SDD- XXXX

(Kelompok 2)

**(Project Aplikasi Toko Lia)**

#### Software Design Document v.0.1

|  |  |
| --- | --- |
| Nama : | Alvon Tritara |
|  | David Dwiputranto |
|  | Hasbie Muhammad Irsyad |
|  | Nuramalia Putri Hapsari |
|  |  |
| Bagian : | Designer |

Date: (06/11/2023)

**TABLE OF CONTENTS**

1.0 INTRODUCTION 4

1.1 Purpose 4

1.2 Scope 4

1.3 Overview 4

1.4 Reference Material 4

1.5 Definitions and Acronyms 4

2.0 SYSTEM OVERVIEW 4

3.0 SYSTEM ARCHITECTURE 4

3.1 Architectural Design 4

3.2 Decomposition Description 5

3.3 Design Rationale 5

4.0 DATA DESIGN 5

4.1 Data Description 5

4.2 Data Dictionary 5

5.0 COMPONENT DESIGN 5

6.0 HUMAN INTERFACE DESIGN 5

6.1 Overview of User Interface 5

6.2 Screen Images 6

6.3 Screen Objects and Actions 6

7.0 REQUIREMENTS MATRIX 6

8.0 APPENDICES 6

## PENDAHULUAN

## Tujuan

Dokumen ini dibuat untuk menggambarkan arsitektur dari sistem pada aplikasi Toko Lia yang rinci sehingga dapat digunakan untuk menjadi panduan untuk programmer dalam membangun aplikasi. Pengguna utama dari dokumen ini adalah anggota Kelompok 2, khususnya untuk PIC Programmer.

## Ruang Lingkup

Aplikasi Toko Lia adalah aplikasi yang dibuat dengan tujuan membantu proses pencatatan transaksi harian di Toko Lia. Aplikasi menggunakan dasar dari sistem sebuah aplikasi kasir, dimana transaksi akan disimpan dalam basis data. Kumpulan transaksi yang disimpan kemudian dapat digunakan untuk membuat laporan laba rugi harian/bulanan, pencatatan hutang, hingga informasi perubahan stok barang. Setiap desain yang dibuat pada project ini bersifat khusus dan terpersonalisasi pada kegiatan rutin yang terjadi di Toko Lia, sehingga beberapa desain mungkin hanya dapat berlaku pada project ini dan berlandaskan kemudahan penggunaan untuk rutinitas yang dilakukan pada Toko Lia

## Ikhtisar

Software Design Document (SDD) ini adalah dokumen yang dibuat sebagai panduan untuk anggota kelompok 2 dalam membangun software aplikasi Toko Lia, sehingga setiap proses pembuatan komponen bisa sesuai dengan kebutuhan dan tidak terlewat.

|  |  |
| --- | --- |
| **Section** | **Isi** |
| 1 Pendahuluan | * 1. Tujuan   2. Ruang Lingkup   3. Ikhtisar   4. Referensi   5. Definisi dan Singkatan |
| 2 Gambaran Umum Sistem | Fitur fungsional, lingkungan implementasi |
| 3 Arsitektur Sistem | 3.1 Rancangan Arsitektur  3.2 Deskripsi Dekomposisi  3.3 Alasan Perancangan |
| 4 Rancangan Data | 4.1 Deskripsi Data  4.2 Kamus Data |
| 5 Rancangan Komponen |  |
| 6 Rancangan Antarmuka | 6.1 Gambaran Umum Antarmuka  6.2 Tampilan Layar  6.3 Objek Layar dan Tindakan |
| 7 Matriks Persyaratan |  |
| 8 Lampiran |  |

Susunan tim pengembang “Kelompok 2” :

|  |  |
| --- | --- |
| **Nama** | **Posisi** |
| Nuramalia Putri Hapsari | Project Manager, System Analyst |
| Alvon Tritara | UI / UX Designer |
| Hasbie Muhammad Irsyad | Programmer |
| David Dwiputranto | Programmer, Documenter |

