LAPORAN TUGAS BESAR BASIS DATA

Aplikasi PickMe Store

Laporan ini disusun untuk memenuhi Tugas Mata Kuliah Basis Data Praktikum.

Disusun oleh:

Muhammad Daffa Raihandika	211524050
Mochamad Ferdy Fauzan	211524049
Raka Mahardika Maulana	211524056

PROGRAM STUDI D4 TEKNIK INFORMATIKA

JURUSAN TEKNIK KOMPUTER DAN INFORMATIKA

POLITEKNIK NEGERI BANDUNG

2022

KATA PENGANTAR

Segala puji bagi Allah SWT atas nikmat yang sudah diberikan, penulis sudah dapat membuat serta menyelesaikan Laporan dengan judul "Aplikasi PickMeStore". Adapun pembuatan laporan ini ditujukan untuk syarat memenuhi tugas besar Sistem Basis Data Praktikum. Tim penulis dapat menyelesaikan laporan dengan baik tidak lepas berkat bantuan dari teman-teman dan kampus tim menimba ilmu.

Tidak lupa, kami mengucapkan terima kasih kepada Pak Urip Teguh dan Pak Muhammad Riza Alifi yang telah membimbing dan membantu kami dalam proses penyusunan laporan ini.Ucapan terimakasih juga kami sampaikan kepada teman - teman yang telah membantu sehingga Laporan dengan judul “Aplikasi PickMeStore" dapat dibuat.

Laporan ini menjelaskan sistem database "Aplikasi PickMeStore" dan Business proccess Dalam penggunaan Sistem Database online shop ini.

Tim penulis menyadari bahwa masih ada kekurangan dan kesalahan dalam laporan yang disusun. Oleh karena itu penulis mohon maaf atas kesalahan tersebut. Kritik dan saran dari pembaca senantiasa ditunggu oleh penulis guna meningkatkan kualitas tulisan kedepan.

Bandung, 06 Desember 2022

Penulis

DAFTAR ISI

KATA PENGANTAR ..... **i**
DAFTAR ISI ..... **i**
DAFTAR GAMBAR ..... **iii**
DAFTAR TABEL ..... **iv**
BAB I PENDAHULUAN **1**
1.1 Latar Belakang 1
1.2 Definisi Masalah 1
1.3 Batasan Masalah 2
1.4 Overview Sistem 2
BAB II ANALISIS **4**
2.1 Deskripsi Sistem 4
1.1 Business Process 5
1.1.1 Event Description 6
2.1.1 Context Diagram, DFD 32
2.1.1.1 Context Diagram (DFD Level 0) 32
2.1.1.2 DFD Level 1 33
2.1.1.3 DFD Level 2 34
BAB III PERANCANGAN **37**
3.1 Model Skema Logical 37
3.1.1 Notasi Chen 37
3.1.2 Oracle Data Modeler 38
3.2 Model Skema Relational 39
3.2.1 Relational (Physical) Data Model 39
3.2.2 DDL 40
3.2.3 DML 45
3.3 [Normalisasi] 50
3.3.1 UNF 51
3.3.2 1NF 51
3.3.3 2NF 53
3.3.4 3NF 55
3.4 Mapping Business Process ke PLSQL 57

3.4.1 Procedure 57
3.4.2 Function 68
3.4.3 Trigger 70
3.5 Arsitektur Aplikasi 73
3.5.1 Database 73
3.5.2 Front-End 73
3.5.3 Back-End 74
BAB IV IMPLEMENTASI 75
4.1 Teknologi Pengembangan Aplikasi 75
4.1.1 Data Layer 75
4.1.2 Service layer 75
4.1.3 Application layer 75
4.2 Testing 76
4.2.1 Testing Admin 76
4.2.2 Testing Customer 77
BAB V KESIMPULAN 81
BAB VI LESSON LEARNED 82
6.1 Muhammad Daffa Raihandika 82
6.2 Mochamad Ferdy Fauzan 82
6.3 Raka Mahardika Maulana 82
DAFTAR PUSTAKA 83

DAFTAR GAMBAR

Gambar 1.1 Business Process Model and Notation (BPMN) Sistem Aplikasi PickMeStore 5
Gambar 1.2 Context Diagram dari Sistem Aplikasi PickMeStore..... 32
Gambar 1.3 DFD Level 0 dari Sistem Aplikasi PickMeStore 33
Gambar 1.4 DFD Level 2 dari Process 1.0 34
Gambar 1.5 DFD Level 2 dari Process 2.0 35
Gambar 1.6 DFD Level 2 dari Process 3.0 36
Gambar 3.1.1 Skema Logical Sistem Aplikasi PickMeStore dengan Notasi Chen 37
Gambar 3.1.2 Skema Logical Sistem Aplikasi PickMeStore dengan Oracle Data Modeler 38
Gambar 3.2.1 Skema Relational Sistem Aplikasi PickMeStore dengan Oracle Data Modeler..... 39
Gambar 3.3.1 Contoh Payment Invoice dari Sistem Aplikasi PickMeStore 50
Gambar 3.3.2 Normalisasi dalam Bentuk 3NF dari Payment Invoice yang menghasilkan beberapa tabel baru masing – masing memiliki id sebagai primary key 56
Gambar 3.5.1 Arsitektur Aplikasi mulai dari Data layer, Service layer hingga Application layer..... 73

DAFTAR TABEL

Tabel 1.1.1 Tabel Event Register 6
Tabel 1.1.2 Tabel Event Login 8
Tabel 1.1.3 Tabel Event Search Product by Cateogory 10
Tabel 1.1.4 Tabel Event Search Product 11
Tabel 1.1.5 Tabel Event Checkout 13
Tabel 1.1.6 Tabel Event Penanganan Stok 18
Tabel 1.1.7 Tabel Event Top Up Saldo 22
Tabel 1.1.8 Tabel Add Stock Product 24
Tabel 1.1.9 Tabel Event Cancel Order 25
Tabel 1.1.10 Tabel Event Change Password 28
Tabel 1.1.11 Tabel Event Delete Product 30
Tabel 3.2.1 DDL Tabel User 40
Tabel 3.2.2 DDL Tabel Address 41
Tabel 3.2.3 DDL Tabel Product 41
Tabel 3.2.4 DDL Tabel Category 42
Tabel 3.2.5 DDL Tabel Order_Items 43
Tabel 3.2.6 DDL Tabel Order_Details 43
Tabel 3.2.7 DDL Tabel Payment_Details 45
Tabel 3.2.8 DML Fitur Registrasi Akun 45
Tabel 3.2.9 DML Fitur Menambah Alamat 46
Tabel 3.2.10 DML Fitur Menambah Kategori 46
Tabel 3.2.11 DML Fitur Menambah Produk 47
Tabel 3.2.12 DML Fitur Update Stok Produk 47
Tabel 3.2.13 DML Update Password, Email, dan No. Hp User 47
Tabel 3.2.14 DML Fitur Update Alamat 48
Tabel 3.2.15 DML Fitur Menghapus Akun 48
Tabel 3.2.16 DML Fitur Menghapus Alamat 49
Tabel 3.3.1 Normalisasi dalam Bentuk UNF dari Payment Invoice 51
Tabel 3.3.2 Normalisasi dalam Bentuk 1NF dari Payment Invoice 51
Tabel 3.3.3 Tabel Customer, Normaliasi dalam Bentuk 2NF dari Payment Invoice 53
Tabel 3.3.4 Tabel Address, Normaliasi dalam Bentuk 2NF dari Payment Invoice 53
Tabel 3.3.5 Tabel Payment, Normaliasi dalam Bentuk 2NF dari Payment Invoice 54
Tabel 3.3.6 Tabel Product, Normaliasi dalam Bentuk 2NF dari Payment Invoice54
Tabel 3.4.1 Procedure Register 57
Tabel 3.4.2 Procedure AddProduct 57
Tabel 3.4.3 Procedure ShowProductDetails 58
Tabel 3.4.4 Procedure ChangePassword 58

Tabel 3.4.5 Procedure AddAddress 59
Tabel 3.4.6 Procedure CreateOrderDetails 60
Tabel 3.4.7 Procedure Order_Per_Produk 60
Tabel 3.4.8 Procedure SumOrderProduct 61
Tabel 3.4.9 Procedure TopUpSaldo 61
Tabel 3.4.10 Procedure AddStokProduct 62
Tabel 3.4.11 Procedure PaymentStatus 62
Tabel 3.4.12 Procedure PayOrder 63
Tabel 3.4.13 Procedure CancelOrder 64
Tabel 3.4.14 Procedure CreateDetailPesanan 65
Tabel 3.4.15 Procedure CreateInvoice 66
Tabel 3.4.16 Procedure DeleteProduct 67
Tabel 3.4.17 Procedure SearchProductFromCategory 67
Tabel 3.4.18 Function Login 68
Tabel 3.4.19 Function GetUserId 69
Tabel 3.4.20 Function GetProductId 69
Tabel 3.4.21 Function GetOrderDetailsId 69
Tabel 3.4.22 Trigger trg_update_stok 70
Tabel 3.4.23 Trigger cancelled_order 71
Tabel 3.4.24 Trigger trg_transaksi_lunas 72
Tabel 3.4.25 Trigger trg_expired_payment 72

BAB I
PENDAHULUAN

1.1 Latar Belakang

Seiring perkembangan waktu dan teknologi, manusia memilih jalan hidup yang lebih praktis dan serba mudah, salah satu contoh dari perkembangan teknologi yang dapat merealisasikan hal tersebut adalah dengan munculnya online shop.

Online shop merupakan toko yang semua proses transaksinya dilakukan secara digital melalui perantara device yang digunakan oleh pembeli dan penjual. dengan ini konsumen tidak perlu pergi keluar rumah untuk membeli barang yang diinginkan.

Dengan online shop konsumen dapat membeli barang kapanpun dan dimanapun, begitu juga dengan penjual yang dapat menjual produknya dengan kondisi serupa. Di sini konsumen ditawarkan barang dalam bentuk display yang disediakan penjual sebagai gambaran dari produk yang akan dibeli oleh konsumen.

Sebagai salah satu alternatif bisnis digital di zaman modern tentunya online shop membawa beberapa manfaat bagi penjual dan pembeli, beberapa di antaranya membuat pengalaman belanja menjadi mudah dan praktis, bisa membeli produk maupun barang dari tempat yang jauh , dan dapat dengan mudah membandingkan harga barang. Untuk merealisasikan hal ini maka kami membuat PickMe Store untuk menjawab keresahan - keresahan konsumen selama ini.

1.2 Definisi Masalah

Aplikasi PickMe Store merupakan aplikasi yang akan dikembangkan oleh kelompok kami sebagai aplikasi online shop. Pada aplikasi ini terdapat 2 role, yaitu admin dan customer.

- **Admin**
  - Dapat mengelola database secara keseluruhan
  - Dapat memasukkan barang ke dalam database
  - Dapat melihat barang yang terdapat dalam database
  - Dapat mengganti barang yang terdapat dalam database
  - Dapat menghapus barang yang terdapat dalam database

- **Customer**
  - Dapat melihat barang yang tersedia
  - Top up saldo
  - Memesan barang
  - Checkout pesanan

- Cancel order

Dalam studi kasus ini, customer tidak dapat memasukkan barang ke dalam database, mengganti, dan menghapus barang yang terdapat dalam database. Hal ini dikarenakan admin yang hanya mampu mengelola database secara keseluruhan, sehingga tingkatan admin berada di atas customer

1.3 Batasan Masalah

Dari definisi masalah pada studi kasus yang kami buat, terdapat beberapa batasan dari solusi yang akan diberikan. Batasan-batasan tersebut diantaranya adalah :

- Database akan dikembangkan menjadi aplikasi berbasis web, user dapat mengakses web yang kami buat melalui web browser pada mobile maupun desktop.
- User yang ingin membeli barang pada aplikasi yang kami buat (Customer) harus memiliki akun. Jika user belum memiliki akun maka harus membuat akun terlebih dahulu.
- Customer dapat membeli barang apabila stok barang tersebut masih tersedia, barang tersebut dapat dipesan lebih dari satu buah dan lebih dari satu jenis barang.
- Pesanan yang sudah dicheckout akan kedaluarsa maksimal satu hari jika customer belum melakukan pembayaran. Pembayaran hanya dapat dilakukan virtual payment.
- Apabila customer melakukan cancel order, maka stok dari produk yg dipesan oleh customer tersebut akan ditambah sesuai dengan quantity yg dipesan oleh customer tersebut.
- Satu order dilakukan satu pembayaran.
- Customer masing-masing memiliki saldo.
- Customer dapat membeli barang apabila saat melakukan pembayaran customer memiliki saldo yg cukup, apabila customer tidak memiliki saldo yg cukup maka customer tidak dapat membeli barang tersebut.
- Customer dapat melakukan top up saldo.
- Customer tidak dapat mengubah jumlah pesanan yg sudah dipesan, apabila customer ingin mengubah jumlah pesanan yg dipesan, maka customer harus melakukan cancel order terlebih dahulu, lalu customer dapat melakukan order kembali.

1.4 Overview Sistem

PickMe Store adalah sebuah e- commerce content management system (CMS) yang menghubungkan penjual dan pembeli dengan sistem

toko. Toko dari PickMe Store sendiri merupakan toko yang bergerak di bidang usaha clothing atau fashion.

Pihak penjual atau admin dapat menambah, memperbarui, menghapus, mengganti serta mengatur barang yang dijual pada laman aplikasi. Pihak pembeli atau customer dapat membeli barang yang terdapat dalam katalog sesuai dengan ketersediaan stok barang di database PickMe Store.

BAB II
ANALISIS

2.1 Deskripsi Sistem

PickMe Store adalah sebuah e-commerce content management system (CMS) yang menghubungkan penjual dan pembeli dengan sistem toko. Toko dari PickMe Store sendiri merupakan toko yang bergerak di bidang usaha clothing atau fashion. Pihak penjual atau admin dapat menambah, memperbarui, menghapus, mengganti serta mengatur barang yang dijual pada laman aplikasi. Pihak pembeli atau customer dapat membeli barang yang terdapat dalam katalog sesuai dengan ketersediaan stok barang di database PickMe Store.

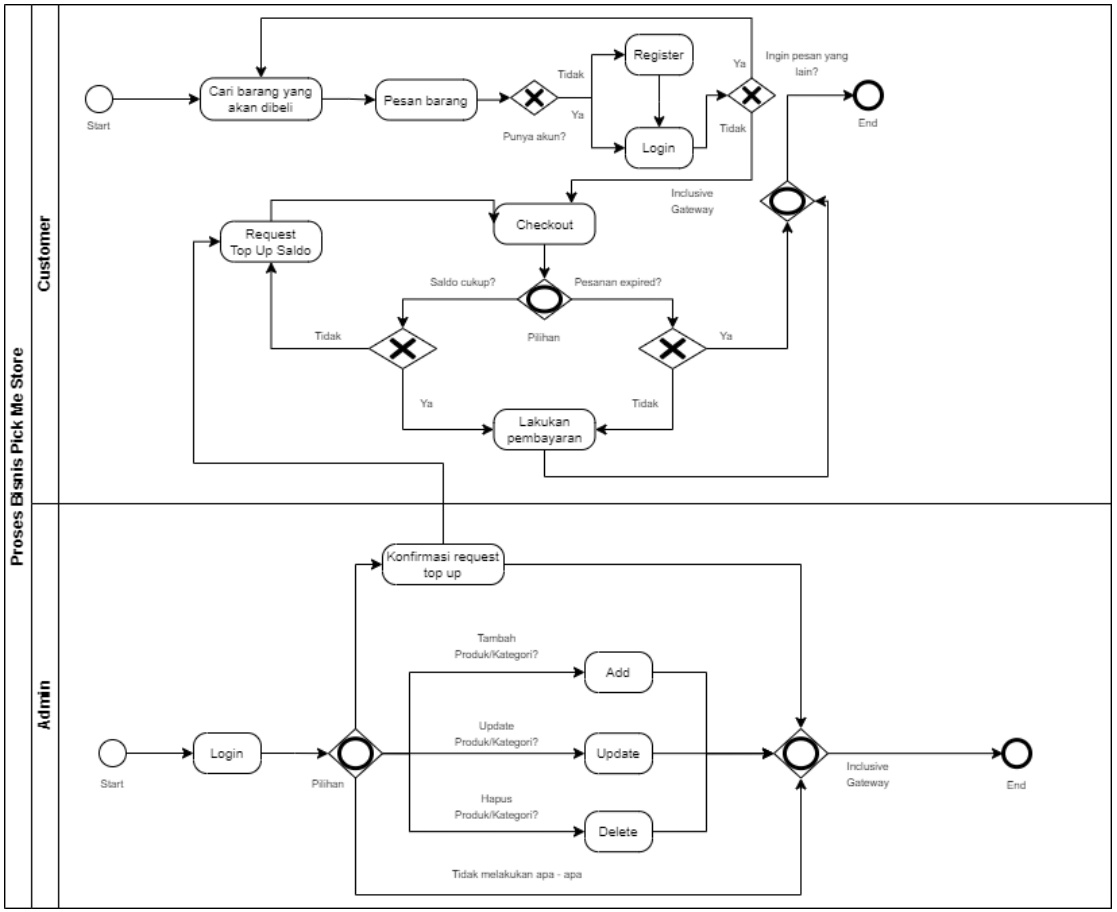
Calon pembeli dapat melihat barang apa saja yang tersedia pada laman aplikasi. Namun jika ingin membelinya, pembeli diharuskan mempunyai akun yang terdaftar di database PickMe Store.

Jika belum memiliki akun, maka calon pembeli harus registrasi akun terlebih dahulu. Apabila calon pembeli telah login menggunakan akun yang sudah terdaftar, ia dapat memesan barang yang hendak dibeli . Ketika memasuki proses checkout, maka pembeli diminta untuk memasukkan alamat tujuan pengiriman barang dan melakukan pembayaran.

Tersedia laman transaksi yang memuat riwayat transakasi yang dilakukan oleh pembeli menggunakan akunnya. Selain itu, pembeli dapat melihat tanggal transaksi, transaksi yang belum dan sudah dibayar, serta total pembayaran dari transaksi. Pembeli melakukan pembatalan pesanan.

Penjual juga dapat membatalkan pesanan pembeli apabila pesanan yang dilakukan tidak memenuhi ketentuan pembelian, misalnya stok barang tiba-tiba habis sebelum stok barang diperbarui. Pesanan yang sudah dicheckout memiliki waktu 1 hari untuk melakukan proses pembayaran. Jika dalam kurun waktu tersebut pesanan belum dibayar, maka pesanan otomatis akan kedaluarsa.

