**DOKUMEN PEMBANGUNAN PERANGKAT LUNAK**

**PERUSAHAAN TAS KULIT**

****

**Dipersiapkan Oleh :**

**Tim :** Singsuksesnya

|  |  |
| --- | --- |
| Nama | NPM |
| Dicky Saputra | 2242007 |
| Galih Raihan Anshori | 2242029 |
| Muhammad Ariq Faridzki | 2242004 |
| Zaidan Ghiffari Azhar | 2242021 |

**SEKOLAH TINGGI MANAJEMEN INFORMATIKA & KOMPUTER**

**“AMIKBANDUNG”**

**2023**

Lembar pengesahan

Telah disetujui dan disahkan, Laporan Akhir Advanced Analysis System and Design dari:

Nama TIM : Singsuksesnya

Dengan judul : Perusahaan Tas Kulit

**“ DOKUMEN PEMBANGUNAN PERANGKAT LUNAK “**

**PERUSAHAAN TAS KULIT**

Bandung, 4 April 2023

Menyetujui,

Dosen Pengampu, Team Leader,

( Octanty proplayer discrete ) ( Muhammad Ariq Faridzki )

# **DAFTAR ISI**

[**DAFTAR ISI** i](#_Toc503691362)

[**DAFTAR TABEL** ii](#_Toc503691363)

[**DAFTAR GAMBAR** iv](#_Toc503691364)

[**DAFTAR SIMBOL** vii](#_Toc503691365)

[**1.** **BAB I PENDAHULUAN** 1](#_Toc503691366)

[**1.1.** **Pendefinisian Lingkup** 1](#_Toc503691367)

[**1.2.** **Pembagian Kerja Tim** 1](#_Toc503691368)

[**1.3.** **Penjelasan Proses Bisnis** 1](#_Toc503691369)

[**2.** **BAB II PROSES BISNIS DAN REQUIREMENT** 2](#_Toc503691370)

[**2.1.** **PROSES BISNIS** 2](#_Toc503691371)

[**2.1.1.** **Pendefinisian Model Bisnis** 2](#_Toc503691372)

[**2.1.2.** **Business Use Case** 2](#_Toc503691373)

[**2.1.3.** **Aktivitas Business Use Case (Activity Diagram)** 3](#_Toc503691374)

[**2.1.4.** **Business Object Model** 7](#_Toc503691375)

[**2.2.** **REQUIREMENT** 10](#_Toc503691376)

[**2.2.1.** **User Requirement Perangkat Lunak** 10](#_Toc503691377)

[**2.2.2.** **Daftar Kebutuhan Perangkat Lunak** 10](#_Toc503691378)

[**2.2.3.** **Model Use Case** 11](#_Toc503691379)

[**3.** **BAB III ANALIS DAN DESAIN** 14](#_Toc503691380)

[**3.1.** **ANALISIS** 14](#_Toc503691381)

[**3.1.1.** **Realisasi Use Case** 14](#_Toc503691382)

[**3.2.** **DESAIN** 62](#_Toc503691383)

[**3.2.1.** **Coding Standard and Naming Convention** 62](#_Toc503691384)

[**3.2.2.** **Diagram Kelas Perancangan** 70](#_Toc503691385)

[**3.2.3.** **Perancangan Data dengan ORM** 79](#_Toc503691386)

[**3.2.4.** **Deskripsi Algoritma / Query Operasi** 79](#_Toc503691387)

[**3.2.5.** **Antarmuka Perangkat Lunak** 83](#_Toc503691388)

[**3.2.6.** **Perancangan Arsitektur / MVC** 89](#_Toc503691389)

[**3.2.7.** **Daftar File** 90](#_Toc503691390)

[**3.2.8.** **Daftar Data** 92](#_Toc503691391)

# **DAFTAR TABEL**

[Tabel 1.1 Tabel Pembagian Kerja Tim 1](#_Toc503691392)