## Referensi

* SDD Template by Dr. Lulu Chaerani M.
* SRS v.1.0 Aplikasi Toko Lia by Kelompok 2
* SDD v.1.9 Aplikasi Busayu 15 April 2018 on github : firstiaulyaa/RPL-D-5

## Definisi dan Singkatan

|  |  |
| --- | --- |
| **Istilah** | **Definisi** |
| DBMS | Database Management System, perangkat lunak pengelola basis data |
| SDK | Software Development Kit, package yang berisi fungsi-fungsi pre-programmed yang dapat digunakan untuk membangun sebuah perangkat lunak |
| SDD | Software Design Document |
|  |  |

## Gambaran Umum Sistem

Proyek sistem informasi ini akan menghasilkan produk berupa aplikasi berbasis android bernama Aplikasi Toko Lia, dimana aplikasi ini akan difungsikan untuk menggantikan kegiatan bisnis pada Toko Lia yang selama ini pencatatan dilakukan menggunakan tulis tangan dan tidak ada perhitungan keuntungan dari penjualan. Produk aplikasi ini akan difungsikan untuk menggantikan kegiatan-kegiatan berulang yang dapat dilakukan lebih akurat dan cepat dengan dibantu aplikasi sistem informasi.

Fitur fungsional yang dapat digunakan oleh class user pada sistem ini berupa :

|  |  |  |
| --- | --- | --- |
| **No.** | **Fungsi** | **Keterangan** |
| 1. | Setup Produk | Membuat produk-produk dagangan dengan informasi lengkap harga modal dan harga jual beserta tanggal kadaluwarsa hingga jumlah stok yang tersedia |
| 2. | Update Stok | Merubah stok dari setiap produk (menambahkan atau mengurangi), sebagai bentuk penanggulangan kesalahan input data ataupun kegiatan restok rutin |
| 3. | Pencarian | Fitur pencarian produk dengan kata kunci berdasarkan nama |
| 4. | Fitur Kasir | Fitur untuk membuat transaksi dan menyatakan transaksi tersebut sebagai pembelian, hutang, penggunaan pribadi, ataupun dibuang karena melewati masa kadaluwarsa |
| 5. | Kasbon | Fitur untuk menambahkan transaksi ke tab kasbon pelanggan untuk mencatat total transaksi yang belum terbayarkan oleh pelanggan |

