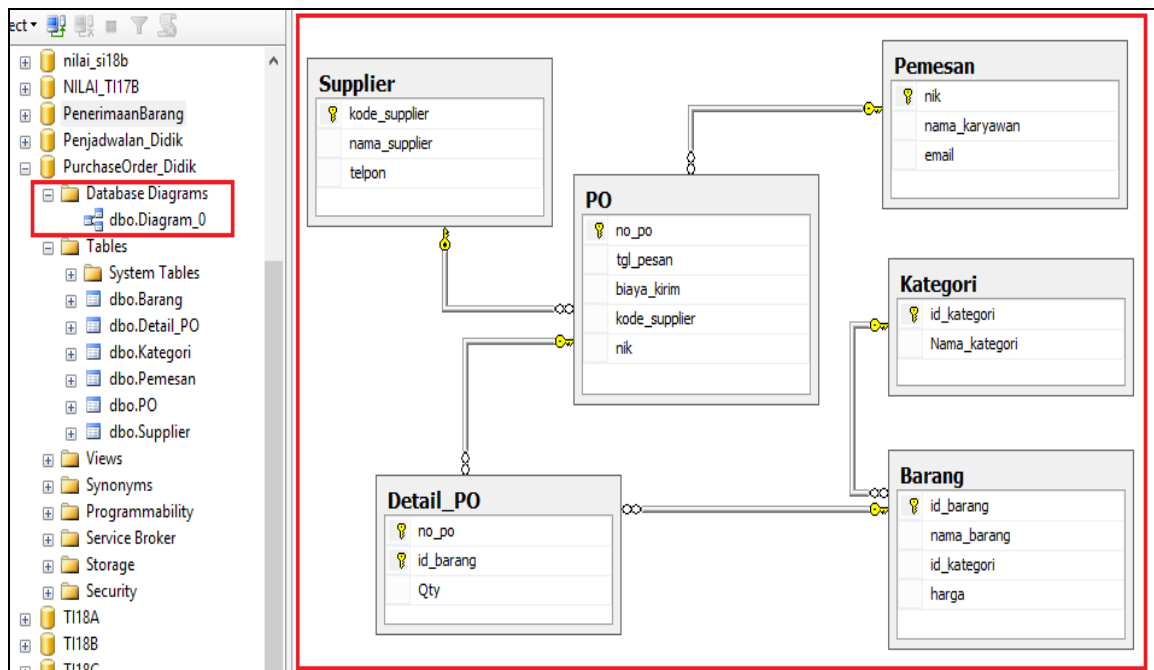


Sumber: Hasil Penelitian (2019)

Gambar 20. Relasi tabel Kategori ke Barang

3.5. Membuat *diagram database*

Setelah mengimplementasikan pembuatan tabel dan engisian data untuk Supplier, Pemesan, PO, Detail_PO, Barang dan Kategori serta relasi antar tabel – tabel. Langkah selanjutnya adalah membuat diagram dari basis data *Purchase Order* (PO) seperti gambar berikut ini.



Sumber: Hasil Penelitian (2019)

Gambar 21. Diagram basis data *Purchases Order* (PO)

4. Kesimpulan

Berdasarkan hasil implementasi dari basis data *Purchase Order* (PO) yang telah dilakukan, kesimpulan dari penelitian ini adalah: 1). Implementasi dari basis data *Purchase Order* (PO) menggunakan RDBMS *SQL Server 2008* yang didasarkan dari *Entity Relationship Diagram* (ERD) beserta struktur dari tabel-tabel dengan menggunakan bahasa query. 2). Langkah berikutnya adalah dengan membuat basis data, kemudian pembentukan tabel-tabel, pengisian data dari tabel tersebut dengan menggunakan query. 3). Langkah akhir dengan melakukan pembuatan relasi antar tabel yaitu relasi tabel *Supplier* ke *PO*, *Pemesan* ke *PO*, *PO* ke *Detail_PO*, *Barang* ke *Detail_PO* dan *Kategori* ke *Barang* yang selanjutnya akan terbentuk diagram basis data *Purchase Order* (PO) 4). Dari hasil diagram tersebut, dapat dijadikan dasar dalam implementasi pembuatan program dengan menggunakan bahasa pemrograman seperti *PHP*, *VB.Net* ataupun *Java*.

Referensi

- Indrajani. 2018. *Database Design All in One (Theory, Practice and Case Study)*. Jakarta: PT Elex Media Komputindo.
- Nuryana M, Sulistiyono. 2014. *Analisa dan Perancangan Sistem Front Office Hotel Bidakara Serang*. *Protekinfo* 1: 1–5.
- Sari A, Milwadhari S. 2016. *Basis Data Oracle Fundamental*. Yogyakarta: Deepublish.
- Setiyadi D. 2018. *Normalisasi Dalam Perancangan Basis Data Relasional Purchase Order (PO)*. 3: 67–78.
- Sutanta E. 2011. *Basis Data dalam Tinjauan Konseptual*. Yogyakarta: Andi.
- Wahana K. 2013. *SQL Server 2012*. Yogyakarta: Andi.