[Tabel 1.2 Tabel Penjelasan Proses Bisnis 1](#_Toc503691393)

[Tabel 2.1 Tabel Daftar Business Use Case 2](#_Toc503691394)

# **DAFTAR GAMBAR**

[Gambar 2.1 Gambar Business Use Case Diagram 2](#_Toc503691439)

[Gambar 2.2 Gambar Activity Diagram – Pemesanan Makanan 3](#_Toc503691440)

[Gambar 2.3 Activity Diagram – Pembayaran Makanan 4](#_Toc503691441)

[Gambar 2.4 Activity Diagram – Gabungan 5](#_Toc503691442)

# **DAFTAR SIMBOL**

| **No.** | **Simbol** | **Nama Simbol** | **Deskripsi** |
| --- | --- | --- | --- |
| 1. | Capture | Use Case | Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor |
| 2. | Usecase Actor | Usecase Actor | Aktor adalah seseorang yang berinteraksi langsung dengan system. |
| 3. | Business Actor | Business Usecase Actor | Business Usecase Actor peran yang dilakukan oleh seseorang atau sekelompok orang dimana peran ini memberikan sesuatu agar suatu proses bisnis dapat berjalan. |
| 4. | Business Usecase | Business Use case | Menggambarkan suatu himpunan aliran kerja dalam organisasi yang mempunyai nilai terhadap business actor. |
| 5. | **C:\Users\re_lust\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Usecase Boundary.png** | Usecase System Boundary | Batasan system dari usecase yang menandakan luas system hanya sebesar yang tertera. |
| 6. | **C:\Users\re_lust\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Sequence Form.png** | Sequence Form / Boundary | Sequence Form biasanya berupa user interface yang berhubungan langsung dengan actor. |
| 7. | **Sequence Controller** | Sequence Controller | Sequence Controller biasanya berupa internal interface atau bagian dalam dari system yang bertugas memproses |
| 8. | **C:\Users\re_lust\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Sequence Entity.png** | Sequence Entity | Sequence Entity berperan seperti database pada sebuah system. |
| 9. | **Sequence Call** | Sequence Call | Sequence Call berupa method yang akan dieksekusi oleh controller atau entity. |
| 10. | **C:\Users\re_lust\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Sequence Sent.png** | Sequence Message | Sequence Message merupakan interaksi antara user atau actor dengan system melalui form. |
| 11. | **Sequence Return** | Sequence Return | Sequence Return untuk menginformasikan hasil yang terjadi setelah method tereksekusi. |
| 12. | C:\Users\re_lust\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Sequence Stimulus.png | Sequence Lifeline | Lifeline adalah eksekusi objek selama sequence. |
| 13. | Activity Action | Activity Action | Activity Action merupakan langkah-langkah dalam sebuah proses yang berguna untuk menggambarkan rincian aktivitas yang terjadi. |
| 14. | Activity Decision | Activity Decision | Activity Decision digunakan untuk menggambarkan sebuah kegiatan yang memiliki pilihan lebih dari satu, biasanya dibutuhkan untuk mengambil sebuah keputusan. |
| 15. | Activity Start Event | Activity Start Event | Activity Start Event digunakan untuk memulai sebuah proses. |
| 16. | C:\Users\re_lust\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Activity Final Event.png | Activity Final Event | Activity Final State digunakan untuk mengakhiri sebuah proses. |