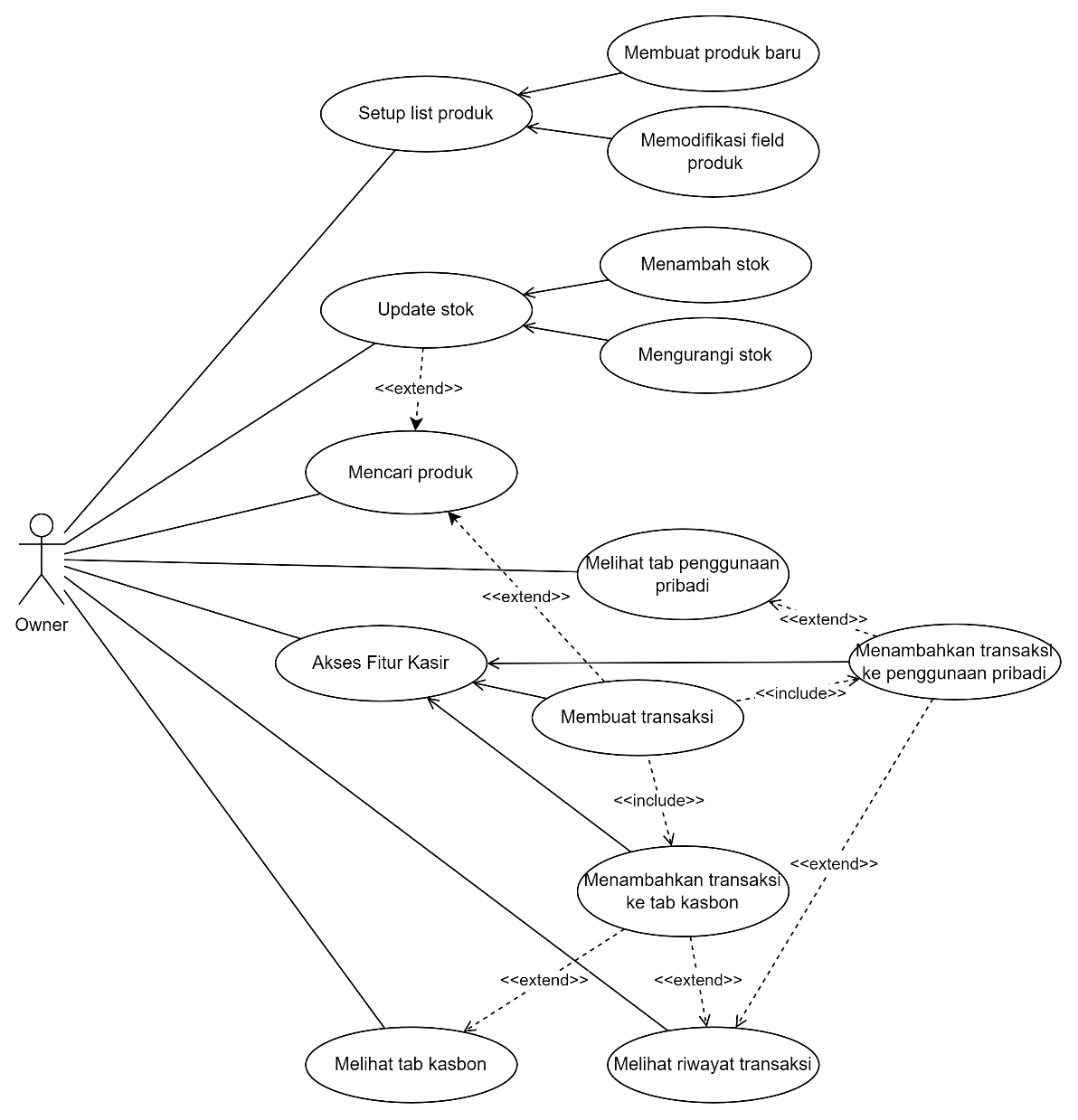
Dalam pembuatan perangkat lunak ini, dibuat menggunakan :

* Development Tool : Android Studio ( SDK Platform : Android 12.0 API Level 31 )
* DBMS : SQLite (kemungkinan berubah)