1.1 Business Process



Gambar 1.1.1 Business Process Model and Notation (BPMN) Sistem Aplikasi PickMeStore

1.1.1 Event Description

Tabel 1.1.1 Tabel Event Register

Deskripsi				
Customer ingin membuat akun baru				
Respon Sistem				
No	Deskripsi Respon	Input	Output	Aturan
1	System akan mengarahkan customer untuk membuat username dan password untuk login ke akun , setelah itu user akan diminta untuk mengisi biodata diri.	Username Password Retype password Fullname Nomor telepon email	Biodata customer telah tersimpan.	Username available atau harus berbeda dengan username user lain. Username maksimal 15 karakter Password minimal 8 karakter user akan diminta untuk memasukan password 2 kali , sebagai verifikasi dan kedua password harus sama. Email harus sesuai dengan format email Nomor telepon harus diawali dengan 0,62 atau +62
Kamus Data: user: {Id_User, Username, Password, full_name, Email, No_telp, saldo, Role}				
Nama Tabel	User			

No.	Nama Kolom	Tipe Data	Constraint	Deskripsi	Contoh Data
1.	Id_User	Integer	Primary key	Id_Customer merupakan primary key tabel Akun	1
2.	Full_name	Varchar	Maksimal 50 karakter	Merupakan field untuk nama lengkap customer	Daffa Raihandika
3.	Username	Varchar	Username harus available (belum ada yang menggunakan username tersebut) Username maksimal 15 karakter	Username akan digunakan untuk login. Username harus unik (tidak boleh ada username yang sama)	daffaraihandika
4.	Password	Varchar	Minimal 8 karakter	Password yang diinputkan oleh customer akan dienkripsi terlebih dahulu dengan cara hashing password untuk keamanan, sehingga password yang terdapat di database tidak dapat dibaca oleh pengelola database	sebelum dienkripsi : rakaganteng12* setelah dienkripsi : b10a8db164e0754105b7
5.	Email	Varchar	Harus terdapat domain	Email yang diinputkan customer ke dalam database harus valid	ferdyfauzan@gmail.com
6.	No_telepon	Varchar	Maksimal 15 karakter Karakter harus berupa angka tidak boleh ada	Merupakan field untuk nomor telepon customer	0878653201287

			karakter lain selain angka Harus diawali 0, 62 atau +62		
7	Role	Varchar	Field ini hanya dapat diisi dengan value ‘Admin’ atau ‘Customer’	Field untuk menandakan pemilik akun apakah admin atau customer	Customer
8	saldo	Integer	Saldo harus lebih besar sama dengan nol	Field untuk jumlah saldo yang dimiliki oleh user.	500000

Tabel 1.1.2 Tabel Event Login

Deskripsi				
user ingin login menggunakan akun yang telah dibuat				
Respon Sistem				
No	Deskripsi Respon	Input	Output	Aturan
1	System akan mengarahkan user untuk memasukan username dan password untuk login ke akun .	Username Password	user berhasil login	Username sesuai dengan yang dibuat password valid
Kamus Data: user: {Id_User, Username, Password, full_name, Email, No_telp, saldo, Role}				

Nama Tabel	User				
No.	**Nama Kolom**	**Tipe Data**	**Constraint**	**Deskripsi**	**Contoh Data**
1.	Id_User	Integer	Primary key	Id_Customer merupakan primary key tabel Akun	1
2.	Full_name	Varchar	Maksimal 50 karakter	Merupakan field untuk nama lengkap customer	Daffa Raihandika
3.	Username	Varchar	Username harus available (belum ada yang menggunakan username tersebut) Username maksimal 15 karakter	Username akan digunakan untuk login. Username harus unik (tidak boleh ada username yang sama)	daffaraihandika
4.	Password	Varchar	Minimal 8 karakter	Password yang diinputkan oleh customer akan dienkripsi terlebih dahulu dengan cara hashing password untuk keamanan, sehingga password yang terdapat di database tidak dapat dibaca oleh pengelola database	sebelum dienkripsi : rakaganteng12* setelah dienkripsi : b10a8db164e0754105b7
5.	Email	Varchar	Harus terdapat domain	Email yang diinputkan customer ke dalam database harus valid	ferdyfauzan@gmail.com
6.	No_telepon	Varchar	Maksimal 15 karakter	Merupakan field untuk nomor telepon customer	0878653201287

			Karakter harus berupa angka tidak boleh ada karakter lain selain angka Harus diawali 0, 62 atau +62		
7	Role	Varchar	Field ini hanya dapat diisi dengan value ‘Admin’ atau ‘Customer’	Field untuk menandakan pemilik akun apakah admin atau customer	Customer
8	saldo	Integer	Saldo harus lebih besar sama dengan nol	Field untuk jumlah saldo yang dimiliki oleh user.	500000

Tabel 1.1.3 Tabel Event Search Product by Cateogory

Deskripsi				
Pembeli ingin mencari barang menggunakan kategori				
Respon Sistem				
No	Deskripsi Respon	Input	Output	Aturan

1	Customer dapat menemukan barang yang diinginkan dengan mencari kategori menggunakan fitur searching yang dimana fitur ini terdapat pada beranda, saat customer sudah memasukan nama kategori yang diinginkan maka aplikasi kami akan menampilkan info barang yang terdapat pada kategori tersebut.	Nama kategori	Barang yang terdapat pada kategori tersebut	Kategori yang dicari harus sesuai dengan kategori yang tersedia pada toko PickMeStore.

Kamus Data:

- Category : {id_kategori, nama_kategori}

Nama Tabel	User				
No.	**Nama Kolom**	**Tipe Data**	**Constraint**	**Deskripsi**	**Contoh Data**
1.	Id_Kategori	Integer	Primary key	Id_kategori merupakan primary key tabel category	3
2.	Nama_kategori	Varchar	Maksimal 30 karakter	Merupakan field untuk nama kategori	Hoodie

Tabel 1.1.4 Tabel Event Search Product

Deskripsi

Pembeli mencari barang yang diinginkan			
Respon Sistem			
Deskripsi Respon	Input	Output	Aturan
Customer dapat mencari barang yang diinginkan dengan fitur searching yang dimana fitur ini terdapat pada beranda , saat customer sudah memasukan nama barang yang diinginkan maka aplikasi kami akan menampilkan info barang yang diinginkan user	Nama barang	Customer dapat melihat info barang (nama barang ,harga barang, stok dan keterangan ketersediaan barang) yang dicari	Nama barang valid. Barang yang tersedia (Available) jika stok > 0

Kamus Data:

- product: {Id_product, Nama_barang,deskripsi, Harga, size,Id_Kategori, Stok}

Nama Tabel	Product				
No.	**Nama Kolom**	**Tipe Data**	**Constraint**	**Deskripsi**	**Contoh Data**

1.	Id_product	Integer	Primary key	Id_product merupakan primary key tabel product	3
2.	Nama_barang	Varchar	Tidak mengandung karakter khusus	Merupakan field untuk nama barang	Lightweight puffer jacket
3.	Harga	Integer	Nilai tidak boleh negatif	Merupakan field untuk harga barang	350000
4.	deskripsi	text		Field untuk deskripsi barang	Baju ini terbuat dari kualitas bahan terbaik
5.	Id_Kategori	Integer	Foreign key dari tabel Kategori	Untuk mengetahui barang tersebut tergolong ke dalam kategori apa	2
6.	Stok	Integer	Tidak boleh negatif	Merupakan field untuk jumlah barang yang tersedia di toko	14
7.	size	varchar	Nilai hanya boleh “S”,”M”,”L”, dan “XL”	Merupakan field untuk ukuran barang	M

Tabel 1.1.5 Tabel Event Checkout

Deskripsi
Pembeli membeli barang yang diinginkan

Respon Sistem			
Deskripsi Respon	Input	Output	Aturan
Saat customer melakukan checkout barang, customer akan diminta untuk memilih ukuran barang, jumlah barang yang ingin dibeli, dan memasukan alamat pengiriman. Setelah data telah sesuai , customer dapat menekan tombol submit untuk memesan barang saat semua informasi dan data yang diminta sudah lengkap	Jumlah barang yang ingin dibeli Alamat tujuan	Total harga Status transaksi	Nama barang valid. Barang yang tersedia (Available) jika stok > 0

Kamus Data:

- product: {Id_product, Nama_barang,deskripsi, Harga, size,Id_Kategori, Stok}
- Order_details : { Id_OrderDetails, Harga_Total, Id_User, status, Created_at, Modified_at}
- Order_Items : { Id_OrderItem, quantity, Total_Perproduk, Id_Product, Id_OrderDetails}
- Payment_details : { Id_PaymentDetails,status,created_at, Id_OrderDetails}
- Address : { id_address, alamat_lengkap, Provinsi, Kota, Kecamatan, kode_pos, id_user}

Nama Tabel	Product				
No.	**Nama Kolom**	**Tipe Data**	**Constraint**	**Deskripsi**	**Contoh Data**
1.	Id_product	Integer	Primary key	Id_product merupakan primary key tabel product	3
2.	Nama_barang	Varchar	Tidak mengandung karakter khusus	Merupakan field untuk nama barang	Lightweight puffer jacket
3.	Harga	Integer	Nilai tidak boleh negatif	Merupakan field untuk harga barang	350000
4.	deskripsi	text		Field untuk deskripsi barang	Baju ini terbuat dari kualitas bahan terbaik
5.	Id_Kategori	Integer	Foreign key dari tabel Kategori	Untuk mengetahui barang tersebut tergolong ke dalam kategori apa	2
6.	Stok	Integer	Tidak boleh negatif	Merupakan field untuk jumlah barang yang tersedia di toko	14
7.	size	varchar	Nilai hanya boleh “S”,”M”,”L”, dan “XL”	Merupakan field untuk ukuran barang	M

Nama Tabel	Address				
No.	**Nama Kolom**	**Tipe Data**	**Constraint**	**Deskripsi**	**Contoh Data**

1.	Id_Alamat	Integer	Primary key	Id_Alamat merupakan primary key tabel address	2
2.	Provinsi	Varchar	Maksimal 30 karakter Tidak mengandung karakter khusus	Provinsi yang ada pada field ini, yaitu provinsi yang ada di Indonesia	Jawa Barat
3.	Kota	Varchar	Maksimal 50 karakter Tidak mengandung karakter khusus	Kota yang ada pada field ini, yaitu kota/kabupaten yang terdapat pada provinsi customer	Bandung
4.	Kecamatan	Varchar	Maksimal 30 karakter Tidak mengandung karakter khusus	Kecamatan yang ada pada field ini, yaitu kecamatan yang terdapat pada kota/kabupaten customer	Parongpong
5.	Kode_pos	Char	Maksimal 6 karakter Karakter harus berupa angka, tidak boleh ada karakter lain selain angka	Kode pos yang ada pada field ini, yaitu kode pos yang terdapat pada kecamatan customer	40619
6.	Alamat_lengkap	Varchar	Maksimal 50 karakter	Merupakan field untuk alamat lengkap customer	Komplek Serra Valley De Lima blok D-6
7.	Id_User	Integer	Foreign key dari tabel user	Untuk mengetahui informasi akun yang memiliki alamat tersebut	1

Nama	order_details				
No.	**Nama Kolom**	**Tipe Data**	**Constraint**	**Deskripsi**	**Contoh Data**
1.	Id_OrderDetails	Integer	Primary key	Id_OrderDetail merupakan primary key tabel order_details	2

2.	Harga_Total	Integer	harga total tidak boleh negatif	merupakan field untuk harga total yang di beli user.	78.000
3.	Id_user	Integer	foreign key	Untuk mengetahui informasi akun yang memiliki alamat tersebut	1
4.	Status	Varchar	maksimal 20 karakter hanya berisi nilai “Canceled”,”Ordered”, dan “Expired”	merupakan field untuk status order details yaitu canceled,expired atau ordered	Ordered
5.	Created_at	Timestamp	Waktu dan tanggal sesuai zona waktu sesi local user	Merupakan field untuk mengetahui kapan user membuat order details.	11-DES-22 07.28.32.374935 AM +00:00
6.	Modified_at	Timestamp	Waktu dan tanggal sesuai zona waktu sesi local user	Merupakan field untuk mengetahui kapan user memodifikasi order detailsnya	11-DES-22 07.28.32.374935 AM +00:00

Nama	order_Items				
No.	**Nama Kolom**	**Tipe Data**	**Constraint**	**Deskripsi**	**Contoh Data**
1.	Id_OrderItem	Integer	Primary key	Id_OrderItem merupakan primary key tabel order_items	2
2.	Quantity	Integer	nilai quantity tidak boleh negatif	merupakan field untuk quantity	14
3.	Total_Perproduk	Integer	tidak boleh negatif	field untuk mengetahui harga barang per produk	20.000

4.	Id_Product	Integer	foreign key dari table product	Untuk menentukan produk mana yang akan dipesan	23
5.	Id_OrderDetails	Integer	foreign key dari table order_details	berfungsi untuk menyatukan produk produk yang dipesan menjadi satu pesanan oleh 1 user.	5

Nama	payment_details				
No.	**Nama Kolom**	**Tipe Data**	**Constraint**	**Deskripsi**	**Contoh Data**
1.	Id_PaymentDetails	Integer	Primary key	Id_PaymentDeatils merupakan primary key tabel payment_details	2
2.	Status	Varchar	hanya berisi nilai “Lunas” dan “tidak lunas”	merupakan field untuk status	Lunas
3.	Created_at	Timestamp	Waktu dan tanggal sesuai zona waktu sesi local user	Merupakan field untuk mengetahui kapan user melakukan pembayaran	11-DES-22 07.28.32.374935 AM +00:00
4.	Id_OrderDetails	Integer	foreign key dari table product	Untuk mengetahui details order yang akan dibayar oleh customer	7

Tabel 1.1.6 Tabel Event Penanganan Stok

Deskripsi

Penanganan stok saat terjadi transaksi .				
Respon Sistem				
No	Deskripsi Respon	Input	Output	Aturan
1	Saat user melakukan checkout dan stok barang masih tersedia, sistem akan mengurangi stok yang tersedia pada database barang dan stok barang terbaru akan ditampilkan pada informasi barang di aplikasi. Saat user melakukan checkout dan stok barang tidak tersedia, user tidak dapat melakukan checkout.	Jumlah barang yang akan dibeli	status transaksi berhasil/gagal. menampilkan informasi barang dengan jumlah stok terbaru	Stok barang >= jumlah barang yang ingin dibeli
2	Saat ada 2 atau lebih transaksi checkout yang dilakukan dalam waktu bersamaan , maka transaksi yang paling cepat melakukan pembayaran akan langsung diproses dan transaksi lain harus menunggu transaksi yang pertama selesai untuk melakukan pembayaran . misalkan : sisa stok barang hanya tersedia untuk ukuran M sebanyak 5 pcs , lalu ada	Jumlah barang yang akan dibeli	status transaksi berhasil/gagal. menampilkan informasi barang dengan jumlah stok terbaru	Stok barang >= jumlah barang yang ingin dibeli , dan pembayaran berhasil dilakukan.

	2 user yang ingin membeli barang tersebut di waktu yang bersamaan dan masing masing ingin membeli 3 pcs , maka user pertama yang melakukan checkout akan langsung diproses , dan user kedua akan diminta menunggu , apabila pembayaran user pertama berhasil maka user kedua tidak dapat melakukan pembayaran (status transaksi gagal) karena jumlah stok barang yang tersedia < jumlah barang yang ingin dibeli . namun apabila pembayaran user pertama gagal , maka user kedua dapat melakukan pembayaran.			

Kamus Data:

- product: {Id_product, Nama_barang,deskripsi, Harga, size,Id_Kategori, Stok}
- Order_Items : { Id_OrderItem, quantity, Total_Perproduk, Id_Product, Id_OrderDetails}

Nama Tabel	Product				
No.	**Nama Kolom**	**Tipe Data**	**Constraint**	**Deskripsi**	**Contoh Data**
1.	Id_product	Integer	Primary key	Id_product merupakan primary key tabel product	3
2.	Nama_barang	Varchar	Tidak mengandung karakter khusus	Merupakan field untuk nama barang	Lightweight puffer jacket
3.	Harga	Integer	Nilai tidak boleh negatif	Merupakan field untuk harga barang	350000

4.	deskripsi	text		Field untuk deskripsi barang	Baju ini terbuat dari kualitas bahan terbaik
5.	Id_Kategori	Integer	Foreign key dari tabel Kategori	Untuk mengetahui barang tersebut tergolong ke dalam kategori apa	2
6.	Stok	Integer	Tidak boleh negatif	Merupakan field untuk jumlah barang yang tersedia di toko	14
7.	size	varchar	Nilai hanya boleh “S”,”M”,”L”, dan “XL”	Merupakan field untuk ukuran barang	M

Nama	order_Items				
No.	**Nama Kolom**	**Tipe Data**	**Constraint**	**Deskripsi**	**Contoh Data**
1.	Id_OrderItem	Integer	Primary key	Id_OrderItem merupakan primary key tabel order_items	2
2.	Quantity	Integer	nilai quantity tidak boleh negatif	merupakan field untuk quantity	14
3.	Total_Perproduk	Integer	tidak boleh negatif	field untuk mengetahui harga barang per produk	20.000
4.	Id_Product	Integer	foreign key dari table product	Untuk menentukan produk mana yang akan dipesan	23
5.	Id_OrderDetails	Integer	foreign key dari table order_details	berfungsi untuk menyatukan produk produk yang dipesan menjadi satu pesanan oleh 1 user.	5

Tabel 1.1.7 Tabel Event Top Up Saldo

Deskripsi				
Customer ingin melakukan top up saldo				
Respon Sistem				
No	Deskripsi Respon	Input	Output	Aturan
1	Saat customer ingin melakukan top up saldo , customer harus memasukan jumlah saldo yang ingin ditambahkan .	username password jumlah saldo yang ingin ditambahkan	update saldo terbaru	saldo yang ditambahkan > 0

Kamus Data:

- user: {Id_User, Username, Password, full_name, Email, No_telp, saldo, Role}

Nama Tabel	User				
No.	**Nama Kolom**	**Tipe Data**	**Constraint**	**Deskripsi**	**Contoh Data**
1.	Id_User	Integer	Primary key	Id_Customer merupakan primary key tabel Akun	1
2.	Full_name	Varchar	Maksimal 50 karakter	Merupakan field untuk nama lengkap customer	Daffa Raihandika
3.	Username	Varchar	Username harus available (belum ada yang menggunakan username tersebut)	Username akan digunakan untuk login. Username harus unik	daffaraihandika