# **BAB I PENDAHULUAN**

## **Pendefinisian Lingkup**

Pembeli datang ke foodcourt untuk melakukan pembelian makanan. Proses pemesanan di foodcourt di mulai dari pembeli melakukan pemesanan di stand penjual yang dipilih, kemudian penjual mencatat dan memeriksa pesanan pembeli yang dipesan. Jika tersedia maka penjual akan melakukan pencetakan daftar pembelian sebanyak dua rangkap dengan sistem yang ada, jika tidak tersedia (pembeli tidak ingin memesan menu lain) maka pembeli meninggalkan stand dan tidak jadi memesan dan meninggalkan foodcourt. Daftar pembelian yang telah dicetak akan disimpan oleh penjual dan cetak lainnya diberikan kepada pembeli sebagai bukti pemesanan yang sah. Daftar pembelian yang diterima pembeli nantinya akan diberikan ke kasir untuk proses pembayaran. Kasir akan menginputkan kode pembelian yang tertulis di daftar pembelian dan aplikasi akan menampilkan daftar pesanan pembeli dan biaya yang harus dibayar oleh pembeli, lalu penjual menginformasikan biaya yang harus dibayar oleh pembeli dan pembeli memberikan uang pembayaran kepada kasir. Kasir menginputkan uang pembayaran yang diberikan oleh pembeli dan aplikasi mengkalkulasi uang pembayaran yang diinputkan dan menampilkan kembalian pembeli apabila nilai uang yang diberikan lebih besar dari biaya pembelian. Lalu kasir akan mencetak struk pembayaran melalui aplikasi yang nantinya diberikan kepada pembeli sebagai bukti pembayaran yang sah. Struk pembayaran yang diterima oleh pembeli nantinya diserahkan kepada stand penjual yang digunakan oleh penjual untuk memproses pembuatan pesanan yang sebelumnya telah dipesan oleh pembeli. Lingkup proses pemesanan yang dibahas hanya pesanan ditempat / makan ditempat, tidak membahas proses pemesanan secara *take away* atau dibawa pulang.

## **Pembagian Kerja Tim**

Tim terdiri dari 3 (tiga orang tenaga ahli), yaitu Sistem Analis, Desain Sistem / Programmer & Desain Prototype

Tabel 1.1 Tabel Pembagian Kerja Tim

|  |  |  |
| --- | --- | --- |
| **No.** | **Pembagian Kerja** | **Tenaga Ahli** |
| 1. | Team Leader / Sistem Analis | BBBBB |
| 2. | Desain Sistem | CCCCC |
| 3. | Desain Prototype | M. Rizky Saputra |

## **Penjelasan Proses Bisnis**

Sistem Informasi Penjualan mempunyai beberapa proses bisnis. Dalam pekerjaan pembangunan Sistem Informasi Penjualan, proses bisnis yang akan ditinjau adalah sebagai berikut:

Tabel 1.2 Tabel Penjelasan Proses Bisnis

|  |  |  |
| --- | --- | --- |
| **No.** | **Proses Bisnis** | **Deskripsi** |
| 1. | Pemesanan Makanan | Proses Bisnis ini adalah bagaimana proses memesan makanan. |
| 2 | Pembayaran Makanan | Proses Bisnis ini adalah bagaimana proses memesan makanan. |

# **BAB II PROSES BISNIS DAN REQUIREMENT**

## **PROSES BISNIS**

Pada bagian ini akan menjelaskan proses bisnis yang ada pada food court yang kami amati

### **Pendefinisian Model Bisnis**

Uraikan tujuan dari pendefinisian model bisnis ini

### **Business Use Case**

Pada bagian ini akan menjelaskan tentang Business Use Case yang ada di aplikasi SoftFood

|  |
| --- |
| **Business Use Case Diagram** |
| Gambar 2.1 Gambar Business Use Case Diagram |
| **Penjelasan** |
| Ini adalah Business Use Case yang ada di aplikasi SoftFood. Dalam Business Use Case ini dijelaskan bahwa seorang pembeli bisa melakukan Pemesanan dan juga pembayaran |