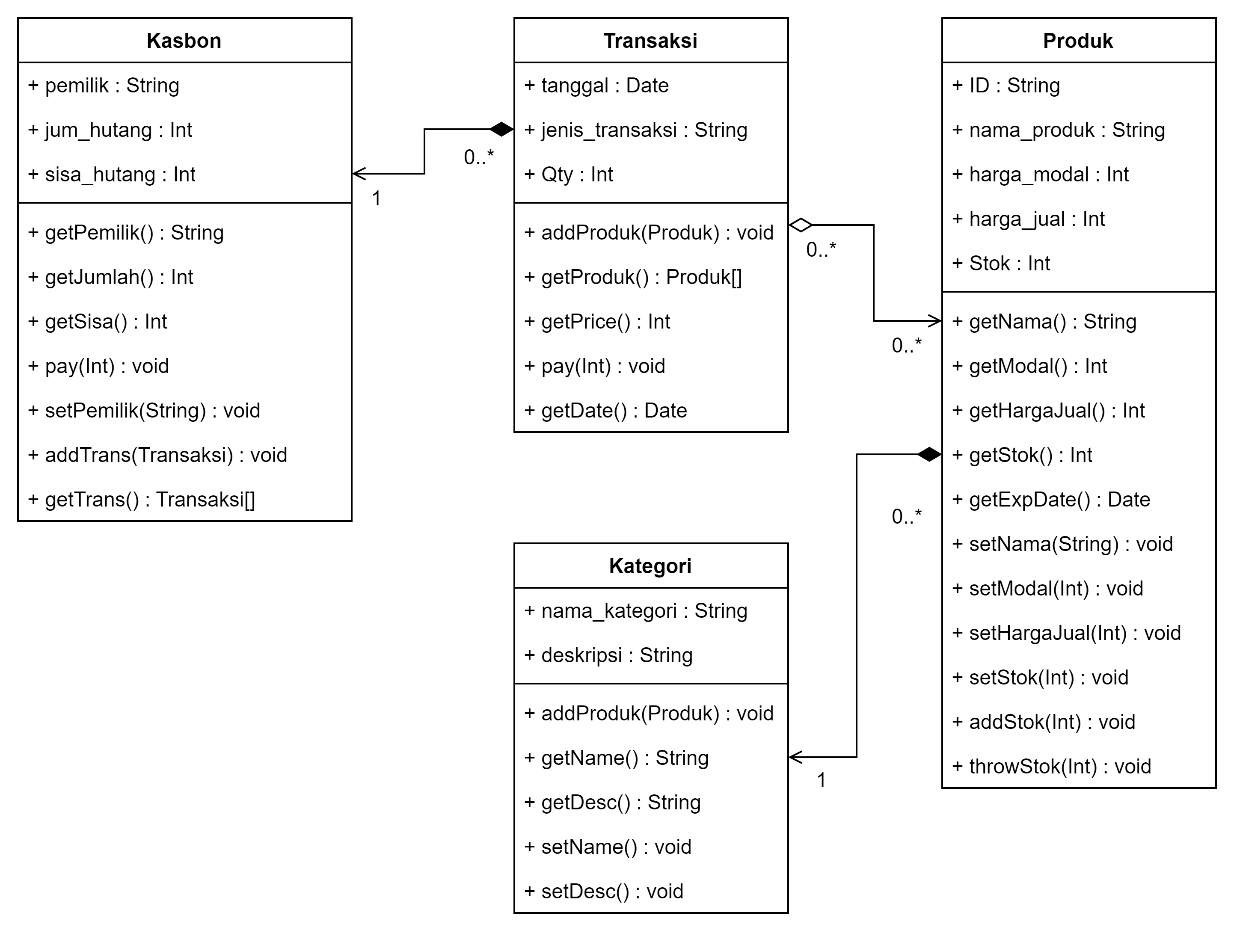
Kelompok pengguna dari aplikasi hanya ada satu yaitu Owner, dimana Owner memiliki hak fungsionalitas penuh pada perangkat lunak. Karena hanya ada satu class user, maka tidak dibuat basis data yang mengatur macam-macam pengguna dan tidak dibuat pengaturan privileges.

## ARSITEKTUR SISTEM

## Rancangan Arsitektur



Fungsionalitas dari aplikasi dijelaskan dengan use case diagram di atas, dimana pemrograman dibuat berbasis objek yang dapat dirincikan dengan class diagram berikut