			Username maksimal 15 karakter	(tidak boleh ada username yang sama)	
4.	Password	Varchar	Minimal 8 karakter	Password yang diinputkan oleh customer akan dienkripsi terlebih dahulu dengan cara hashing password untuk keamanan, sehingga password yang terdapat di database tidak dapat dibaca oleh pengelola database	sebelum dienkripsi : rakaganteng12* setelah dienkripsi : b10a8db164e0754105b7
5.	Email	Varchar	Harus terdapat domain	Email yang diinputkan customer ke dalam database harus valid	ferdyfauzan@gmail.com
6.	No_telepon	Varchar	Maksimal 15 karakter Karakter harus berupa angka tidak boleh ada karakter lain selain angka Harus diawali 0, 62 atau +62	Merupakan field untuk nomor telepon customer	0878653201287
7	Role	Varchar	Field ini hanya dapat diisi dengan value ‘Admin’ atau ‘Customer’	Field untuk menandakan pemilik akun apakah admin atau customer	Customer
8	saldo	Integer	Saldo harus lebih besar sama dengan nol	Field untuk jumlah saldo yang dimiliki oleh user.	500000

Tabel 1.1.8 Tabel Add Stock Product

Deskripsi				
Admin ingin menambahkan stok pada produk				
Respon Sistem				
No	Deskripsi Respon	Input	Output	Aturan
1	Saat admin ingin menambahkan stok pada suatu produk, admin harus menginputkan jumlah stok yang ingin ditambahkan ke suatu produk.	Jumlah stok	Jumlah stok terbaru	Jumlah stok yang ditambahkan harus > 0

Kamus Data:

- product: {Id_product, Nama_barang,deskripsi, Harga, size,Id_Kategori, Stok}

Nama Tabel	Product				
No.	**Nama Kolom**	**Tipe Data**	**Constraint**	**Deskripsi**	**Contoh Data**
1.	Id_product	Integer	Primary key	Id_product merupakan primary key tabel product	3
2.	Nama_barang	Varchar	Tidak mengandung karakter khusus	Merupakan field untuk nama barang	Lightweight puffer jacket
3.	Harga	Integer	Nilai tidak boleh negatif	Merupakan field untuk harga barang	350000

4.	deskripsi	text		Field untuk deskripsi barang	Baju ini terbuat dari kualitas bahan terbaik
5.	Id_Kategori	Integer	Foreign key dari tabel Kategori	Untuk mengetahui barang tersebut tergolong ke dalam kategori apa	2
6.	Stok	Integer	Tidak boleh negatif	Merupakan field untuk jumlah barang yang tersedia di toko	14
7.	size	varchar	Nilai hanya boleh “S”,”M”,”L”, dan “XL”	Merupakan field untuk ukuran barang	M

Tabel 1.1.9 Tabel Event Cancel Order

Deskripsi				
Customer ingin membatalkan pesanan				
Respon Sistem				
No	Deskripsi Respon	Input	Output	Aturan
1	Saat customer ingin membatalkan pesanan	Username password	Status order menjadi canceled	Barang di cancel saat belum dibayar
Kamus Data: user: {Id_User, Username, Password, full_name, Email, No_telp, saldo, Role} Order_details : { Id_OrderDetails, Harga_Total, Id_User, status, Created_at, Modified_at}				

Nama Tabel	User				
No.	**Nama Kolom**	**Tipe Data**	**Constraint**	**Deskripsi**	**Contoh Data**
1.	Id_User	Integer	Primary key	Id_Customer merupakan primary key tabel Akun	1
2.	Full_name	Varchar	Maksimal 50 karakter	Merupakan field untuk nama lengkap customer	Daffa Raihandika
3.	Username	Varchar	Username harus available (belum ada yang menggunakan username tersebut) Username maksimal 15 karakter	Username akan digunakan untuk login. Username harus unik (tidak boleh ada username yang sama)	daffaraihandika
4.	Password	Varchar	Minimal 8 karakter	Password yang diinputkan oleh customer akan dienkripsi terlebih dahulu dengan cara hashing password untuk keamanan, sehingga password yang terdapat di database tidak dapat dibaca oleh pengelola database	sebelum dienkripsi : rakaganteng12* setelah dienkripsi : b10a8db164e0754105b7
5.	Email	Varchar	Harus terdapat domain	Email yang diinputkan customer ke dalam database harus valid	ferdyfauzan@gmail.com

6.	No_telepon	Varchar	Maksimal 15 karakter Karakter harus berupa angka tidak boleh ada karakter lain selain angka Harus diawali 0, 62 atau +62	Merupakan field untuk nomor telepon customer	0878653201287
7	Role	Varchar	Field ini hanya dapat diisi dengan value ‘Admin’ atau ‘Customer’	Field untuk menandakan pemilik akun apakah admin atau customer	Customer
8	saldo	Integer	Saldo harus lebih besar sama dengan nol	Field untuk jumlah saldo yang dimiliki oleh user.	500000

Nama	order_details				
No.	**Nama Kolom**	**Tipe Data**	**Constraint**	**Deskripsi**	**Contoh Data**
1.	Id_OrderDetails	Integer	Primary key	Id_OrderDetail merupakan primary key tabel order_details	2
2.	Harga_Total	Integer	harga total tidak boleh negatif	merupakan field untuk harga total yang di beli user.	78.000
3.	Id_user	Integer	foreign key	Untuk mengetahui informasi akun yang memiliki alamat tersebut	1
4.	Status	Varchar	maksimal 20 karakter hanya berisi nilai “Canceled”,”Ordered”, dan “Expired”	merupakan field untuk status order details yaitu canceled,expired atau ordered	Ordered

5.	Created_at	Timestamp	Waktu dan tanggal sesuai zona waktu sesi local user	Merupakan field untuk mengetahui kapan user membuat order details.	11-DES-22 07.28.32.374935 AM +00:00
6.	Modified_at	Timestamp	Waktu dan tanggal sesuai zona waktu sesi local user	Merupakan field untuk mengetahui kapan user memodifikasi order detailsnya	11-DES-22 07.28.32.374935 AM +00:00

Tabel 1.1.10 Tabel Event Change Password

Deskripsi				
User ingin mengganti password				
Respon Sistem				
No	Deskripsi Respon	Input	Output	Aturan
1	Saat user ingin mengganti password , user harus memasukan username,	username password lama password baru	password berhasil diubah	username dan password harus valid

	password lama, dan password baru			

Kamus Data:

- user: {Id_User, Username, Password, full_name, Email, No_telp, saldo, Role}

Nama Tabel	User				
No.	**Nama Kolom**	**Tipe Data**	**Constraint**	**Deskripsi**	**Contoh Data**
1.	Id_User	Integer	Primary key	Id_Customer merupakan primary key tabel Akun	1
2.	Full_name	Varchar	Maksimal 50 karakter	Merupakan field untuk nama lengkap customer	Daffa Raihandika
3.	Username	Varchar	Username harus available (belum ada yang menggunakan username tersebut) Username maksimal 15 karakter	Username akan digunakan untuk login. Username harus unik (tidak boleh ada username yang sama)	daffaraihandika
4.	Password	Varchar	Minimal 8 karakter	Password yang diinputkan oleh customer akan dienkripsi terlebih dahulu dengan cara hashing password untuk keamanan, sehingga password yang terdapat di database tidak dapat dibaca oleh pengelola database	sebelum dienkripsi : rakaganteng12* setelah dienkripsi : b10a8db164e0754105b7

5.	Email	Varchar	Harus terdapat domain	Email yang diinputkan customer ke dalam database harus valid	ferdyfauzan@gmail.com
6.	No_telepon	Varchar	Maksimal 15 karakter Karakter harus berupa angka tidak boleh ada karakter lain selain angka Harus diawali 0, 62 atau +62	Merupakan field untuk nomor telepon customer	0878653201287
7	Role	Varchar	Field ini hanya dapat diisi dengan value ‘Admin’ atau ‘Customer’	Field untuk menandakan pemilik akun apakah admin atau customer	Customer
8	saldo	Integer	Saldo harus lebih besar sama dengan nol	Field untuk jumlah saldo yang dimiliki oleh user.	500000

Tabel 1.1.11 Tabel Event Delete Product

Deskripsi				
Admin ingin menghapus produk				
Respon Sistem				
No	Deskripsi Respon	Input	Output	Aturan

1	Saat admin ingin menghapus produk, admin harus menginputkan nama barang dan ukuran barang yang ingin dihapus.	Nama barang Ukuran barang	Produk berhasil dihapus	Produk yang ingin dihapus harus sudah ada pada table produk.

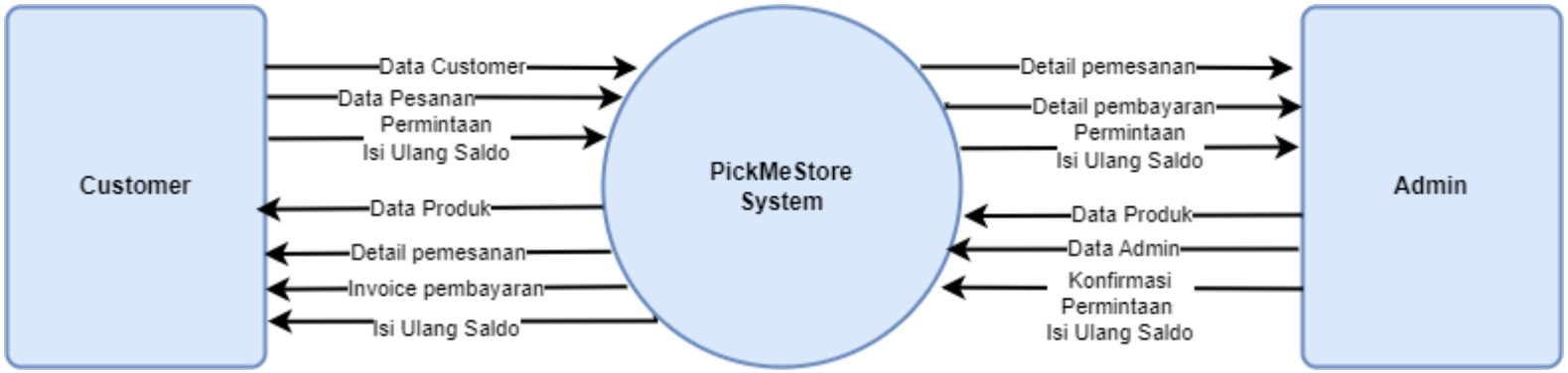
Kamus Data:

- product: {Id_product, Nama_barang,deskripsi, Harga, size,Id_Kategori, Stok}

Nama Tabel	Product				
No.	**Nama Kolom**	**Tipe Data**	**Constraint**	**Deskripsi**	**Contoh Data**
1.	Id_product	Integer	Primary key	Id_product merupakan primary key tabel product	3
2.	Nama_barang	Varchar	Tidak mengandung karakter khusus	Merupakan field untuk nama barang	Lightweight puffer jacket
3.	Harga	Integer	Nilai tidak boleh negatif	Merupakan field untuk harga barang	350000
4.	deskripsi	text		Field untuk deskripsi barang	Baju ini terbuat dari kualitas bahan terbaik
5.	Id_Kategori	Integer	Foreign key dari tabel Kategori	Untuk mengetahui barang tersebut tergolong ke dalam kategori apa	2
6.	Stok	Integer	Tidak boleh negatif	Merupakan field untuk jumlah barang yang tersedia di toko	14
7.	size	varchar	Nilai hanya boleh “S”,”M”,”L”, dan “XL”	Merupakan field untuk ukuran barang	M

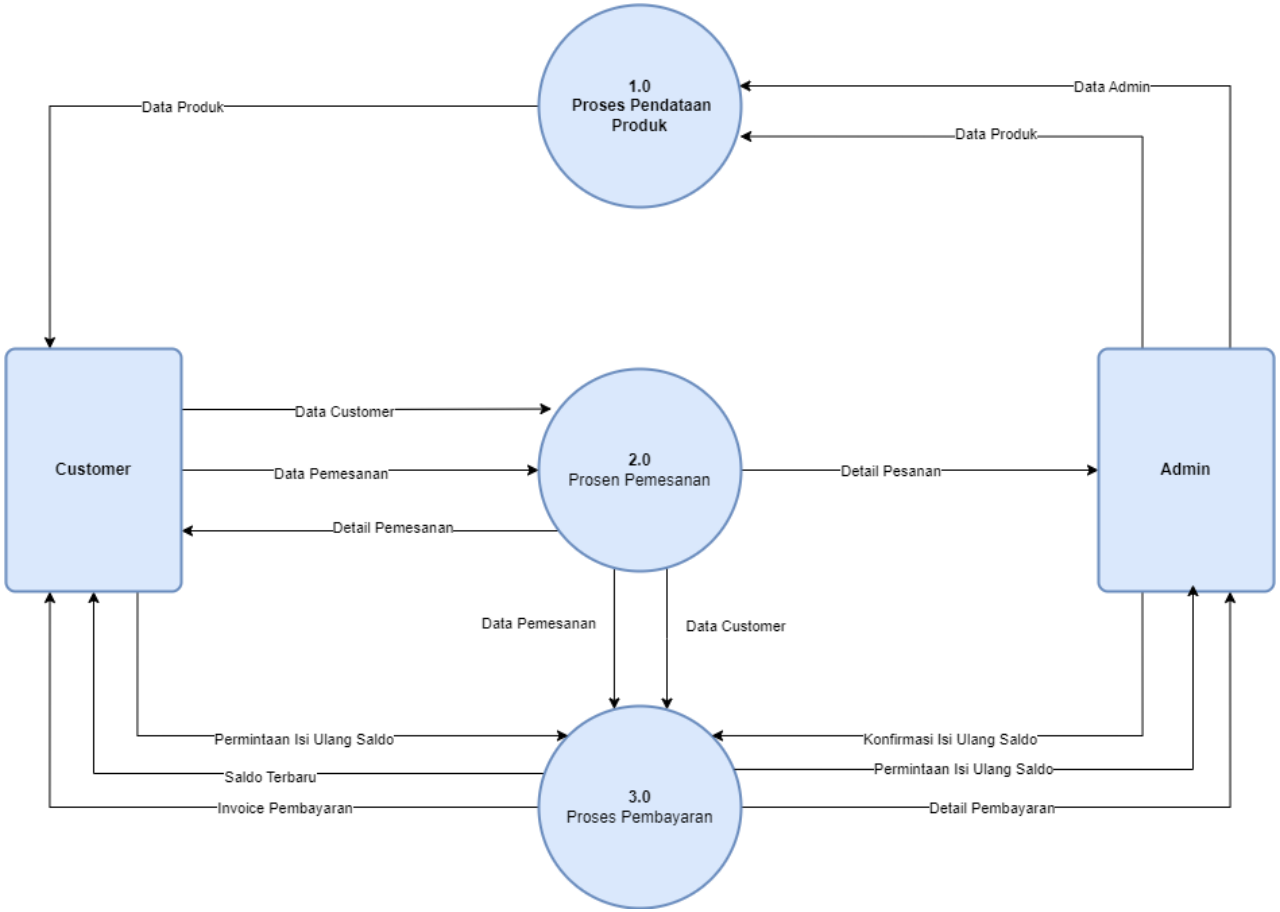
2.1.1 Context Diagram, DFD

2.1.1.1 Context Diagram (DFD Level 0)



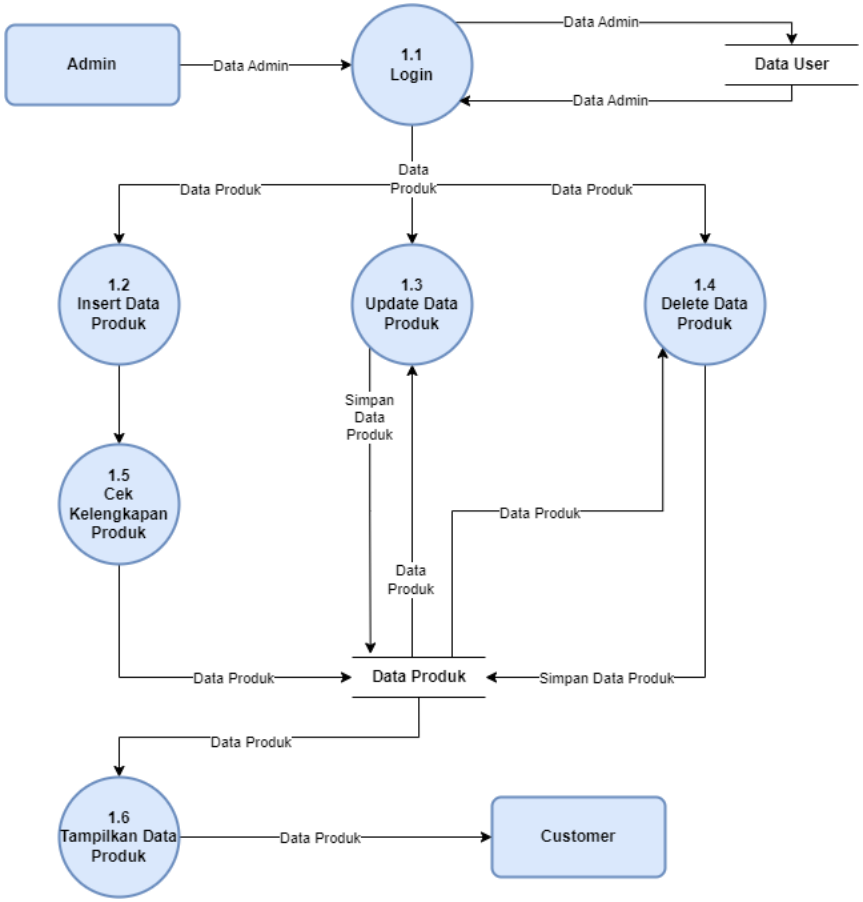
Gambar 1.1.2 Context Diagram dari Sistem Aplikasi PickMeStore