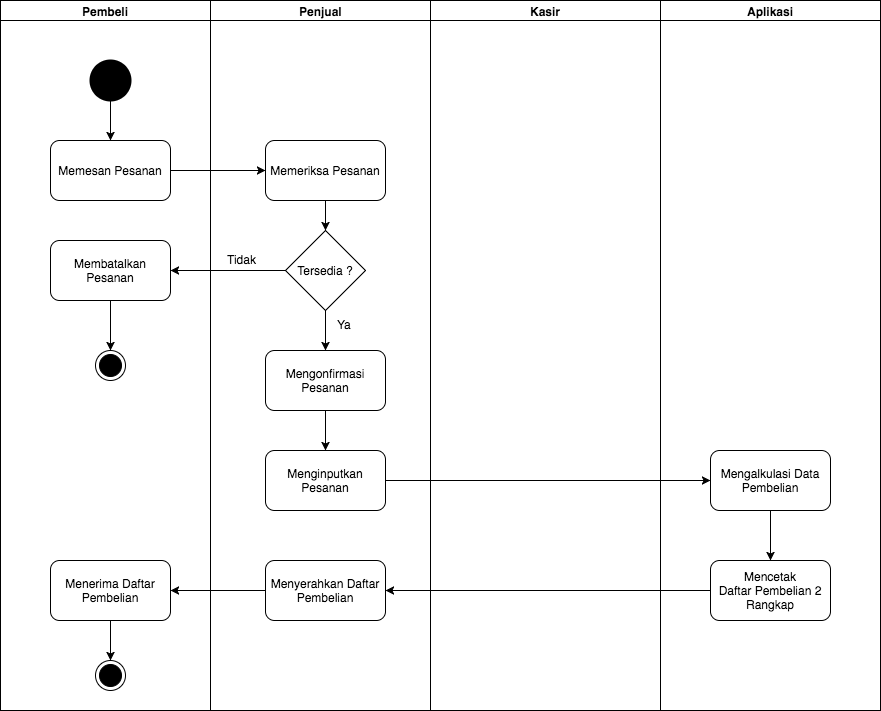
Tabel 2.1 Tabel Daftar Business Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| Daftar Business Use Case | | | |
| **No. BA** | **Business Actor** | **No. BC** | **Business Use Case** |
| BA-01 | Pembeli | BC-01 | Pemesanan Makanan |
| BC-02 | Pembayaran Makanan |