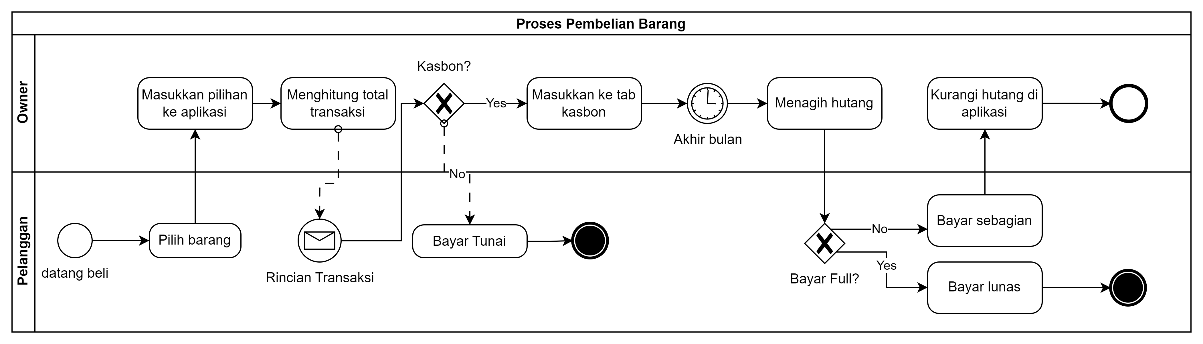


Develop a modular program structure and explain the relationships between the modules to achieve the complete functionality of the system. This is a high level overview of how responsibilities of the system were partitioned and then assigned to subsystems. Identify each high level subsystem and the roles or responsibilities assigned to it. Describe how these subsystems collaborate with each other in order to achieve the desired functionality. Don’t go into too much detail about the individual subsystems. The main purpose is to gain a general understanding of how and why the system was decomposed, and how the individual parts work together. Provide a diagram showing the major subsystems and data repositories and their interconnections. Describe the diagram if required.

## Deskripsi Dekomposisi

Provide a decomposition of the subsystems in the architectural design. Supplement with text as needed. You may choose to give a functional description or an object­oriented (OO) description. For a functional description, put top­level data flow diagram (DFD) and structural decomposition diagrams. For an OO description, put subsystem model, object diagrams, generalization hierarchy diagram(s) (if any), aggregation hierarchy diagram(s) (if any), interface specifications, and sequence diagrams here.

Dekomposisi pada



## Alasan Perancangan

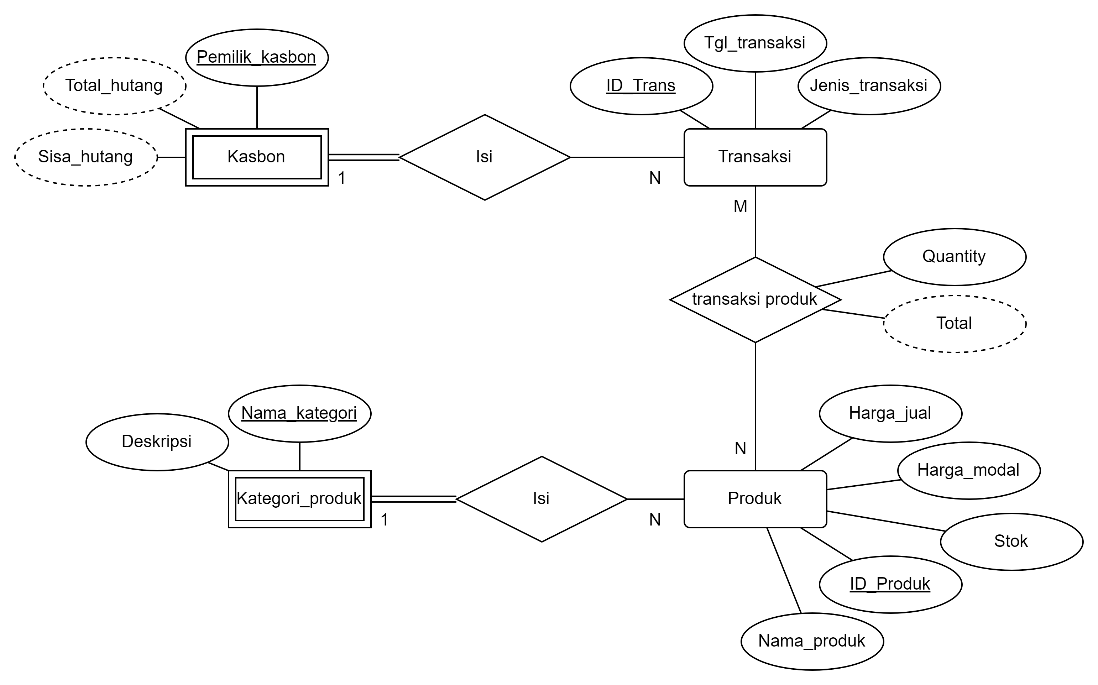
Discuss the rationale for selecting the architecture described in 3.1 including critical issues and trade/offs that were considered. You may discuss other architectures that were considered, provided that you explain why you didn’t choose them.