2.1.1.2 DFD Level 1

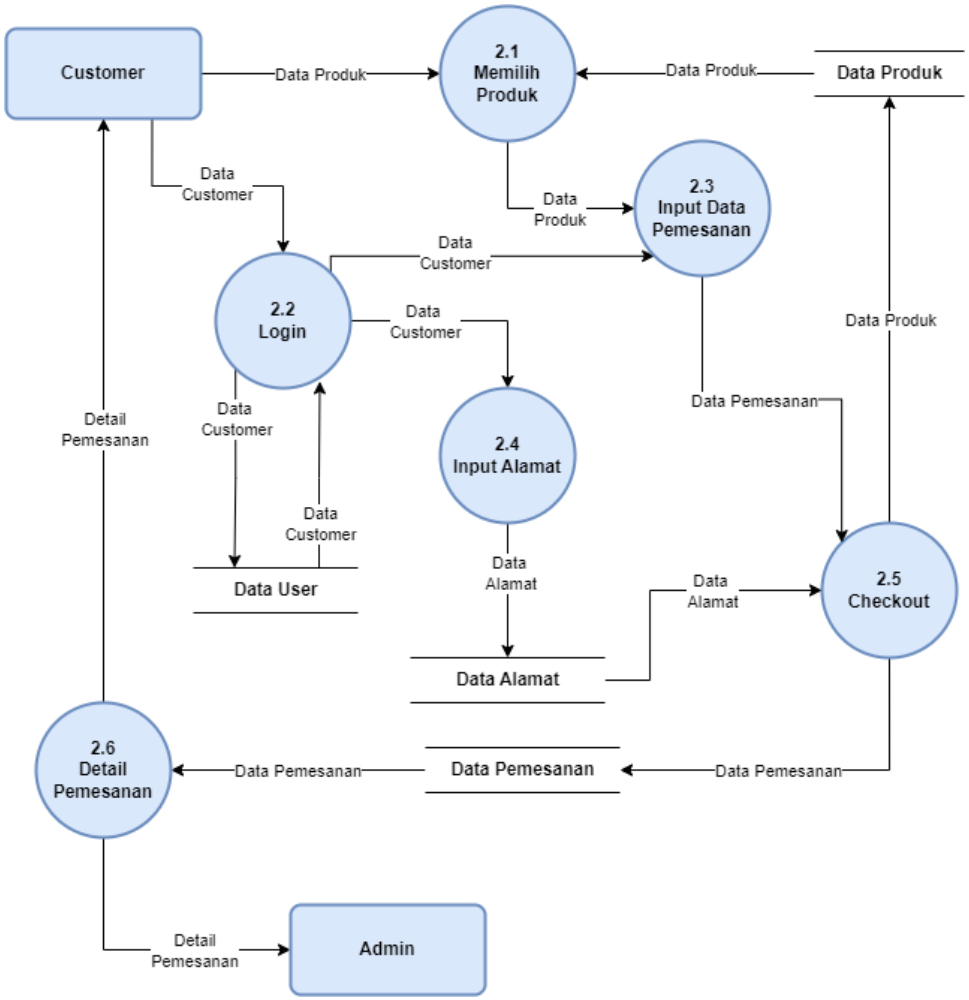


Gambar 1.1.3 DFD Level 0 dari Sistem Aplikasi PickMeStore

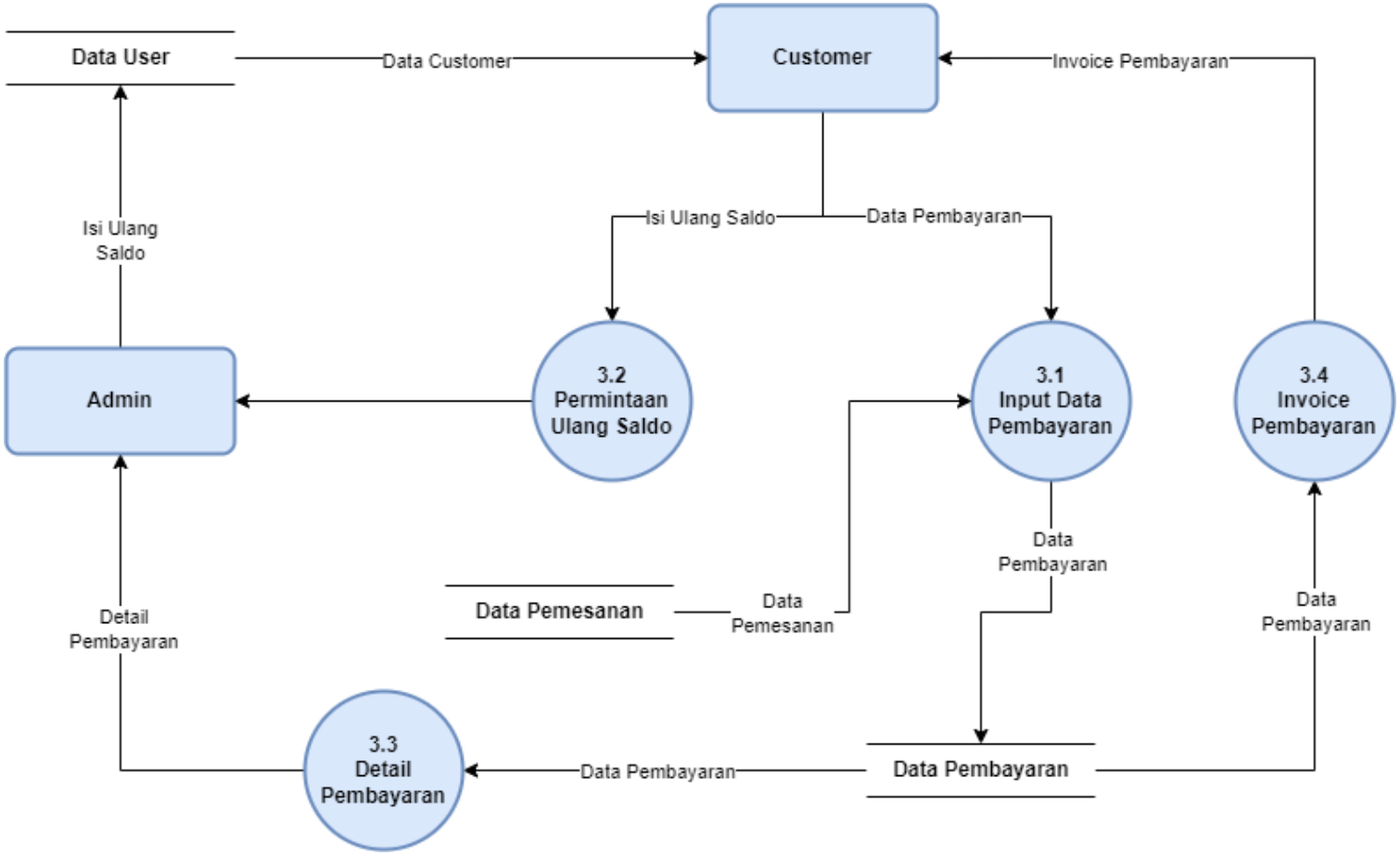
2.1.1.3 DFD Level 2



Gambar 1.1.4 DFD Level 2 dari Process 1.0



Gambar 1.1.5 DFD Level 2 dari Process 2.0

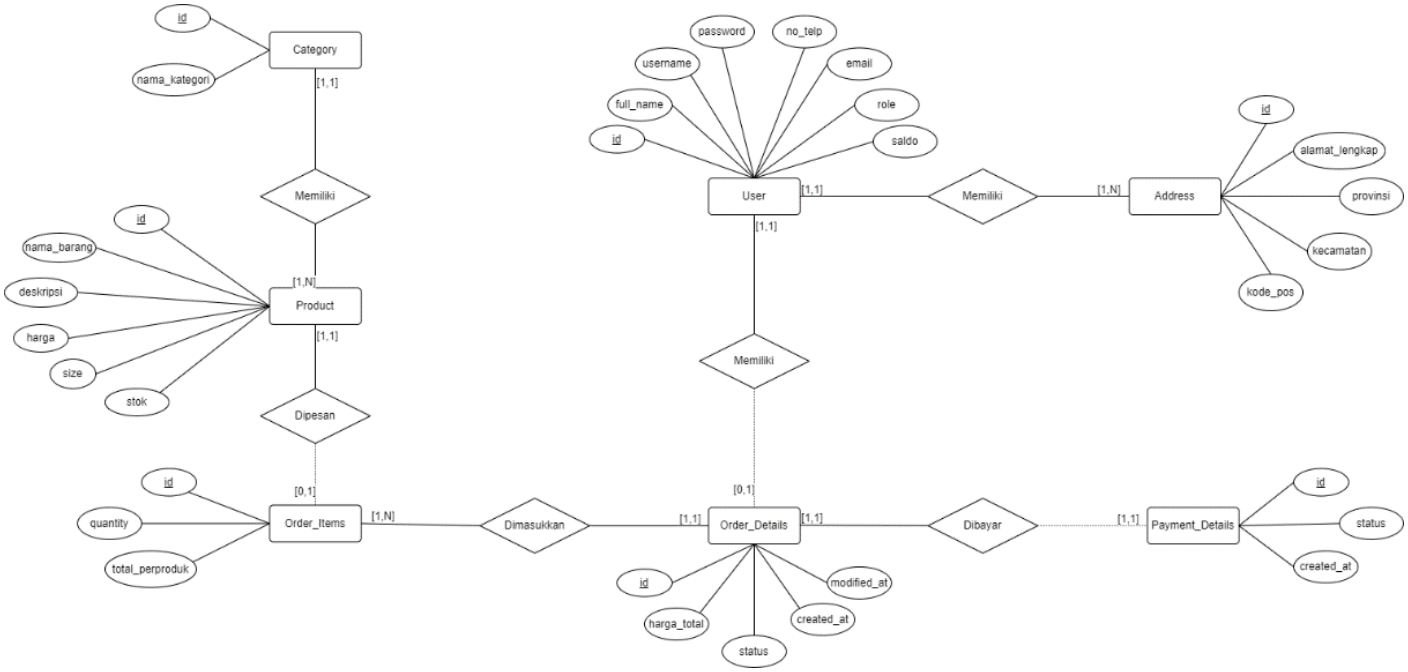


Gambar 1.1.6 DFD Level 2 dari Process 3.0

BAB III
PERANCANGAN

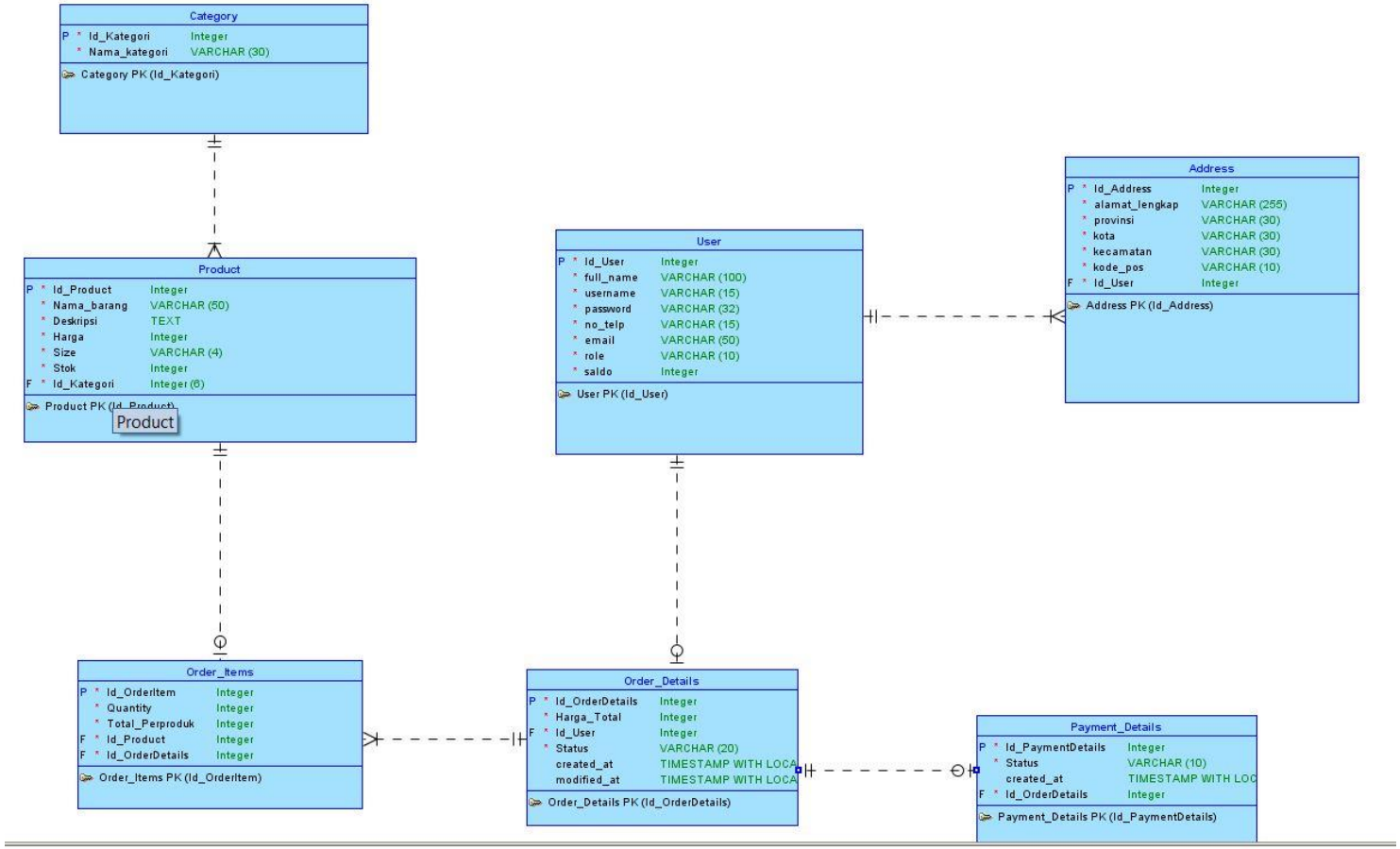
3.1 Model Skema Logical

3.1.1 Notasi Chen



Gambar 3.1.1 Skema Logical Sistem Aplikasi PickMeStore dengan Notasi Chen

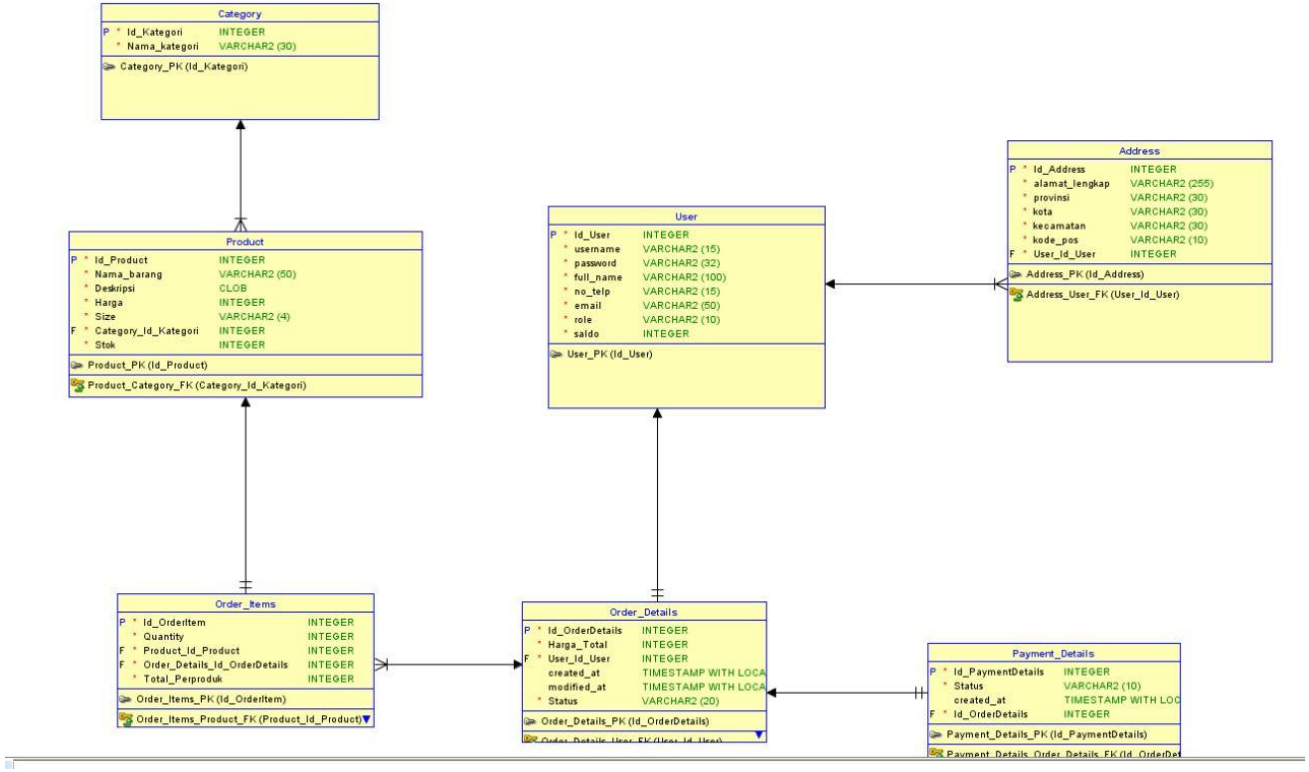
3.1.2 Oracle Data Modeler



Gambar 3.1.2 Skema Logical Sistem Aplikasi PickMeStore dengan Oracle Data Modeler

3.2 Model Skema Relational

3.2.1 Relational (Physical) Data Model



Gambar 3.2.1 Skema Relational Sistem Aplikasi PickMeStore dengan Oracle Data Modeler

3.2.2 DDL

Tabel 3.2.1 DDL Tabel User

Nama Tabel	User
Deskripsi	Membuat tabel "User" yang bertujuan untuk menampung data akun user, baik admin maupun customer.