### **Aktivitas Business Use Case (Activity Diagram)**

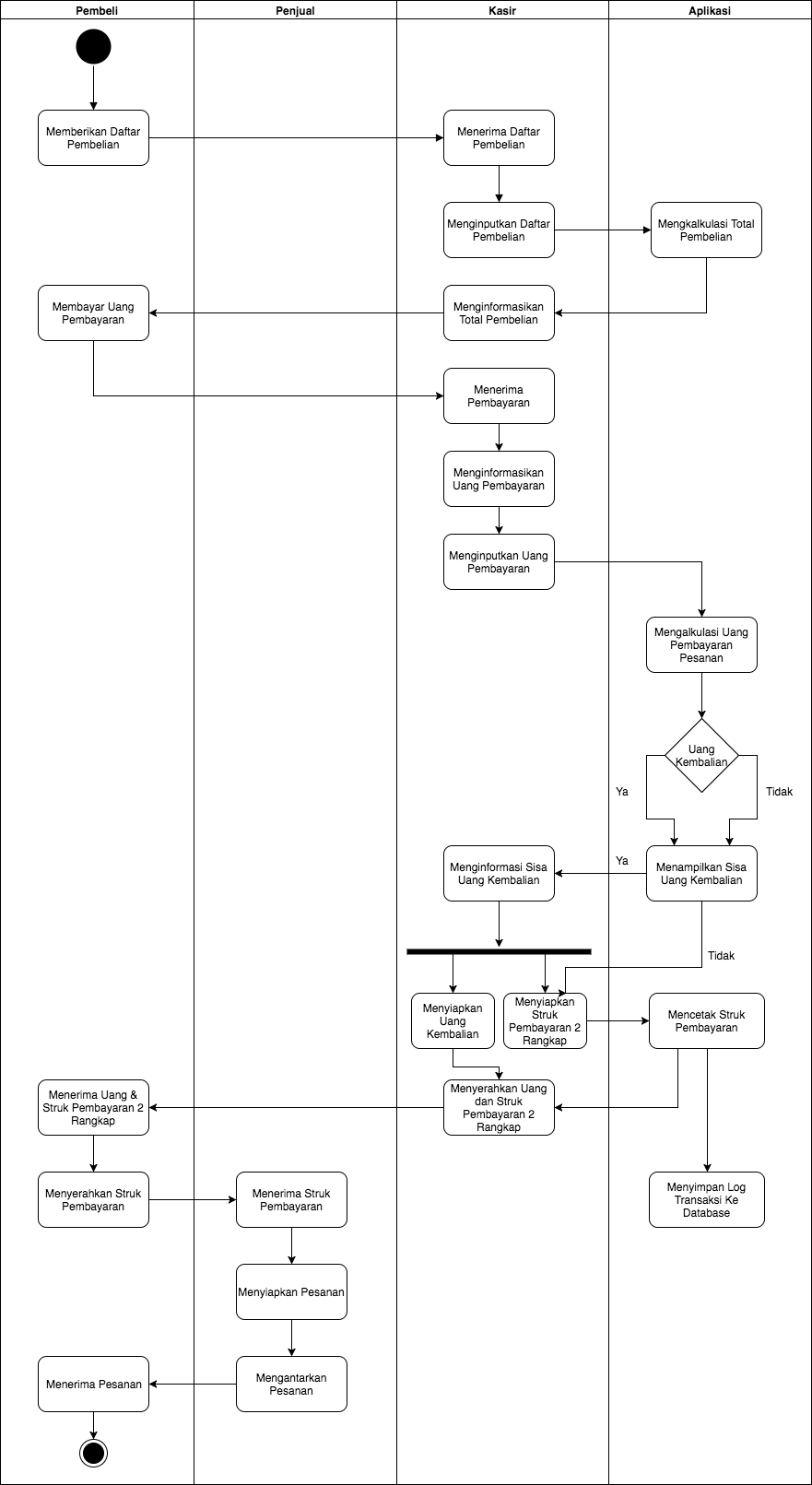
Dibawah ini adalah gambar dari diagram aktivitas yang terbagi menjadi dua bagian dan satu bagian gabungan