## RANCANGAN DATA

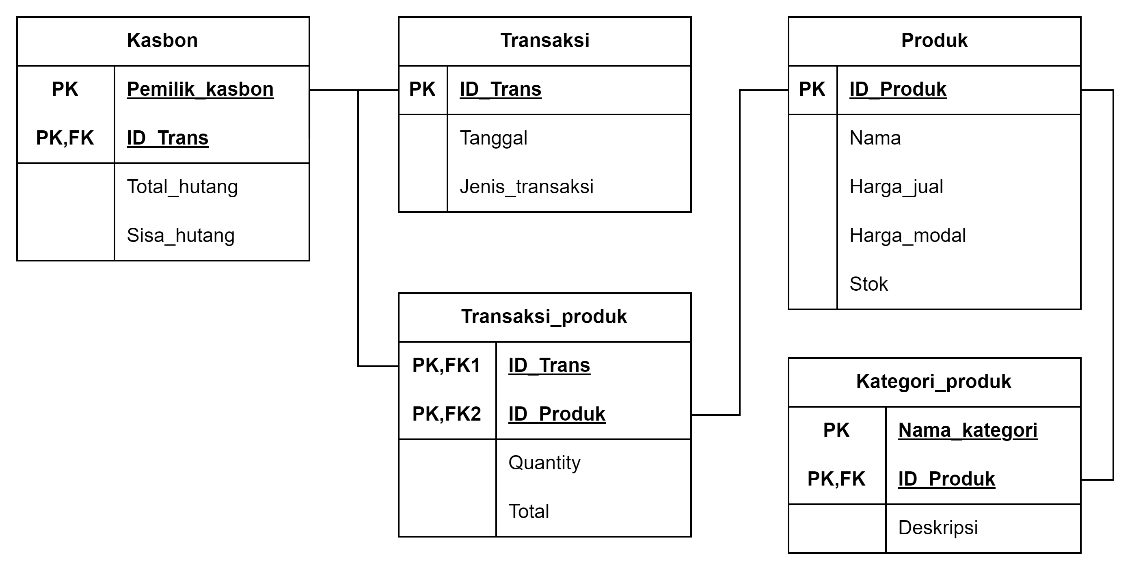
## Deskripsi Data

Sumber data pada aplikasi berasal dari pengaturan dan input dari pengguna yang akan disimpan ke dalam basis data yang dirincikan ke dalam :

* Entity Relationship Diagram (ERD) :



Berdasarkan ERD yang tersebut, akhirnya akan dibuat 5 (lima) buah tabel yang saling berelasi :



Setiap input dari pengguna aplikasi, akan disimpan ke dalam basis data yang terdiri dari 5 (lima) tabel : Tabel Kasbon, Tabel Transaksi, Tabel Transaksi\_produk, Tabel Produk, dan Tabel Kategori\_produk. Kecuali ID\_Trans dan ID\_Produk, setiap isi field berasal dari input pengguna di dalam aplikasi. ID\_Trans dan ID\_Produk dibuat di dalam program secara otomatis menggunakan rules yang tidak perlu campur tangan pengguna.

*Explain how the information domain of your system is transformed into data structures. Describe how the major data or system entities are stored, processed and organized. List any databases or data storage items.*