Script SQL

```
CREATE TABLE "User" (
    id_user   INTEGER GENERATED BY DEFAULT AS
IDENTITY NOT NULL PRIMARY KEY,
    username  VARCHAR2(15) NOT NULL,
    password  VARCHAR2(32) NOT NULL,
    full_name VARCHAR2(100) NOT NULL,
    no_telp   VARCHAR2(15) NOT NULL,
    email     VARCHAR2(50) NOT NULL,
    role      VARCHAR2(10) NOT NULL,
    saldo     INTEGER NOT NULL
);

ALTER TABLE "User"
  MODIFY Saldo INTEGER DEFAULT 0;


ALTER TABLE "User"
    ADD CONSTRAINT Check_Saldo CHECK (saldo >= 0
);

CREATE UNIQUE INDEX username__idx ON
    "User" (
        username
    ASC );

CREATE UNIQUE INDEX email__idx ON
    "User" (
        email
    ASC );

CREATE UNIQUE INDEX notelp__idx ON
    "User" (
        no_telp
    ASC );

ALTER TABLE "User"
ADD CONSTRAINT email_format CHECK (REGEXP_LIKE
(email, '^[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Za-
z]{2,4}$')) ENABLE NOVALIDATE;

ALTER TABLE "User"
```

```
ADD CONSTRAINT phone_format CHECK (REGEXP_LIKE
(no_telp, '^(\+62|62|0)8[1-9][0-9]{6,9}$')) ENABLE
NOVALIDATE;
```

Screenshot Hasil

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	ID_USER	NUMBER(38,0)	No	"DAFFARAIHANDIKA4"."ISEQ$$_75170".nextval	1	(null)
2	USERNAME	VARCHAR2(15 BYTE)	No	(null)	2	(null)
3	PASSWORD	VARCHAR2(32 BYTE)	No	(null)	3	(null)
4	FULL_NAME	VARCHAR2(100 BYTE)	No	(null)	4	(null)
5	NO_TELP	VARCHAR2(15 BYTE)	No	(null)	5	(null)
6	EMAIL	VARCHAR2(50 BYTE)	No	(null)	6	(null)
7	ROLE	VARCHAR2(10 BYTE)	No	(null)	7	(null)
8	SALDO	NUMBER(38,0)	No	0	8	(null)

Tabel 3.2.2 DDL Tabel Address

Nama Tabel	Address
Deskripsi	Membuat tabel “Address” yang bertujuan untuk menampung data alamat yang dimiliki user

Script SQL

```
CREATE TABLE address (
    id_address      INTEGER GENERATED BY DEFAULT AS
IDENTITY NOT NULL PRIMARY KEY,
    alamat_lengkap VARCHAR2(255) NOT NULL,
    provinsi        VARCHAR2(30) NOT NULL,
    kota            VARCHAR2(30) NOT NULL,
    kecamatan       VARCHAR2(30) NOT NULL,
    kode_pos        VARCHAR2(10) NOT NULL,
    id_user    INTEGER NOT NULL
);

ALTER TABLE address
    ADD CONSTRAINT address_user_fk FOREIGN KEY (
id_user )
        REFERENCES "User" ( id_user )
        ON DELETE CASCADE;
```

Screenshot Hasil

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	ID_ADDRESS	NUMBER(38,0)	No	"DAFFARAIHANDIKA4"."ISEQ$$_75149".nextval	1	(null)
2	ALAMAT_LENGKAP	VARCHAR2(255 BYTE)	No	(null)	2	(null)
3	PROVINSI	VARCHAR2(30 BYTE)	No	(null)	3	(null)
4	KOTA	VARCHAR2(30 BYTE)	No	(null)	4	(null)
5	KECAMATAN	VARCHAR2(30 BYTE)	No	(null)	5	(null)
6	KODE_POS	VARCHAR2(10 BYTE)	No	(null)	6	(null)
7	ID_USER	NUMBER(38,0)	No	(null)	7	(null)

Tabel 3.2.3 DDL Tabel Product

Nama Tabel	Product

Deskripsi	Membuat tabel “Product” yang bertujuan untuk menampung data produk

Script SQL

```
CREATE TABLE product (
    id_product            INTEGER GENERATED BY
DEFAULT AS IDENTITY NOT NULL PRIMARY KEY,
    nama_barang           VARCHAR2(50) NOT NULL,
    deskripsi             CLOB NOT NULL,
    harga                 INTEGER NOT NULL,
    ukuran                VARCHAR2(4) NOT NULL,
    id_kategori           INTEGER NOT NULL,
    stok                  INTEGER NOT NULL
);

ALTER TABLE Product
    ADD CONSTRAINT Cek_harga CHECK (harga > 0);

ALTER TABLE Product
    ADD CONSTRAINT Cek_ukuran CHECK (ukuran = 'S'
OR ukuran = 'M' OR ukuran = 'L' OR ukuran = 'XL');

ALTER TABLE Product
    ADD CONSTRAINT Cek_stok CHECK (stok >= 0);

ALTER TABLE product
    ADD CONSTRAINT product_category_fk FOREIGN KEY
( id_kategori )
        REFERENCES category ( id_kategori )
        ON DELETE CASCADE;
```

Screenshot Hasil

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	ID_ADDRESS	NUMBER(38,0)	No	"DAFFARAIHANDIKA4"."ISEQ$$_75149".nextval	1	(null)
2	ALAMAT_LENGKAP	VARCHAR2(255 BYTE)	No	(null)	2	(null)
3	PROVINSI	VARCHAR2(30 BYTE)	No	(null)	3	(null)
4	KOTA	VARCHAR2(30 BYTE)	No	(null)	4	(null)
5	KECAMATAN	VARCHAR2(30 BYTE)	No	(null)	5	(null)
6	KODE_POS	VARCHAR2(10 BYTE)	No	(null)	6	(null)
7	ID_USER	NUMBER(38,0)	No	(null)	7	(null)

Tabel 3.2.4 DDL Tabel Category

Nama Tabel	Category
Deskripsi	Membuat tabel “Category” yang bertujuan untuk menampung data kategori produk

Script SQL

```
CREATE TABLE category (
    id_kategori   INTEGER GENERATED BY DEFAULT AS
IDENTITY NOT NULL PRIMARY KEY,
    nama_kategori VARCHAR2(30) NOT NULL
);
```

Screenshot Hasil

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	ID_KATEGORI	NUMBER(38,0)	No	"DAFFARAIHANDIKA4"."ISEQ$$_75152".nextval	1	(null)
2	NAMA_KATEGORI	VARCHAR2(30 BYTE)	No	(null)	2	(null)

Tabel 3.2.5 DDL Tabel Order_Items

Nama Tabel	Order_Items
Deskripsi	Membuat tabel "Order_items" yang bertujuan untuk menampung jumlah produk dan harga yang dipesan per produk

Script SQL

```
CREATE TABLE order_items (
    id_orderitem                    INTEGER
GENERATED BY DEFAULT AS IDENTITY NOT NULL PRIMARY
KEY,
    quantity                        INTEGER NOT
NULL,
    id_product              INTEGER NOT NULL,
    id_orderdetails                 INTEGER NOT
NULL,
    total_perproduk                 INTEGER NOT NULL
);

ALTER TABLE order_items
  MODIFY id_orderdetails INTEGER NULL;

CREATE UNIQUE INDEX order_items__idx ON
    order_items (
        id_product
    ASC );

ALTER TABLE order_items
    ADD CONSTRAINT order_items_product_fk FOREIGN
KEY ( id_product )
        REFERENCES product ( id_product )
        ON DELETE CASCADE;
```

Screenshot Hasil

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	ID_ORDERITEM	NUMBER(38,0)	No	"DAFFARAIHANDIKA4"."ISEQ$$_75159".nextval	1	(null)
2	QUANTITY	NUMBER(38,0)	No	(null)	2	(null)
3	ID_PRODUCT	NUMBER(38,0)	No	(null)	3	(null)
4	ID_ORDERDETAILS	NUMBER(38,0)	No	(null)	4	(null)
5	TOTAL_PERPRODUK	NUMBER(38,0)	No	(null)	5	(null)

Tabel 3.2.6 DDL Tabel Order_Details

Nama Tabel	Order_Details

Deskripsi	Membuat tabel “Order_details” yang bertujuan untuk menampung total harga produk yang dipesan oleh customer

Script SQL

```
CREATE TABLE order_details (
    id_orderdetails                    INTEGER
GENERATED BY DEFAULT AS IDENTITY NOT NULL PRIMARY
KEY,
    harga_total                        INTEGER NOT
NULL,
    id_user                            INTEGER NOT
NULL,
    status                             VARCHAR2(20)
NOT NULL,
    created_at                         TIMESTAMP
WITH LOCAL TIME ZONE,
    modified_at                        TIMESTAMP
WITH LOCAL TIME ZONE
);

ALTER TABLE order_details
    ADD CONSTRAINT Check_status_order CHECK
(status =  'Expired' OR status = 'Canceled' OR
status = 'Ordered');

ALTER TABLE order_details
  MODIFY status DEFAULT 'Ordered';

ALTER TABLE order_details
  MODIFY id_paymentdetails INTEGER NULL;

CREATE UNIQUE INDEX order_details__idx ON
    order_details (
        id_user
    ASC );

CREATE UNIQUE INDEX order_details_idx ON
    order_details (
        id_paymentdetails
    ASC );

ALTER TABLE order_details
    ADD CONSTRAINT order_details_user_fk FOREIGN
KEY ( id_user )
        REFERENCES "User" ( id_user )
        ON DELETE CASCADE;
```

Screenshot Hasil

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENT
1	ID_ORDERDETAILS	NUMBER(38,0)	No	"DAFFARAIHANDIKA4"."ISEQ$$_75156".nextval	1	(null)
2	HARGA_TOTAL	NUMBER(38,0)	No	(null)	2	(null)
3	ID_USER	NUMBER(38,0)	No	(null)	3	(null)
4	STATUS	VARCHAR2(20 BYTE)	No	'Ordered'	4	(null)
5	CREATED_AT	TIMESTAMP(6) WITH LOCAL TIME ZONE	Yes	(null)	5	(null)
6	MODIFIED_AT	TIMESTAMP(6) WITH LOCAL TIME ZONE	Yes	(null)	6	(null)

Tabel 3.2.7 DDL Tabel Payment_Details

Nama Tabel	Payment_Details
Deskripsi	Membuat tabel "Payment_Details" yang bertujuan untuk menampung data pembayaran detail pesanan

Script SQL

```
CREATE TABLE payment_details (
    id_paymentdetails              INTEGER
GENERATED BY DEFAULT AS IDENTITY NOT NULL PRIMARY
KEY,
    status                         VARCHAR2(10) NOT
NULL,
    id_orderdetails                INTEGER NOT
NULL,
    created_at                     TIMESTAMP WITH
LOCAL TIME ZONE
);

CREATE UNIQUE INDEX payment_details__idx ON
    payment_details (
        id_orderdetails
    ASC );

ALTER TABLE payment_details
    ADD CONSTRAINT
payment_details_order_details_fk FOREIGN KEY (
id_orderdetails )
        REFERENCES order_details ( id_orderdetails
)
        ON DELETE CASCADE;
```

Screenshot Hasil

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	ID_PAYMENTDETAILS	NUMBER(38,0)	No	"DAFFARAIHANDIKA4"."ISEQ$$_75162".nextval	1	(null)
2	STATUS	VARCHAR2(10 BYTE)	No	(null)	2	(null)
3	ID_ORDERDETAILS	NUMBER(38,0)	No	(null)	3	(null)
4	CREATED_AT	TIMESTAMP(6) WITH LOCAL TIME ZONE	Yes	(null)	4	(null)

3.2.3 DML

Tabel 3.2.8 DML Fitur Registrasi Akun

Nama Fitur	Registrasi Akun

Deskripsi	User melakukan registrasi untuk membuat akun admin atau customer

Script SQL

```
INSERT INTO "User" (username, password, full_name, no_telp, email, role)
    VALUES('alfhandre', 'andreaja', 'Andre Lutfiansyah', '087700720823', 'andre@gmail.com', 'Customer');

INSERT INTO "User" (username, password, full_name, no_telp, email, role)
    VALUES('admin', 'admin321', 'Pick Me Store', '087700727987', 'pickmestore@gmail.com', 'Admin');
```

Screenshot Hasil

	ID_USER	USERNAME	PASSWORD	FULL_NAME	NO_TELP	EMAIL	ROLE	SALDO
1	4	rakamhrdkaa	rakamhrdka321	Maul	087709842323	rakamhrdka@gmail.com	Customer	0
2	5	ferdyfauzan	ferdi123	Ferdy Fauzan	087709847423	ferdif@gmail.com	Customer	0
3	6	daffaraihandika	daffa123	Daffa Raihandika	087700727562	draihandika@gmail.com	Customer	1000000
4	7	alfhandre	andreaja	Andre Lutfiansyah	087700720823	andre@gmail.com	Customer	0
5	8	admin	admin321	Pick Me Store	087700727987	pickmestore@gmail.com	Admin	0

Tabel 3.2.9 DML Fitur Menambah Alamat

Nama Fitur	Menambah Alamat
Deskripsi	User menambahkan alamat untuk tujuan pengiriman

Script SQL

```
INSERT INTO Address (alamat_lengkap, provinsi, kota, kecamatan, kode_pos, id_user)
    VALUES('Jl Soma', 'Jawa Barat', 'Bandung', 'Antapani','402405', 7);
```

Screenshot Hasil

	ID_ADDRESS	ALAMAT_LENGKAP	PROVINSI	KOTA	KECAMATAN	KODE_POS	ID_USER
1	1	Komplek Serra Valley DeLima	Jawa Barat	Bandung	Ciwaruga	40559	4
2	2	Jl Soma	Jawa Barat	Bandung	Antapani	402405	7

Tabel 3.2.10 DML Fitur Menambah Kategori

Nama Fitur	Menambah Kategori
Deskripsi	Admin menambahkan kategori produk

Script SQL

```
INSERT INTO category(nama_kategori)
     VALUES('T-Shirt');
```

Screenshot Hasil

	ID_KATEGORI	NAMA_KATEGORI
1	1	T-Shirt

Tabel 3.2.11 DML Fitur Menambah Produk

Nama Fitur	Menambah Produk
Deskripsi	Admin menambahkan produk
Script SQL	
`INSERT INTO Product (nama_barang, deskripsi, harga, ukuran ,id_kategori, stok)` `VALUES('Black T-Shirt Oversize', 'Bahan Double-faced menggabungkan tampilan clean dan tekstur lembut, Desain kerah yang stylish, Lengan half, drop shoulders, dan siluet wide fit.', 200000, 'L', 1, 11);`	
Screenshot Hasil	

	ID_PRODUCT	NAMA_BARANG	DESKRIPSI	HARGA	UKURAN	ID_KATEGORI	STOK
1	1	Black Regular Fit Hoodie	Sweter bertudung lengan panjang berbahan kain sweter yang lembut dengan ...	300000	M	3	12
2	2	Black T-Shirt Oversize	Bahan Double-faced menggabungkan tampilan clean dan tekstur lembut, Desa...	200000	L	1	11

Tabel 3.2.12 DML Fitur Update Stok Produk

Nama Fitur	Update Stok Produk
Deskripsi	Admin meng-update stok produk tertentu
Script SQL	

`UPDATE product`
`SET stok = 15`
`WHERE id_product = 2;`	

Screenshot Hasil	

Sebelum update

	ID_PRODUCT	NAMA_BARANG	DESKRIPSI	HARGA	UKUR
1	1	Black Regular Fit Hoodie	Sweter bertudung lengan panjang berbahan kain sweter yang lembut dengan ...	300000	M
2	2	Black T-Shirt Oversize	Bahan Double-faced menggabungkan tampilan clean dan tekstur lembut, Desa...	200000	L

Setelah update

	ID_PRODUCT	NAMA_BARANG	DESKRIPSI	HARGA	UKUR
1	1	Black Regular Fit Hoodie	Sweter bertudung lengan panjang berbahan kain sweter yang lembut dengan ...	300000	M
2	2	Black T-Shirt Oversize	Bahan Double-faced menggabungkan tampilan clean dan tekstur lembut, Desa...	200000	L

Tabel 3.2.13 DML Update Password, Email, dan No. Hp User

Nama Fitur	Update Password, Email, dan No Hp User
Deskripsi	User meng-update password, emailn dan no hp
Script SQL	

`UPDATE "User"`
`SET saldo = 250000`
`WHERE username = 'admin'`	

Screenshot Hasil	
Sebelum update	

	ID_USER	USERNAME	PASSWORD	FULL_NAME	NO_TELP	EMAIL
1	4	rakamhrdkaa	rakamhrdka321	Maul	087709842323	rakamhrdka@gmail.com
2	5	ferdyfauzan	ferdi123	Ferdy Fauzan	087709847423	ferdif@gmail.com
3	6	daffarai...	daffa123	Daffa Raihandika	087700727562	draihandika@gmail.com
4	7	alfhandre	andreaja	Andre Lutfiansyah	087700720823	andre@gmail.com
5	8	admin	admin321	Pick Me Store	087700727987	pickmestore@gmail.com

Setelah update

	ID_USER	USERNAME	PASSWORD	FULL_NAME	NO_TELP	EMAIL
1	4	rakamhrdkaa	rakamhrdka321	Maul	087709842323	rakamhrdka@gmail.com
2	5	ferdyfauzan	ferdi123	Ferdy Fauzan	087709847423	ferdif@gmail.com
3	6	daffarai...	daffa123	Daffa Raihandika	087700727562	draihandika@gmail.com
4	7	alfhandre	andreaja	Andre Lutfiansyah	087700720823	andre@gmail.com
5	8	admin	admin321	Pick Me Store	087700727987	pickmestore@gmail.com

Tabel 3.2.14 DML Fitur Update Alamat

Nama Fitur	Update Alamat
Deskripsi	User meng-update alamatnya

Script SQL

```
UPDATE address
SET alamat_lengkap = 'Jl Soma no 6'
WHERE id_address = 2;
```

Screenshot Hasil

Sebelum update

	ID_ADDRESS	ALAMAT_LENGKAP	PROVINSI	KOTA	KECAMATAN	KODE_POS	ID_USER
1	1	Komplek Serra Valley DeLima	Jawa Barat	Bandung	Ciwaruga	40559	4
2	2	Jl Soma	Jawa Barat	Bandung	Antapani	402405	7

Setelah update

	ID_ADDRESS	ALAMAT_LENGKAP	PROVINSI	KOTA	KECAMATAN	KODE_POS	ID_USER
1	1	Komplek Serra Valley DeLima	Jawa Barat	Bandung	Ciwaruga	40559	4
2	2	Jl Soma no 6	Jawa Barat	Bandung	Antapani	402405	7

Tabel 3.2.15 DML Fitur Menghapus Akun

Nama Fitur	Menghapus Akun
Deskripsi	User menghapus akunnya

Script SQL

```
DELETE FROM "User" WHERE username = 'rakamhrdkaa';
```

Screenshot Hasil

Sebelum update

	ID_USER	USERNAME	PASSWORD	FULL_NAME	NO_TELP	EMAIL
1	4	rakamhrdkaa	rakamhrdka321	Maul	087709842323	rakamhrdka@gmail.com
2	5	ferdyfauzan	ferdi123	Ferdy Fauzan	087709847423	ferdif@gmail.com
3	6	daffarai...	daffa123	Daffa Raihandika	087700727562	draihandika@gmail.com
4	7	alfhandre	andreaja	Andre Lutfiansyah	087700720823	andre@gmail.com
5	8	admin	admin321	Pick Me Store	087700727987	pickmestore@gmail.com