#### Activity Diagram – Pemesanan Makanan



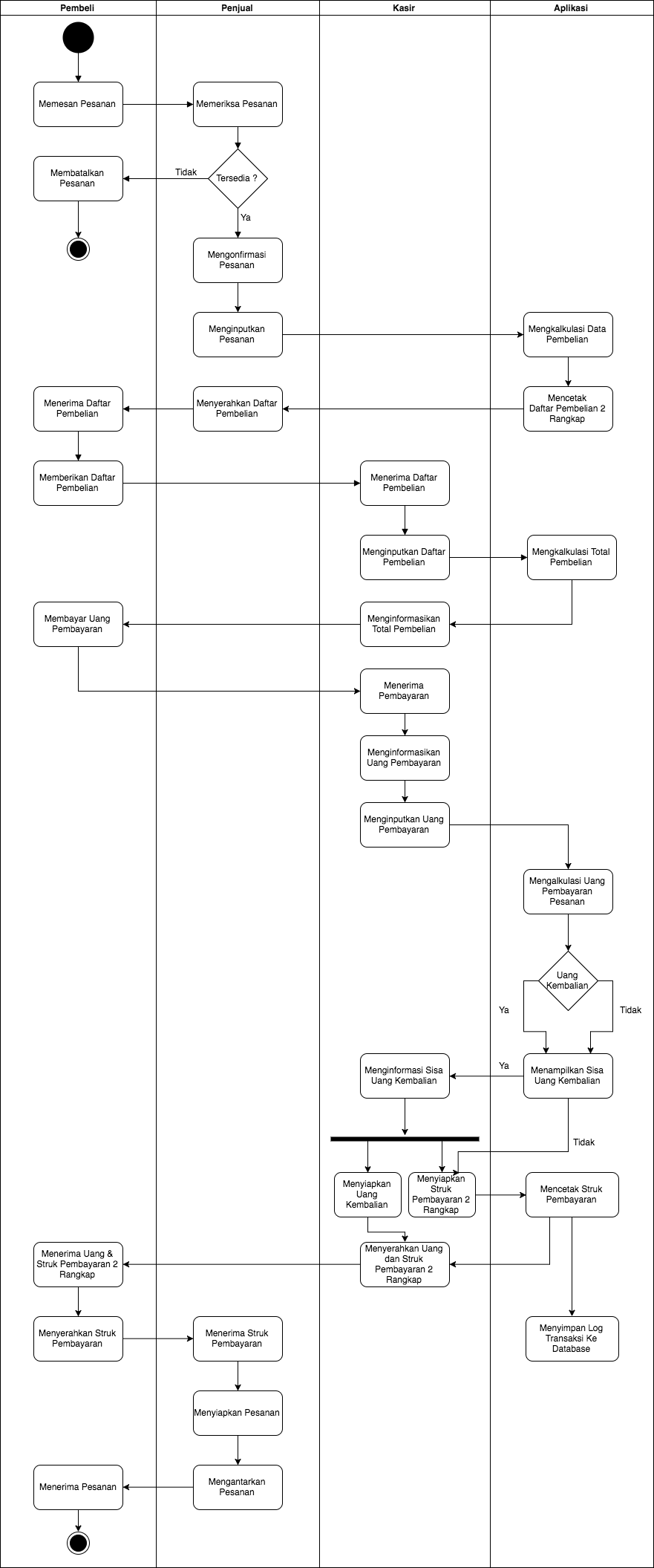
Gambar 2.2 Gambar Activity Diagram – Pemesanan Makanan

#### Activity Diagram – Pembayaran Makanan



Gambar 2.3 Activity Diagram – Pembayaran Makanan

#### Activity Diagram – Gabungan



Gambar 2.4 Activity Diagram – Gabungan

Tabel 2.2 Tabel Daftar Aktor / Worker Activity

|  |  |  |  |
| --- | --- | --- | --- |
| Daftar Actor / Worker dan Activity | | | |
| **No BA/BW** | **Business Actor / Worker** | **No A** | **Activity** |
| BA-01 | Pembeli | A-01 | Memesan Pesanan |
| BW-01 | Penjual | A-02 | Memeriksa Pesanan |
| BA-01 | Pembeli | A-03 | Membatalkan Pesanan |
| BW-01 | Penjual | A-04 | Mengonfirmasi Pesanan |
| A-05 | Menginputkan Pesanan |
| A-06 | Menyerahkan Daftar Pembelian |
| BA-01 | Pembeli | A-07 | Menerima Daftar Pembelian |
| A-08 | Memberikan Daftar Pembelian |
| BW-02 | Kasir | A-07 | Menerima Daftar Pembelian |
| A-09 | Menginputkan Daftar Pembelian |
| A-10 | Menginformasikan Total Pembelian |
| BA-01 | Pembeli | A-11 | Membayar Uang Pembayaran |
| BW-02 | Kasir | A-12 | Menerima Pembayaran |
| A-13 | Menginformasikan Uang Pembayaran |
| A-14 | Menginputkan Uang Pembayaran |
| A-15 | Menginformasikan Sisa Uang Pembayaran |
| A-16 | Menyiapkan Uang Kembalian |
| A-17 | Menyiapkan Struk Pembayaran 2 Rangkap |
| A-18 | Menyerahkan Uang dan Struk Pembayaran 2 Rangkap |
| BA-01 | Pembeli | A-19 | Menerima Uang dan Struk Pembayaran 2 Rangkap |
| A-20 | Menyerahkan Struk Pembayaran |
| BW-01 | Penjual | A-21 | Menerima Struk Pembayaran |
| A-22 | Menyiapkan Pesanan |
| A-23 | Mengantar Pesanan |
| BA-01 | Pembeli | A-24 | Menerima Pesanan |