## Kamus Data

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Nama Tabel | | : | | Produk | |
| Keterangan | | : | | Tabel untuk menyimpan record tentang produk yang dijual di Toko Lia | |
| Primary Key | | : | | ID\_Produk | |
| Jumlah Field | | : | | 5 | |
| **No.** | **Nama Field** | | **Tipe Data (Size)** | | **Deskripsi** |
| 1. | ID\_Produk\* | | Char(10) | | Kode identifikasi produk |
| 2. | Nama | | Char(50) | | Nama produk yang dijual di toko |
| 3. | Harga\_jual | | Integer | | Harga jual produk |
| 4. | Harga\_modal | | Integer | | Harga perolehan dari produk |
| 5. | Stok | | Integer | | Jumlah barang yang tersedia di toko |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Nama Tabel | | : | | Transaksi | |
| Keterangan | | : | | Tabel untuk menyimpan identifier dari transaksi tanpa isi produk | |
| Primary Key | | : | | ID\_Trans | |
| Jumlah Field | | : | | 3 | |
| **No.** | **Nama Field** | | **Tipe Data (Size)** | | **Deskripsi** |
| 1. | ID\_Trans\* | | Char(10) | | Kode identifikasi transaksi |
| 2. | Tanggal | | Date | | Tanggal dibuatnya transaksi |
| 3. | Jenis\_transaksi | | Char(20) | | Jenis transaksi kategorikal (akan diset dalam program : “pembelian”, “penggunaan sendiri”, “hutang”, “kadaluwarsa”) |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Nama Tabel | | : | | Transaksi\_produk | |
| Keterangan | | : | | Tabel untuk menyimpan isian produk yang ada di dalam transaksi | |
| Primary Key | | : | | ID\_Trans, ID\_Produk (composite) | |
| Jumlah Field | | : | | 4 | |
| **No.** | **Nama Field** | | **Tipe Data (Size)** | | **Deskripsi** |
| 1. | ID\_Trans\* | | Char(10), FK | | Kode identifikasi transaksi |
| 2. | ID\_Produk\* | | Char(10), FK | | Kode identifikasi produk |
| 3. | Quantity | | Integer | | Jumlah produk yang ada di dalam transaksi |
| 4. | Total | | Integer | | Total harga transaksi |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Nama Tabel | | : | | Kategori\_produk | |
| Keterangan | | : | | Tabel untuk menyimpan kategori, dimana produk akan dikategorikan | |
| Primary Key | | : | | Nama\_kategori, ID\_Produk (composite) | |
| Jumlah Field | | : | | 3 | |
| **No.** | **Nama Field** | | **Tipe Data (Size)** | | **Deskripsi** |
| 1. | Nama\_kategori\* | | Char(20) | | Nama dari kategori yang dibuat |
| 2. | ID\_Produk\* | | Char(10), FK | | Kode identifikasi produk |
| 3. | Deskripsi | | Char(50) | | Deskripsi singkat yang dapat menjelaskan kategori |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Nama Tabel | | : | | Kasbon | |
| Keterangan | | : | | Tabel yang menyimpan akun kasbon beserta total hutang yang dimiliki oleh akun | |
| Primary Key | | : | | Pemilik\_kasbon, ID\_Trans | |
| Jumlah Field | | : | | 4 | |
| **No.** | **Nama Field** | | **Tipe Data (Size)** | | **Deskripsi** |
| 1. | Pemilik\_kasbon\* | | Char(20) | | Nama dari pemilik kasbon |
| 2. | ID\_Trans\* | | Char(10), FK | | Kode identifikasi transaksi |
| 3. | Total\_hutang | | Integer | | Total nilai transaksi yang ada di akun kasbon |
| 4. | Sisa\_hutang | | Integer | | Sisa nilai transaksi yang belum terbayar di akun kasbon |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Class | | : | | Kasbon | |
| Keterangan | | : | |  | |
| Primary Key | | : | | PemilikID | |
| Jumlah Field | | : | | 10 | |
| **No.** | **Nama Method** | | **Return Type** | | **Parameter** |
| 1. | Pemilik\* | | String | |  |
| 2. | Jum\_hutang | | Integer | |  |
| 3. | Sisa\_hutang | | Integer | |  |
| 4. | getPemilik() | | String | |  |
| 5. | getJumlah() | | Integer | |  |
| 6. | getSisa() | | Integer | |  |
| 7. | pay(int amount) | | Void | | Satu parameter bertipe **int** yang disebut **amount** |
| 8. | setPemilik | | Void | | Satu parameter bertipe **String** yang disebut **pemilikBaru** |
| 9. | AddTrans | | Void | | Satu parameter bertipe **Transaksi** yang disebut **transaksiBaru** |
| 10. | getTrans | | Transaksi[] | |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Class | | : | | Transaksi | |
| Keterangan | | : | |  | |
| Primary Key | | : | | Jenis\_transaksi | |