Setelah update

	ID_USER	USERNAME	PASSWORD	FULL_NAME	NO_TELP	EMAIL	ROLE	SALDO
1	5	ferdyfauzan	ferdi123	Ferdy Fauzan	087709847423	ferdif@gmail.com	Customer	1000000
2	6	daffarai...	daffa123	Daffa Raihandika	087700727562	draihandika@gmail.com	Customer	1000000
3	7	alfhandre	andreaja	Andre Lutfiansyah	087700720823	andre@gmail.com	Customer	800000
4	8	admin	admin321	Pick Me Store	087700727987	pickmestore@gmail.com	Admin	250000

Tabel 3.2.16 DML Fitur Menghapus Alamat

Nama Fitur	Menghapus Alamat
Deskripsi	User menghapus alamat
Script SQL	
`DELETE FROM Address WHERE id_address = 2;`	
Screenshot Hasil	

Sebelum update

	ID_ADDRESS	ALAMAT_LENGKAP	PROVINSI	KOTA	KECAMATAN	KODE_POS	ID_USER
1	2	Jl Soma no 6	Jawa Barat	Bandung	Antapani	402405	7

Setelah update

ID_ADDR...	ALAMAT_...	PROVINSI	KOTA	KECAMATAN	KODE_POS	ID_USER

3.3 [Normalisasi]

PICK ME STORE

Payment Invoice

Customer Id	:	3	**Payment Id**	:	7
Customer Username	:	alfhandre	**Date**	:	05 December 2022
Customer Name	:	Andre Lutfiansyah	**Status**	:	**LUNAS**
Customer Phone	:	085156534830			
Customer Email	:	andrelutfiansyah24@gmail.com			
Customer Address	:	Jl. Soma RT/RW 005/015, Babakan Surabaya, Kiaracondong, Bandung, Jawa Barat, 40283			

No	Product Id	Product Name	Category	Size	Quantity	Price	Subtotal
1	4	Black T-Shirt Oversize	T-Shirt	M	2	Rp 200.000	Rp 400.000
2	5	Black Regular Fit Hoodie	Hoodie	L	1	Rp 300.000	Rp 300.000
3	6	Baggy Blue Jeans	Celana Panjang	XL	2	Rp 450.000	Rp 900.000
			Total Price			Rp	1.600.000

Signature

Mochamad Ferdy Fauzan

Admin/Seller

Gambar 3.3.1 Contoh Payment Invoice dari Sistem Aplikasi PickMeStore

3.3.1 UNF

Tabel 3.3.1 Normalisasi dalam Bentuk UNF dari Payment Invoice

Payment Id	Date	Status	Customer Id	Customer Username	Customer Name	Customer Phone	Customer Email	Customer Address
7	05 December 2022	LUNAS	3	alfhandre	Andre Lutfiansyah	085156534830	andrelutfiansyah24@gmail.com	Jl. Soma RT/RW 005/015 Babakan Surabaya, Kiaracondong, Bandung, Jawa Barat, 40283

No	Product Id	Product Name	Category	Size	Quantity	Price	Subtotal	Total Price	Admin Name
1	4	Black T-Shirt	T-Shirt	M	2	Rp 200.000	Rp 400.000	Rp 1.600.000	Mochamad Ferdy Fauzan
2	5	Black Regular	Hoodie	L	1	Rp 300.000	Rp 300.000		
3	6	Baggy Blue Jeans	Celana Panjang	XL	2	Rp 450.000	Rp 900.000		

3.3.2 1NF

Tabel 3.3.2 Normalisasi dalam Bentuk 1NF dari Payment Invoice

Payment Id	Date	Status	Customer Id	Customer Username	Customer Name	Customer Phone	Customer Email	Customer Address	Kecamatan	Kota	Provinsi	Kode Pos
7	05 December 2022	LUNAS	3	alfhandre	Andre Lutfiansyah	085156534830	andrelutfiansyah24@gmail.com	Jl. Soma RT/RW 005/015 Babakan Surabaya	Kiaracondong	Bandung	Jawa Barat	40283
7	05 December 2022	LUNAS	3	alfhandre	Andre Lutfiansyah	085156534830	andrelutfiansyah24@gmail.com	Jl. Soma RT/RW 005/015 Babakan Surabaya	Kiaracondong	Bandung	Jawa Barat	40283
7	05 December 2022	LUNAS	3	alfhandre	Andre Lutfiansyah	085156534830	andrelutfiansyah24@gmail.com	Jl. Soma RT/RW 005/015 Babakan Surabaya	Kiaracondong	Bandung	Jawa Barat	40283

Product Id	Product Name	Category	Size	Quantity	Price	Subtotal	Total Price	Admin Name
4	Black T-Shirt Oversize	T-Shirt	M	2	Rp 200.000	Rp 400.000	Rp 1.600.000	Mochamad Ferdy Fauzan
5	Black Regular Fit Hoodie	Hoodie	L	1	Rp 300.000	Rp 300.000	Rp 1.600.000	Mochamad Ferdy Fauzan
6	Baggy Blue Jeans	Celana Panjang	XL	2	Rp 450.000	Rp 900.000	Rp 1.600.000	Mochamad Ferdy Fauzan

3.3.3 2NF

Tabel 3.3.3 Tabel Customer, Normaliasi dalam Bentuk 2NF dari Payment Invoice

Customer					
Customer Id	**Customer Username**	**Customer Name**	**Customer Phone**	**Customer Email**	**Admin Name**
3	alfhandre	Andre Lutfiansyah	085156534830	andrelutfiansyah24@gmail.com	Mochamad Ferdy Fauzan

Tabel 3.3.4 Tabel Address, Normaliasi dalam Bentuk 2NF dari Payment Invoice

Address					
Customer Id	**Customer Address**	**Kecamatan**	**Kota**	**Provinsi**	**Kode Pos**
3	Jl. Soma RT/RW 005/015 Babakan Surabaya	Kiaracondong	Bandung	Jawa Barat	40283

Tabel 3.3.5 Tabel Payment, Normaliasi dalam Bentuk 2NF dari Payment Invoice

Payment							
Product Id	**Quantity**	**Subtotal**	**Total Price**	**Customer Id**	**Payment Id**	**Date**	**Status**
4	2	Rp 400.000	Rp 1.600.000	3	7	05 December 2022	LUNAS
5	1	Rp 300.000	Rp 1.600.000	3	7	05 December 2022	LUNAS
6	2	Rp 900.000	Rp 1.600.000	3	7	05 December 2022	LUNAS

Tabel 3.3.6 Tabel Product, Normaliasi dalam Bentuk 2NF dari Payment Invoice

Product				
Product Id	**Product Name**	**Category**	**Size**	**Price**
4	Black T-Shirt Oversize	T-Shirt	M	Rp 200.000
5	Black Regular Fit Hoodie	Hoodie	L	Rp 300.000
6	Baggy Blue Jeans	Celana Panjang	XL	Rp 450.000

3.3.4 3NF

Address

Address Id	User Id	Customer Address	Kecamatan	Kota	Provinsi	Kode Pos
	3	Jl. Soma RT/RW 005/015 Babakan Surabaya	Kiaracondong	Bandung	Jawa Barat	40283

User

User Id	Username	Name	Phone	Email	Role
3	alfhandre	Andre Lutfiansyah	085156534830	andrelutfiansyah24@gmail.com	Customer
		Mochamad Ferdy Fauzan			Admin

Order Item

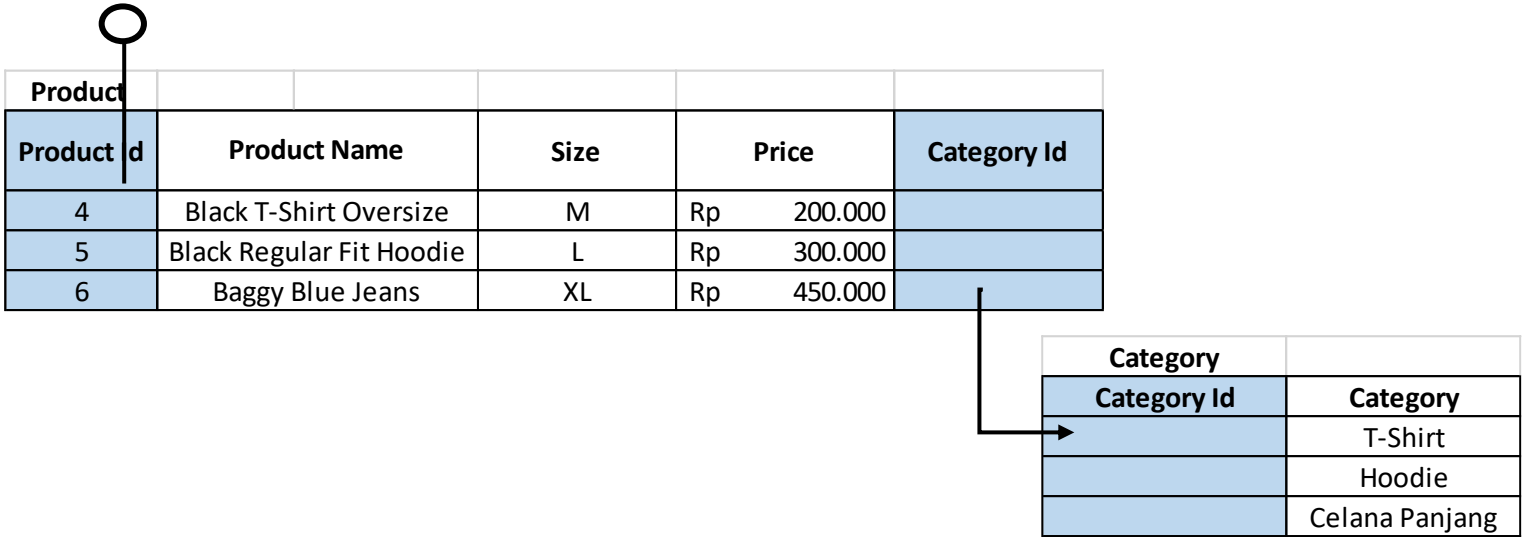
Order Item Id	Product Id	Quantity	Subtotal	Order Details Id
	4	2	Rp 400.000	
	5	1	Rp 300.000	
	6	2	Rp 900.000	

Order Details

Order Details Id	Total Price	User Id
	Rp 1.600.000	3

Payment Details

Payment Id	Date	Status	Order Details Id
7	05 December 2022	LUNAS	



Gambar 3.3.2 Normalisasi dalam Bentuk 3NF dari Payment Invoice yang menghasilkan beberapa tabel baru masing – masing memiliki id sebagai primary key

3.4 Mapping Business Process ke PLSQL

3.4.1 Procedure

Tabel 3.4.1 Procedure Register

Nama Procedure	Register
Deskripsi	Membuat registrasi akun customer
Parameter	Input : username_in, password_in
	Output : -
Script SQL	

```
create or replace PROCEDURE Register (username_in
IN VARCHAR2, password_in IN VARCHAR2,
retype_password IN VARCHAR2, full_name_in IN
VARCHAR2, no_telp_in  VARCHAR2, email_in IN
VARCHAR2)
IS
BEGIN
        IF (password_in = retype_password AND
LENGTH(password_in) >= 8) THEN
            INSERT INTO "User"(username, password,
full_name, no_telp,email, role)
                VALUES(username_in, password_in,
full_name_in, no_telp_in,email_in, 'Customer');

            dbms_output.put_line ('Register
Success!!');
        ELSIF (password_in != retype_password)
THEN
            dbms_output.put_line ('Register
Failed');
            dbms_output.put_line ('Password Not
Match!!');
        END IF;

        IF (LENGTH(password_in) < 8) THEN
            dbms_output.put_line ('Register
Failed');
            dbms_output.put_line ('Password
minimal 8 karakter!!');
        END IF;
        COMMIT;
END;
```

Tabel 3.4.2 Procedure AddProduct

Nama Procedure	AddProduct
Deskripsi	Menambahkan product
Parameter	Input :name_barang_in, deskripsi_in, id_kategori_in, stok_in

	Output : -
Script SQL	

```
create or replace PROCEDURE AddProduct
(nama_barang_in VARCHAR2, deskripsi_in CLOB,
harga_in INTEGER, ukuran_in VARCHAR2,
id_kategori_in INTEGER, stok_in INTEGER)
IS
BEGIN
    INSERT INTO Product (nama_barang, deskripsi,
harga, ukuran, id_kategori, stok)
        VALUES (nama_barang_in, deskripsi_in,
harga_in, ukuran_in, id_kategori_in, stok_in);
    dbms_output.put_line ('Produk ' ||
nama_barang_in || ' berhasil ditambahkan');

    COMMIT;
END;
```

Tabel 3.4.3 Procedure ShowProductDetails

Nama Procedure	showProductDetails
Deskripsi	Menampilkan detail produk yang dicari
Parameter	Input : name_barang_in
	Output : -
Script SQL	

```
create or replace PROCEDURE showProductDetails (
    nama_barang_in IN VARCHAR2)
AS
    c1 SYS_REFCURSOR;
BEGIN
    OPEN c1 FOR
    SELECT * FROM Product
    WHERE nama_barang = nama_barang_in;
    DBMS_SQL.RETURN_RESULT(c1);

END;
```

Tabel 3.4.4 Procedure ChangePassword

Nama Procedure	changePassword
Deskripsi	Mengganti password customer
Parameter	Input : username_in, oldPassword_in, newPassword_in
	Output :
Script SQL	

```
create or replace PROCEDURE changePassword (
    username_in IN VARCHAR2, oldPassword IN
VARCHAR2, newPassword IN VARCHAR2)
AS
BEGIN
    FOR i IN (SELECT * FROM "User" where password
= oldPassword AND username = username_in) LOOP
        IF i.password = oldPassword THEN
            UPDATE "User"
            SET password = newPassword
            WHERE password = oldPassword;
            dbms_output.put_line ('Password
berhasil diubah!!');
            EXIT WHEN i.password = newPassword; --
login success
        END IF;
    END LOOP;
END;
```

Tabel 3.4.5 Procedure AddAddress

Nama Procedure	addAddress
Deskripsi	Menambahkan alamat customer
Parameter	Input : username_in, password_in, alamat_lengkap, provinsi_in, kota_in, kecamatan_in, kode_pos_in
	Ouput :
Script SQL	

```
create or replace PROCEDURE addAddress(
    username_in IN  VARCHAR2,
    password_in IN VARCHAR2,
    alamat_lengkap_in IN VARCHAR2,
    provinsi_in IN VARCHAR2,
    kota_in IN VARCHAR2,
    kecamatan_in IN VARCHAR2,
    kode_pos_in IN VARCHAR2)
AS
    user_id INTEGER;
BEGIN
    user_id := GetUserId(username_in,
password_in);
    INSERT INTO Address (alamat_lengkap, provinsi,
kota, kecamatan, kode_pos, id_user)
         VALUES (alamat_lengkap_in, provinsi_in,
kota_in, kecamatan_in, kode_pos_in, user_id);
END;
```

Tabel 3.4.6 Procedure CreateOrderDetails

Nama Procedure	CreateOrderDetails
Deskripsi	Membuat detail pesanan customer
Parameter	Input : username_in, password_in
	Output: -
Script SQL	

```
create or replace PROCEDURE CreateOrderDetails
(username_in IN VARCHAR2, password_in IN VARCHAR2)
IS
    user_id NUMBER;
BEGIN
    user_id := GetUserId(username_in,
password_in);
    INSERT INTO order_details(id_user,harga_total,
created_at)
        VALUES(user_id,0, current_date);

    COMMIT;
END;
```

Tabel 3.4.7 Procedure Order_Per_Produk

Nama Procedure	order_per_produk
Deskripsi	Membuat pesanan per produk
Parameter	Input : product_name, size_in, quantity_in, username_in, password_in
	Output : -
Script SQL	

```
create or replace PROCEDURE order_per_produk
(product_name_in IN VARCHAR2, size_in IN
VARCHAR2,quantity_in NUMBER, username_in IN
VARCHAR2, password_in IN VARCHAR2)
IS
    product_id NUMBER;
    price NUMBER;
    order_details_id NUMBER;
BEGIN
    product_id :=
GetProductId(product_name_in,size_in);
    SELECT harga INTO price
    FROM product WHERE id_product = product_id;
    price := price * quantity_in;

    order_details_id :=
GetOrderDetailsId(username_in, password_in);
```

```
    INSERT INTO order_items (id_orderdetails,
id_product, quantity, total_perproduk)
        VALUES (order_details_id, product_id,
quantity_in, price);

    COMMIT;
END;
```

Tabel 3.4.8 Procedure SumOrderProduct

Nama Procedure	SumOrderProduct
Deskripsi	Melakukan akumulasi dari order_items ke order_details
Parameter	Input : username_in, password_in
	Output :
Script SQL	

```
create or replace PROCEDURE SumOrderProduct
(username_in IN VARCHAR2, password_in IN VARCHAR2)
IS
    order_details_id NUMBER;
    total_harga NUMBER;
BEGIN
    order_details_id :=
GetOrderDetailsId(username_in, password_in);
    SELECT SUM(total_perproduk) INTO total_harga
    FROM order_items
    WHERE id_orderdetails = order_details_id;

    UPDATE order_details
    SET harga_total = total_harga, modified_at =
CURRENT_DATE
    WHERE id_orderdetails = order_details_id;

    COMMIT;
END;
```

Tabel 3.4.9 Procedure TopUpSaldo

Nama Procedure	topUpSaldo
Deskripsi	Customer melakukan top up saldo
Parameter	Input : username_in, password_in, saldo_in
	Output : -
Script SQL	

```
create or replace PROCEDURE topUpSaldo
(username_in IN VARCHAR2, password_in IN VARCHAR2,
saldo_in IN NUMBER)
IS
```

```
    user_id NUMBER;
    saldo_awal NUMBER;
BEGIN
    user_id := GetUserId(username_in,
password_in);
    SELECT saldo INTO saldo_awal
    FROM "User" WHERE id_user = user_id;
    saldo_awal := saldo_awal + saldo_in;

    UPDATE "User"
    SET saldo = saldo_awal
    WHERE id_user = user_id;

    COMMIT;
END;
```

Tabel 3.4.10 Procedure AddStokProduct

Nama Procedure	addStokProduct
Deskripsi	Menambahkan stok produk
Parameter	Input : stok_in, nama_barang_in, ukuran_in
	Output : -
Script SQL	

```
create or replace PROCEDURE addStokProduct
(stok_in IN NUMBER, nama_barang_in IN VARCHAR2,
ukuran_in IN VARCHAR2)
IS
    product_id NUMBER;
    stok_awal NUMBER;
BEGIN
    product_id := GetProductId(nama_barang_in,
ukuran_in);
    SELECT stok INTO stok_awal
    FROM product WHERE id_product = product_id;
    stok_awal := stok_awal + stok_in;

    UPDATE product
    SET stok = stok_awal
    WHERE id_product = product_id;
    dbms_output.put_line ('stok berhasil
ditambahkan');

    COMMIT;
END;
```