| Jumlah Field | | : | | 8 | |
| **No.** | **Nama Method** | | **Return Type** | | **Parameter** |
| 1. | Tanggal | | Date | |  |
| 2. | Jenis\_transaksi\* | | String | |  |
| 3. | Quantity | | Integer | |  |
| 4. | addProduk(Produk) | | Void | | Satu parameter bertipe **Produk** yang disebut **produkBaru** |
| 5. | getProduk() | | Produk[] | |  |
| 6. | getPrice() | | Integer | |  |
| 7. | pay(int) | | Void | | Satu parameter bertipe **int** yang disebut **amount** |
| 8. | getDate() | | Date | |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Class | | : | | Kategori | |
| Keterangan | | : | |  | |
| Primary Key | | : | | nama\_kategori | |
| Jumlah Field | | : | | 7 | |
| **No.** | **Nama Method** | | **Return Type** | | **Parameter** |
| 1. | Nama\_kategori\* | | String | |  |
| 2. | Deskripsi | | String | |  |
| 3. | addProduk(Produk) | | Void | | Satu parameter bertipe **Produk** yang disebut **produkBaru** |
| 4. | getName() | | String | |  |
| 5. | getDesc() | | String | |  |
| 6. | setName() | | Void | | Satu parameter bertipe **String** yang disebut **newName** |
| 7. | setDesc() | | Void | | Satu parameter bertipe **String** yang disebut **newDesc** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Class | | : | | Produk | |
| Keterangan | | : | |  | |
| Primary Key | | : | | ID | |
| Jumlah Field | | : | | 15 | |
| **No.** | **Nama Method** | | **Return Type** | | **Parameter** |
| 1. | ID\* | | String | |  |
| 2. | Nama\_produk | | String | |  |
| 3. | Harga\_modal | | Integer | |  |
| 4. | Harga\_jual | | Integer | |  |
| 5. | Stok | | Integer | |  |
| 6. | getNama() | | String | |  |
| 7. | getModal() | | Integer | |  |
| 8. | getHargaJual() | | Integer | |  |
| 9. | getStok() | | Integer | |  |
| 10. | getNama(String) | | Void | | Satu parameter bertipe **String** yang disebut **namaBaru** |
| 11. | setModal(Int) | | Void | | Satu parameter bertipe **Int** (Integer) yang disebut **modalBaru** |
| 12. | setHargaJual(Int) | | Void | | Satu parameter bertipe **Int** (Integer) yang disebut **hargaJualBaru** |
| 13. | setStok(Int) | | Void | | Satu parameter bertipe **Int** (Integer) yang disebut **stokBaru** |
| 14. | addStok(Int) | | Void | | Satu parameter bertipe **Int** (Integer) yang disebut **tambahanStok** |
| 15. | throwStok(Int) | | Void | | Satu parameter bertipe **Int** (Integer) yang disebut **penguranganStok** |

## COMPONENT DESIGN

In this section, we take a closer look at what each component does in a more systematic way. If you gave a functional description in section 3.2, provide a summary of your algorithm for each function listed in 3.2 in procedural description language (PDL) or pseudocode. If you gave an OO description, summarize each object member function for all the objects listed in 3.2 in PDL or pseudocode. Describe any local data when necessary.

## HUMAN INTERFACE DESIGN

## Overview of User Interface

Describe the functionality of the system from the user’s perspective. Explain how the user will be able to use your system to complete all the expected features and the feedback information that will be displayed for the user.

## Screen Images

Display screenshots showing the interface from the user’s perspective. These can be hand­ drawn or you can use an automated drawing tool. Just make them as accurate as possible. (Graph paper works well.)

## Screen Objects and Actions

A discussion of screen objects and actions associated with those objects.

## REQUIREMENTS MATRIX

Provide a cross­reference that traces components and data structures to the requirements in your softwarerequirements specification (SWRS) document.

Use a tabular format to show which system components satisfy each of the functional requirements from the SWRS. Refer to the functional requirements by the numbers/codes that you gave them in the SWRS.

## APPENDICES

*This section is optional.*

Appendices may be included, either directly or by reference, to provide supporting details that could aid in the understanding of the Software Design Document.