Tabel 3.4.11 Procedure PaymentStatus

Nama Procedure	paymentStatus

Deskripsi	Mengubah status pembayaran
Parameter	Input : username_in, password_in
	Output L -
Script SQL	

```
create or replace PROCEDURE paymentStatus
(username_in IN VARCHAR2, password_in IN VARCHAR2)
IS
    order_details_id NUMBER;
BEGIN
    order_details_id :=
GetOrderDetailsId(username_in, password_in);

    INSERT INTO payment_details(status,
id_orderdetails, created_at)
        VALUES('Belum Lunas', order_details_id,
current_date);
    COMMIT;
END;
```

Tabel 3.4.12 Procedure PayOrder

Nama Procedure	payOrder
Deskripsi	Membayar pesanan customer
Parameter	Input : username_in, password_in
	Output :
Script SQL	

```
create or replace PROCEDURE payOrder (username_in
IN VARCHAR2, password_in IN VARCHAR2)
IS
    order_details_id NUMBER;
    user_id NUMBER;
    total_tagihan NUMBER;
    saldo_user NUMBER;
    current_saldo NUMBER;
BEGIN
    order_details_id :=
GetOrderDetailsId(username_in, password_in);
    user_id := GetUserId(username_in,
password_in);

    SELECT harga_total INTO total_tagihan
    FROM order_details WHERE id_orderdetails =
order_details_id;

    SELECT saldo INTO saldo_user
    FROM "User" WHERE id_user = user_id;

    current_saldo := saldo_user - total_tagihan;
    IF (saldo_user < total_tagihan) THEN
```

```
        dbms_output.put_line ('Saldo anda
kurang!!');
        ELSIF (saldo_user = current_saldo +
total_tagihan) THEN

            UPDATE payment_details
            SET status = 'Lunas'
            WHERE id_orderdetails =
order_details_id;

            UPDATE "User"
            SET saldo = current_saldo
            WHERE id_user = user_id;

            UPDATE "User"
            SET saldo = saldo + total_tagihan
            WHERE role = 'Admin';

    END IF;

    COMMIT;
END;
```

Tabel 3.4.13 Procedure CancelOrder

Nama Procedure	cancelOrder
Deskripsi	Membatalkan pesanan yang dipesan customer
Parameter	Input : username_in, password_in
	Output :
Script SQL	

```
create or replace PROCEDURE cancelOrder
(username_in IN VARCHAR2, password_in IN VARCHAR2)
IS
    order_details_id NUMBER;
BEGIN
    order_details_id :=
GetOrderDetailsId(username_in, password_in);

    UPDATE order_details
    SET status = 'Canceled',
        harga_total = 0
    WHERE id_orderdetails = order_details_id;

    COMMIT;
END;
```

Tabel 3.4.14 Procedure CreateDetailPesanan

Nama Procedure	CreateDetailPesanan
Deskripsi	Untuk memberi informasi mengenai details order customer
Parameter	Input : username_in, password_in
	Output :
Script SQL	

```
create or replace PROCEDURE createDetailPesanan
(username_in IN VARCHAR2, password_in IN VARCHAR2)
AS
    order_details_id NUMBER;
    product_id NUMBER;
    quantity_in NUMBER;
    subtotal NUMBER;
    status_in VARCHAR2(20);
    created_at_in TIMESTAMP WITH LOCAL TIME ZONE;
    total_price NUMBER;
BEGIN
    order_details_id :=
GetOrderDetailsId(username_in, password_in);
    SELECT id_product, quantity, total_perproduk
INTO product_id, quantity_in, subtotal FROM
order_items
    WHERE id_orderdetails = order_details_id;

    SELECT status, created_at INTO status_in,
created_at_in FROM payment_details
    WHERE id_orderdetails = order_details_id;

    SELECT harga_total INTO total_price FROM
order_details
    WHERE id_orderdetails = order_details_id;

    dbms_output.put_line ('Details Order');
    dbms_output.put_line ('-------');
    dbms_output.put_line ('');
    dbms_output.put_line ('Product Id : ' ||
product_id);
    dbms_output.put_line ('Quantity : ' ||
quantity_in);
    dbms_output.put_line ('Subtotal : ' ||
subtotal);
    dbms_output.put_line ('Total Price : ' ||
total_price);
    dbms_output.put_line ('Status : Lunas' );
    dbms_output.put_line ('Date : ' ||
created_at_in);
END;
```

Tabel 3.4.15 Procedure CreateInvoice

Nama Procedure	CreateInvoice
Deskripsi	Untuk memberi informasi mengenai payment invoice
Parameter	Input : username_in, password_in
	Output :
Script SQL	

```
create or replace PROCEDURE createInvoice
(username_in IN VARCHAR2, password_in IN VARCHAR2)
AS
    product_id NUMBER;
    user_id NUMBER;
    full_name_in VARCHAR2(100);
    no_telp_in VARCHAR2(50);
    email_in VARCHAR2(100);
    alamat_lengkap_in VARCHAR2(100);
    provinsi_in VARCHAR2(20);
    kota_in VARCHAR2(20);
    kecamatan_in VARCHAR(25);
    kode_pos_in VARCHAR(10);
BEGIN
    user_id := GetUserId(username_in,
password_in);
    SELECT full_name, no_telp, email INTO
full_name_in, no_telp_in, email_in FROM "User"
    WHERE id_user = user_id;

    SELECT alamat_lengkap, provinsi, kota,
kecamatan, kode_pos INTO alamat_lengkap_in,
provinsi_in, kota_in, kecamatan_in, kode_pos_in
    FROM address
    WHERE id_user = user_id AND rownum = 1;


    dbms_output.put_line ('INVOICE');
    dbms_output.put_line ('-------');
    dbms_output.put_line ('');
    dbms_output.put_line ('Customer Id : ' ||
user_id);
    dbms_output.put_line ('Customer Username : '
|| username_in);
    dbms_output.put_line ('Customer Name : ' ||
full_name_in);
    dbms_output.put_line ('Customer Phone : ' ||
no_telp_in);
    dbms_output.put_line ('Customer Email : ' ||
email_in);
```

```
    dbms_output.put_line ('Customer Address : ' ||
alamat_lengkap_in || ',' || provinsi_in || ',' ||
kota_in || ',' || kecamatan_in || ',' ||
kode_pos_in);
    dbms_output.put_line ('');

    createDetailPesanan(username_in, password_in);

END;
```

Tabel 3.4.16 Procedure DeleteProduct

Nama Procedure	DeleteProduct
Deskripsi	Prosedur ini bertujuan untuk menghapus produk yang ada pada tabel produk
Parameter	Input : nama_barang_in, ukuran_in
	Output :
Script SQL	

```
create or replace PROCEDURE deleteProduct
(nama_barang_in VARCHAR2, ukuran_in VARCHAR2)
IS
    product_id NUMBER;
BEGIN
    product_id := GetProductId(nama_barang_in,
ukuran_in);

    DELETE FROM product
    WHERE id_product = product_id;
    dbms_output.put_line ('Produk ' ||
nama_barang_in || ' berhasil dihapus');

    COMMIT;
END;
```

Tabel 3.4.17 Procedure SearchProductFromCategory

Nama Procedure	SearchProductFromCategory
Deskripsi	Prosedur ini bertujuan untuk mencari produk berdasarkan suatu kategori
Parameter	Input : kategori_in
	Output :
Script SQL	

```
create or replace PROCEDURE
searchProductFromCategory (kategori_in IN
VARCHAR2)
AS
```

```
    category_id NUMBER;
    nama_barang_in VARCHAR2(50);
    c1 SYS_REFCURSOR;
BEGIN
    category_id := GetCategoryId(kategori_in);
    OPEN c1 FOR
    SELECT nama_barang, deskripsi, harga, ukuran,
stok
    FROM product
    WHERE id_kategori = category_id;
    DBMS_SQL.RETURN_RESULT(c1);
END;
```

3.4.2 Function

Tabel 3.4.18 Function Login

Nama Function	Login
Deskripsi	User melakukan login akun
Parameter	Input : username_in, password_in
	Output : -
Script SQL	

```
create or replace NONEDITIONABLE FUNCTION Login
(username_in IN VARCHAR2, password_in IN VARCHAR2)
RETURN INTEGER
IS
BEGIN
    FOR i IN (SELECT * FROM "User" where username
= username_in) LOOP
         FOR j IN (SELECT * FROM "User" where
password = password_in) LOOP
             IF i.password = j.password AND
i.username = j.username THEN
                return 0; --login success
            END IF;
         END LOOP;
         dbms_output.put_line ('Wrong
Password!!');
        return 1; --invalid password
    END LOOP;
    dbms_output.put_line ('Username not
registered!!');
    return 2; --username not registered

    COMMIT;
END;
```

Tabel 3.4.19 Function GetUserId

Nama Function	GetUserId
Deskripsi	Mendapatkan id user yang sedang login
Parameter	Input : username_in, password_in
	Output : -
Script SQL	

```
create or replace FUNCTION GetUserId (
    username_in IN VARCHAR2,
    password_in IN VARCHAR2)
RETURN NUMBER
IS
    isLogin NUMBER;
    user_id NUMBER;
BEGIN
    isLogin := Login(username_in, password_in);
    IF isLogin = 0 THEN
        SELECT id_user INTO user_id FROM "User"
        WHERE username = username_in;
    END IF;
RETURN user_id;
END;
```

Tabel 3.4.20 Function GetProductId

Nama Function	GetProductId
Deskripsi	Mendapatkan id dari produk
Parameter	Input : product_name_in, size_in
	Output : -
Script SQL	

```
create or replace FUNCTION GetProductId (
    product_name_in IN VARCHAR2,
    size_in IN VARCHAR2)
RETURN NUMBER
IS
    product_id NUMBER;
BEGIN
    SELECT id_product INTO product_id
         FROM Product WHERE nama_barang =
product_name_in
         AND ukuran = size_in;
RETURN product_id;
END;
```

Tabel 3.4.21 Function GetOrderDetailsId

Nama Function	GetOrderDetailsId

Deskripsi	Mendapatkan id dari order details
Parameter	Input : username_in, password_in
	Output : -

Script SQL

```
create or replace FUNCTION GetOrderDetailsId (
    username_in IN VARCHAR2,
    password_in IN VARCHAR2)
RETURN NUMBER
IS
    isUser NUMBER;
    order_detail_id NUMBER;
BEGIN
    isUser := GetUserId(username_in, password_in);
    SELECT id_orderdetails INTO order_detail_id
    FROM order_details WHERE id_user = isUser;
RETURN order_detail_id;
END;
```

3.4.3 Trigger

Tabel 3.4.22 Trigger trg_update_stok

Nama Trigger	trg_update_stok
Deskripsi	Trigger untuk menjaga stok ketika insert, update, dan delete stok di order_items. Dengan adanya trigger ini, nilai stok yang ada pada tabel Produk akan terjaga konsistensinya

Script SQL

```
create or replace TRIGGER trg_update_stok
AFTER INSERT OR DELETE OR UPDATE ON order_items
FOR EACH ROW
DECLARE
    tmp_stok NUMERIC;
    tmp_status VARCHAR2(15);
BEGIN
    IF INSERTING THEN
        SELECT stok INTO tmp_stok FROM Product
WHERE id_product = :new.id_product; -- menampung
nilai stok dari tabel produk pada variabel
tmp_stok
        IF (:new.quantity <= tmp_stok) THEN --
kondisi ketika stok produk >= jumlah pesanan
            UPDATE Product SET stok = (tmp_stok -
:new.quantity) WHERE id_product = :new.id_product;
-- mengurangi nilai stok pada tabel produk
sebanyak jumlah pesanan
        ELSE
```

```
            RAISE_APPLICATION_ERROR(-20000, 'Stok
tidak mencukupi'); -- exeception handling ketika
memesan melebihi jumlah stok
        END IF;
    ELSIF UPDATING THEN
        SELECT stok INTO tmp_stok FROM Product
WHERE id_product = :new.id_product; -- menampung
nilai stok dari tabel produk pada variabel
tmp_stok
        IF (:new.quantity = 0) THEN
            UPDATE product SET stok = stok +
:new.quantity;
        ELSIF (:new.quantity <= tmp_stok +
:old.quantity) THEN -- kondisi ketika jumlah <=
stok produk + jumlah pesanan yg lama
            UPDATE product SET stok = ((tmp_stok +
:old.quantity) - :new.quantity) WHERE id_product =
:old.id_product; -- mengupdate nilai stok pada
tabel produk dengan (nilai stok + jumlah pesanan
lama) - jumlah pesanan terbaru
        ELSE
            RAISE_APPLICATION_ERROR(-20000, 'Stok
tidak mencukupi'); -- exeception handling ketika
melakukan update pesanan melebihi jumlah stok
        END IF;
    ELSIF DELETING THEN -- ketika menghapus
pesanan
        UPDATE product SET stok = (stok +
:old.quantity) WHERE id_product = :old.id_product;
-- mengupdate nilai stok pada tabel produk dengan
menambahkan kembali jumlah pesanan
    END IF;
END;
```

Tabel 3.4.23 Trigger cancelled_order

Nama Trigger	cancelled_order
Deskripsi	Trigger ini bertujuan untuk kondisi ketika status order = ‘Canceled’ atau ‘Expired’
Script SQL	

```
create or replace NONEDITIONABLE TRIGGER
trg_cancelled_order
AFTER UPDATE ON Order_Details
FOR EACH ROW
BEGIN
     IF :new.status = 'Canceled' OR :new.status =
'Expired' THEN
         DELETE FROM order_items WHERE
id_orderdetails = :new.id_orderdetails;
```

```
          DELETE FROM payment_details WHERE
id_orderdetails = :new.id_orderdetails;
     END IF;
END;
```

Tabel 3.4.24 Trigger trg_transaksi_lunas

Nama Trigger	trg_transaksi_lunas
Deskripsi	Trigger ini bertujuan untuk kondisi ketika status pembayaran = ‘Lunas’
Script SQL	

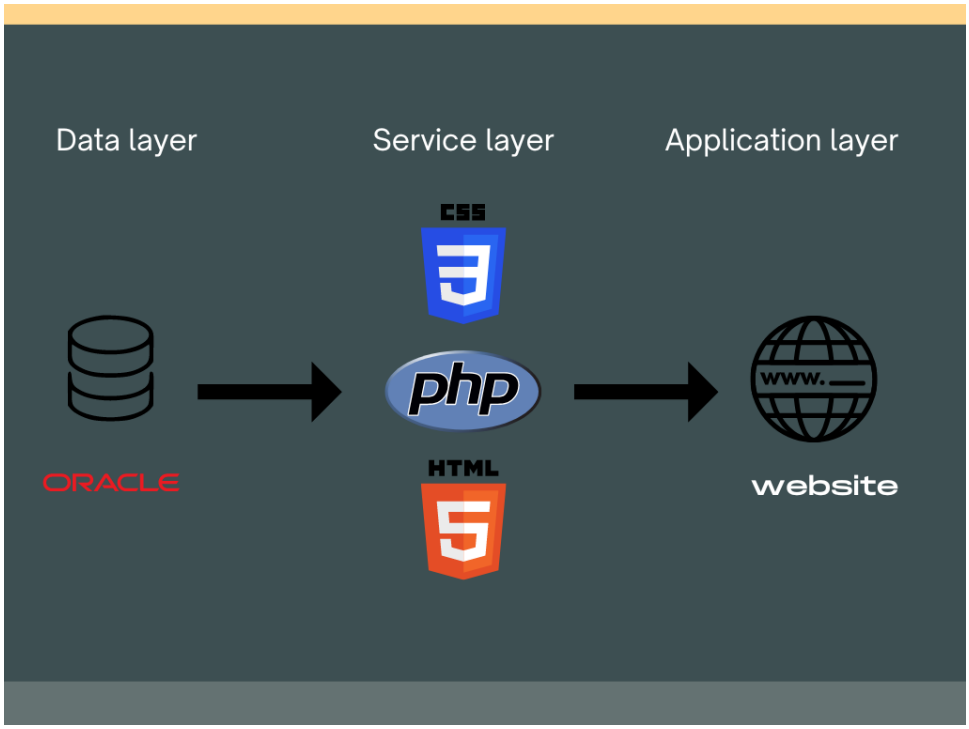
```
create or replace NONEDITIONABLE TRIGGER
trg_transaksi_lunas
AFTER UPDATE ON payment_details
FOR EACH ROW
BEGIN
     IF :new.status = 'Lunas' THEN
            UPDATE order_items
            SET total_perproduk = 0,
                quantity = 0
            WHERE id_orderdetails =
:new.id_orderdetails;
     END IF;
END;
```

Tabel 3.4.25 Trigger trg_expired_payment

Nama Trigger	trg_expired_payment
Deskripsi	Trigger untuk memeriksa pesanan sudah kadaluarsa
Script SQL	

```
create or replace NONEDITIONABLE TRIGGER
trg_expired_payment
AFTER UPDATE ON payment_details
FOR EACH ROW
BEGIN
    IF extract(day from (sysdate -
:old.created_at)) >= 2 AND :new.status = 'Belum
Lunas' THEN
        UPDATE order_details SET status =
'Expired' WHERE :old.id_orderdetails =
:new.id_orderdetails;
    END IF;
END;
```

3.5 Arsitektur Aplikasi



Gambar 3.5.1 Arsitektur Aplikasi mulai dari Data layer, Service layer hingga Application layer

3.5.1 Database

Database atau yang dikenal juga dengan istilah basis data adalah sekumpulan data yang dikelola dengan sedemikian rupa berdasarkan ketentuan tertentu yang saling berkaitan sehingga memudahkan dalam pengelolaannya. PickMeStore menggunakan database oracle untuk informasi online shop.

Oracle adalah sebuah perangkat lunak sistem manajemen basis data relasional (RDBMS). oracle juga menggunakan SQL atau structure query language sebagai bahasa saat mengoperasikan. Dengan menggunakan oracle **Dapat mengolah data dengan cepat dan akurat.**

3.5.2 Front-End

Front end merupakan salah satu bagian dari website yang menampilkan tampilan untuk para pengguna. Pada bagian ini PickMeStore membuat dengan menggunakan 2 bahasa pemrograman web yaitu HTML, dan CSS .

3.5.3 Back-End

Back-End adalah sistem di balik layar yang mengolah database dan juga server. Ya seperti definisinya back-End tidak bisa dilihat oleh user. Pada bagian ini PickMeStore membuat dengan menggunakan bahasa pemrograman PHP

BAB IV
IMPLEMENTASI

4.1 Teknologi Pengembangan Aplikasi

4.1.1 Data Layer

- Oracle data modeler
  adalah alat grafis yang mengesankan yang meningkatkan produktivitas serta menyederhanakan tugas pemodelan data. Dengan menggunakan aplikasi ini, Anda dapat dengan mudah membuat, menelusuri, dan mengedit model tipe logis, rasional, fisik, dan tipe data.
- SQL developer
  SQL Developer adalah salah satu software untuk pengembangan Database. Dengan menggunakan SQL Developer, kita dapat menelusuri objek database, menjalankan pernyataan SQL dan script SQL, serta mengedit dan debug PL / pernyataan SQL

4.1.2 Service layer

- HTML
  HTML adalah kependekan dari Hypertext Markup Language yang merupakan sebuah bahasa markup. HTML adalah kode untuk membuat struktur halaman
  suatu website yang menarik, saling terhubung satu dengan yang lainnya, dan yang pasti dapat diakses melalui internet. HTML memiliki fungsi utama untuk membuat halaman website.
- CSS
  dalam pembuatan website HTML biasanya berpasangan dengan CSS untuk mempercantik website yang dibuat . CSS adalah bahasa Cascading Style Sheet dan biasanya digunakan untuk mengatur tampilan elemen yang tertulis dalam bahasa markup, seperti HTML. CSS berfungsi untuk memisahkan konten dari tampilan visualnya di situs

4.1.3 Application layer

- PHP
  Hypertext Preprocessor atau PHP adalah bahasa penulisan skrip open-source yang banyak digunakan dalam pemrograman atau pengembangan website (web development). Bahasa ini umumnya dijalankan dalam komunikasi sisi server, dan saat ini didukung oleh hampir semua sistem.

4.2 Testing

4.2.1 Testing Admin

- Admin melakukan login

```
Script Output
Task completed in 0.092 seconds

login berhasil!!
Status login : 0


PL/SQL procedure successfully completed.
```

- Admin menambahkan produk

  Disini admin ingin menambahkan produk Red Relaxed Fit Zip-through Hoodie pada tabel produk

```
PL/SQL procedure successfully completed.

Produk Red Relaxed Fit Zip-through Hoodie berhasil ditambahkan


PL/SQL procedure successfully completed.
```

	ID_PRODUCT	NAMA_BARANG	DESKRIPSI	HARGA	UKURAN	ID_KATEGORI	STOK
1	22	Red Relaxed Fit Zip-t...	Zip-through hoodie in sweatshirt fabric. Relaxed fit with a jersey-lined...	300000	L	4	15
2	1	Black Regular Fit Hoodie	Sweter bertudung lengan panjang berbahan kain sweter yang lembut dengan ...	300000	M	3	9
3	2	Black T-Shirt Oversize	Bahan Double-faced menggabungkan tampilan clean dan tekstur lembut, Desa...	200000	L	1	17

- Admin menambahkan stok pada suatu produk

  Admin ingin menambahkan stok sebanyak 10 pcs pada produk Red Relaxed Fit Zip-through Hoodie yang awalnya produk tersebut memiliki stok sebanyak 25 pcs menjadi 35 pcs.

- Jumlah stok sebelum ditambahkan

	ID_PRODUCT	NAMA_BARANG	DESKRIPSI	HARGA	UKURAN	ID_KATEGORI	STOK
1	22	Red Relaxed Fit Zip-t...	Zip-through hoodie in sweatshirt fabric. Relaxed fit with a jersey-lined...	300000	L	4	25
2	1	Black Regular Fit Hoodie	Sweter bertudung lengan panjang berbahan kain sweter yang lembut dengan ...	300000	M	3	9
3	2	Black T-Shirt Oversize	Bahan Double-faced menggabungkan tampilan clean dan tekstur lembut, Desa...	200000	L	1	17

- Jumlah stok setelah ditambahkan

```
stok berhasil ditambahkan


PL/SQL procedure successfully completed.
```

	ID_PRODUCT	NAMA_BARANG	DESKRIPSI	HARGA	UKURAN	ID_KATEGORI	STOK
1	22	Red Relaxed Fit Zip-t...	Zip-through hoodie in sweatshirt fabric. Relaxed fit with a jersey-lined...	300000	L	4	35
2	1	Black Regular Fit Hoodie	Sweter bertudung lengan panjang berbahan kain sweter yang lembut dengan ...	300000	M	3	9
3	2	Black T-Shirt Oversize	Bahan Double-faced menggabungkan tampilan clean dan tekstur lembut, Desa...	200000	L	1	17

- Admin menghapus produk

  Admin ingin menghapus produk Red Relaxed Fit Zip-through Hoodie pada tabel produk

- Isi tabel sebelum produk dihapus

	ID_PRODUCT	NAMA_BARANG	DESKRIPSI	HARGA	UKURAN	ID_KATEGORI	STOK
1	22	Red Relaxed Fit Zip-t...	Zip-through hoodie in sweatshirt fabric. Relaxed fit with a jersey-lined...	300000	L	4	35
2	1	Black Regular Fit Hoodie	Sweter bertudung lengan panjang berbahan kain sweter yang lembut dengan ...	300000	M	3	9
3	2	Black T-Shirt Oversize	Bahan Double-faced menggabungkan tampilan clean dan tekstur lembut, Desa...	200000	L	1	17

- Isi tabel setelah produk dihapus

```
Produk Red Relaxed Fit Zip-through Hoodie berhasil dihapus


PL/SQL procedure successfully completed.
```

	ID_PRODUCT	NAMA_BARANG	DESKRIPSI	HARGA	UKURAN	ID_KATEGORI	STOK
1	1	Black Regular Fit Hoodie	Sweter bertudung lengan panjang berbahan kain sweter yang lembut dengan ...	300000	M	3	9
2	2	Black T-Shirt Oversize	Bahan Double-faced menggabungkan tampilan clean dan tekstur lembut, Desa...	200000	L	1	17

4.2.2 Testing Customer

- Customer bernama Egi berhasil register

```
Register Success!!


PL/SQL procedure successfully completed.
```

	ID_USER	USERNAME	PASSWORD	FULL_NAME	NO_TELP	EMAIL	ROLE	SALDO
1	21	egi	egi12345	egi	085870012345	egi@gmail.com	Customer	0
2	5	ferdyfauzan	ferdi123	Ferdy Fauzan	087709847423	ferdif@gmail.com	Customer	1000000
3	6	daffaraihandika	daffa123	Daffa Raihandika	087700727562	draihandika@gmail.com	Customer	300000
4	7	alfhandre	andreaja	Andre Lutfiansyah	087700720823	andre@gmail.com	Customer	3000000
5	8	admin	admin321	Pick Me Store	087700727987	pickmestore@gmail.com	Admin	250000

- Retype password tidak sesuai dengan password

  Customer atas nama Raka membuat password “raka12345” tapi retype password nya tidak sama dengan password.

```
Register Failed
Password Not Match!!


PL/SQL procedure successfully completed.
```

- Customer membuat password kurang dari 8 karakter

  Customer atas nama raka membuat password “raka12” yang dimana password tersebut kurang dari 8 karakter

```
Register Failed
Password minimal 8 karakter!!


PL/SQL procedure successfully completed.
```

- Username already exist

  Customer melakukan register dengan username “egi” yang dimana username tersebut sudah dipakai customer lain.

```
Error starting at line : 36 in command -
BEGIN Register ('egi', 'egi12345', 'egi12345', 'egi', '085870012345', 'egi@gmail.com'); END;
Error report -
ORA-00001: unique constraint (DAFFARAIHANDIKA4.USERNAME__IDX) violated
ORA-06512: at "DAFFARAIHANDIKA4.REGISTER", line 5
ORA-06512: at line 1
00001. 00000 -  "unique constraint (%s.%s) violated"
*Cause:    An UPDATE or INSERT statement attempted to insert a duplicate key.
           For Trusted Oracle configured in DBMS MAC mode, you may see
           this message if a duplicate entry exists at a different level.
*Action:   Either remove the unique restriction or do not insert the key.
```

- Nomor telepon tidak sesuai format

  Nomor telepon yang dimasukan saat customer melakukan register tidak sesuai format

```
Error starting at line : 37 in command -
BEGIN Register ('raka', 'raka12345', 'raka12345', 'raka', '0076514567654', 'raka@gmail.com'); END;
Error report -
ORA-02290: check constraint (DAFFARAIHANDIKA4.PHONE_FORMAT) violated
ORA-06512: at "DAFFARAIHANDIKA4.REGISTER", line 5
ORA-06512: at line 1
02290. 00000 -  "check constraint (%s.%s) violated"
*Cause:    The values being inserted do not satisfy the named check

*Action:   do not insert values that violate the constraint.
```

- Email tidak sesuai format

  Email yang dimasukan saat customer melakukan register tidak sesuai format

```
Error starting at line : 4 in command -
BEGIN Register ('jakaa', 'jakaa12345', 'jakaa12345', 'jaka', '089762514235', 'jakacom'); END;
Error report -
ORA-02290: check constraint (DAFFARAIHANDIKA4.EMAIL_FORMAT) violated
ORA-06512: at "DAFFARAIHANDIKA4.REGISTER", line 5
ORA-06512: at line 1
02290. 00000 -  "check constraint (%s.%s) violated"
*Cause:    The values being inserted do not satisfy the named check

*Action:   do not insert values that violate the constraint.
```

- Customer melakukan login

  Customer dengan username “ferdyfauzan” ingin melakukan login

```
Status login : 0


PL/SQL procedure successfully completed.
```

- Customer mencari produk

  Customer mencari product dengan nama Black Regular Fit Hoodie

```
PL/SQL procedure successfully completed.

ResultSet #1

ID_PRODUCT NAMA_BARANG                                    DESKRIPSI                                                                                          HARGA UKUR ID_KATEGORI      STOK
---------- ---------------------------------------------- ---------------------------------------------------------------------------------------------------- -------- ---- ----------- ----------
         1 Black Regular Fit Hoodie                       Sweter bertudung lengan panjang berbahan kain sweter yang lembut dengan saku kan     300000 M            3        12
```

- Customer mencari suatu produk menggunakan kategori

  Customer mencari product yang terdapat pada kategori hoodie

```
ResultSet #1

NAMA_BARANG                                    DESKRIPSI                                                                                          HARGA UKUR      STOK
---------------------------------------------- ---------------------------------------------------------------------------------------------------- -------- ---- ----------
Black Regular Fit Hoodie                       Sweter bertudung lengan panjang berbahan kain sweter yang lembut dengan saku kan     300000 M           12
```

- Customer memesan produk

  Customer dengan nama andre memesan produk Black Regular Fit Hoodie ukuran M sebanyak 3 buah

	ID_ORDERITEM	QUANTITY	ID_PRODUCT	ID_ORDERDETAILS	TOTAL_PERPRODUK
1	38	3	1	46	900000

- Customer melakukan pembayaran

  Customer dengan nama andre membayar pesanan produk Black Regular Fit Hoodie , status menjadi lunas.

	ID_PAYM...	STATUS	ID_ORDERDETAILS	CREATED_AT
1	26	Lunas	46	06-12-2022 22.14.19,000000000

- Customer melakukan top up saldo

  Customer atas nama egi ingin melakukan top up saldo sebesar 1000000 , saldo awal yang dimiliki customer adalah 0

– Sebelum melakukan top up

	ID_USER	USERNAME	PASSWORD	FULL_NAME	NO_TELP	EMAIL	ROLE	SALDO
1	21	egi	egi12345	egi	085870012345	egi@gmail.com	Customer	0
2	5	ferdyfauzan	ferdi123	Ferdy Fauzan	087709847423	ferdif@gmail.com	Customer	1000000
3	6	daffaraihandika	daffa123	Daffa Raihandika	087700727562	draihandika@gmail.com	Customer	400000
4	7	alfhandre	andreaja	Andre Lutfiansyah	087700720823	andre@gmail.com	Customer	3000000
5	8	admin	admin321	Pick Me Store	087700727987	pickmestore@gmail.com	Admin	400000

– Setelah melakukan top up

	ID_USER	USERNAME	PASSWORD	FULL_NAME	NO_TELP	EMAIL	ROLE	SALDO
1	21	egi	egi12345	egi	085870012345	egi@gmail.com	Customer	1000000
2	5	ferdyfauzan	ferdi123	Ferdy Fauzan	087709847423	ferdif@gmail.com	Customer	1000000
3	6	daffaraihandika	daffa123	Daffa Raihandika	087700727562	draihandika@gmail.com	Customer	400000
4	7	alfhandre	andreaja	Andre Lutfiansyah	087700720823	andre@gmail.com	Customer	3000000
5	8	admin	admin321	Pick Me Store	087700727987	pickmestore@gmail.com	Admin	400000

- Customer membatalkan pesanan

  Customer dengan username “alfhandre” membatalkan pesanannya

	ID_ORDERDETAILS	HARGA_TOTAL	ID_USER	STATUS	CREATED_AT	MODIFIED_AT
1	46	0	7	Canceled	06-12-2022 22.10.46,000000000	(null)

- Customer mengganti password

  Customer dengan username “egi” mengganti passwordnya yang semula “egi12345” menjadi “egisatria123”

```
Password berhasil diubah!!


PL/SQL procedure successfully completed.
```

	ID_USER	USERNAME	PASSWORD	FULL_NAME	NO_TELP	EMAIL	ROLE	SALDO
1	21	egi	egisatria123	egi	085870012345	egi@gmail.com	Customer	1000000
2	5	ferdyfauzan	ferdi123	Ferdy Fauzan	087709847423	ferdif@gmail.com	Customer	1000000
3	6	daffaraihandika	daffa123	Daffa Raihandika	087700727562	draihandika@gmail.com	Customer	400000

BAB V
KESIMPULAN

Aplikasi PickMeStore merupakan Aplikasi berbasis web , yang dibangun dengan menggunakan Bahasa pemrograman PHP, Menggunakan Oracle sebagai tools untuk membuat database yang berfungsi untuk menyimpan semua data Customer,Admin dan produk. Aplikasi ini bertujuan untuk memudahkan para customer untuk membeli pakaian tanpa harus datang ke toko secara langsung.

Pada aplikasi yang kami buat juga memudahkan customer dalam melihat informasi barang yang dibutuhkannya , juga memiliki fitur- fitur yang dapat membantu customer dalam berbelanja seperti :

1. fitur mencari produk
2. fitur mencari produk berdasarkan kategori.

BAB VI
LESSON LEARNED

6.1 Muhammad Daffa Raihandika

Lesson learn yang saya dapatkan pada tugas besar Sistem Basis Data ini adalah saya dapat mengeksplor lebih jauh mengenai PLSQL/Oracle. Selain itu saya juga menjadi lebih mengerti bagaimana cara mendesain suatu sistem database dan mengimplementasikannya, serta bagaimana membuat sistem database tersebut berjalan sesuai dengan business rules yang telah direncanakan.

6.2 Mochamad Ferdy Fauzan

Lesson learn yang saya dapatkan pada tugas besar Sistem Basis Data Praktikum ini adalah saya dapat memahi sedikit demi sedikit bagaimana suatu database dibuat. Pada matkul ini, tools yang saya gunakan ialah Oracle Data Modeler dan Oracle SQL Developer yang menggunakan bahasa PL/SQL.

Yang paling berkesan menurut saya ialah ketika mendesain dan membuat konsep system database dari awal. Namun banyak hal yang tidak saya dapat selama perkuliahan sehingga saya lebih banyak mengeksplor mandiri. Dalam pengerjaan tugas besar ini kerja sama tim sangatlah penting dan komunikasi adalah kuncinya.

6.3 Raka Mahardika Maulana

Lesson learn yang saya dapatkan pada tugas besar sistem basis data adalah membuat saya lebih mengerti bagaimana cara menggunakan bahasa pemrograman query, juga membuat saya lebih paham bagaimana membuat bisnis rules untuk mendukung konsep yang telah dibuat . Dan membuat saya lebih terlatih dalam bekerja dengan cara team work.

DAFTAR PUSTAKA

The all-in-one workspace for your notes, tasks, wikis, and databases. Notion. (n.d.). Retrieved December 6, 2022, from https://obsidian-coconut-380.notion.site/Normalisasi-8f5ffeb726a246c28d9fc3c4f1b2016c

brilianjoey. (2020, November 1). *Bab 9 ppb: BPMN Elemen Dasar Dan Gateway*. YouTube. Retrieved December 6, 2022, from https://www.youtube.com/watch?v=Z6FkYRMv4bk

brilianjoey. (2020, November 17). *Bab 10 ppb : BPMN Swimlane*. YouTube. Retrieved December 6, 2022, from https://www.youtube.com/watch?v=KCMdYayxrm8&t=220s

Feri. (2022, April 23). *BPMN: Pengertian, Notasi Dan Contoh*. DosenIT.com. Retrieved December 6, 2022, from https://dosenit.com/kuliah-it/business-process-modeling-notation

Guntoro. (2022, December 5). *√ lengkap! contoh DFD dilengkapi Dengan Penjelasannya*. Badoy Studio. Retrieved December 6, 2022, from https://badoystudio.com/contoh-dfd/

Oracle Database 21C - Get Started. Oracle Help Center. (2022, October 18). Retrieved December 6, 2022, from https://docs.oracle.com/en/database/oracle/oracle-database/21/index.html

What is a data flow diagram. Lucidchart. (n.d.). Retrieved December 6, 2022, from https://www.lucidchart.com/pages/data-flow-diagram

What's an example of good e-commerce database design? fabric. (2022, November 4). Retrieved December 6, 2022, from https://fabric.inc/blog/ecommerce-database-design-example/

Www.techonthenet.com. (n.d.). Oracle tutorial. Retrieved December 6, 2022, from https://www.techonthenet.com/oracle/index.php

YouTube. (2020, October 23). *DS-09: Normalisasi database 1NF 2NF hingga 3NF*. YouTube. Retrieved December 6, 2022, from https://www.youtube.com/watch?v=UC_tJx4MBgk

YouTube. (2021, September 28). *Belajar Normalisasi database Dari Struk Transaksi (UNF, 1NF, 2NF, 3NF) + bahas contoh soal*. YouTube. Retrieved December 6, 2022, from https://www.youtube.com/watch?v=wpLokB3Cd2I