### **Business Object Model**

Dibawah ini adalah gambar dari Business Object Model yang terbagi menjadi tiga bagian dan satu bagian gabungan

#### Business Object Model – Pemesanan Makanan

|  |
| --- |
| **Business Object Model Diagram – Pemesanan Makanan** |
| Gambar 2.5 Business Object Model – Pemesanan Makanan |
| **Penjelasan** |
| Business Object Model ini menjelaskan tentang penjual menawarkan Menu dan Daftar pembelian kepada pelanggan untuk melakukan transaksi |

#### Business Object Model – Pembayaran Makanan

|  |
| --- |
| **Business Object Model Diagram – Pembayaran Makanan** |
| Gambar 2.6 Business Object Model – Pembayaran Makanan |
| **Penjelasan** |
| Business Object Model ini menjelaskan tentang pembayaran makanan, yang mana kasir akan menerima data pembelian dan pembayaran melalui Daftar Pembeliandari Penjual dan Struk Pembayaran |

#### Business Object Model – Gabungan

|  |
| --- |
| **Business Object Model Diagram – Gabungan** |
| Gambar 2.7 Business Object Model – Gabungan |
| **Penjelasan** |
| Business Object Model ini adalah gabungan dari semua Business Object Model yang ada di atas. Business Object Model ini mencakup semua transaksi antar penjual dan kasir untuk melayani seorang pembeli. |

Tabel 2.3 Tabel Daftar Worker dan Object

|  |  |  |  |
| --- | --- | --- | --- |
| Daftar Worker dan Object | | | |
| **No BW** | **Business Worker** | **No O** | **Nama Object** |
| BW-01 | Penjual | O-01 | Menu |
| O-02 | Daftar Pembelian |
| BW-02 | Kasir | O-02 | Daftar Pembelian |
| O-03 | Struk Pembayaran |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Traceability | | | | | | | |
| **No BA/BW** | **Business Actor / Worker** | **No BC** | **Business Use-Case** | **No A** | **Activity** | **No O** | **Nama Object** |
| BA-01 | Pembeli | BC-01 | Pemesanan Makanan | A-01 | Memesan Pesanan | O-01 | Menu |
| BW-01 | Penjual | A-02 | Memeriksa Pesanan |
| BA-01 | Pembeli | A-03 | Membatalkan Pesanan | - | - |
| BW-01 | Penjual | A-04 | Mengonfirmasi Pesanan | O-02 | Daftar Pembelian |
| A-05 | Menginputkan Pesanan |
| A-06 | Menyerahkan Daftar Pembelian |
| BA-01 | Pembeli | A-07 | Menerima Daftar Pembelian |
| BC-02 | Pembayaran Makanan | A-08 | Memberikan Daftar Pembelian |
| BW-02 | Kasir | A-07 | Menerima Daftar Pembelian |
| A-09 | Menginputkan Daftar Pembelian |
| A-10 | Menginformasikan Total Pembelian | - | - |
| BA-01 | Pembeli | A-11 | Membayar Uang Pembayaran |
| BW-02 | Kasir | A-12 | Menerima Pembayaran |
| A-13 | Menginformasikan Uang Pembayaran |
| A-14 | Menginputkan Uang Pembayaran |
| A-15 | Menginformasikan Sisa Uang Pembayaran |
| A-16 | Menyiapkan Uang Kembalian |
| A-17 | Menyiapkan Struk Pembayaran 2 Rangkap | O-03 | Struk Pembayaran |
| A-18 | Menyerahkan Uang dan Struk Pembayaran 2 Rangkap |
| BA-01 | Pembeli | A-19 | Menerima Uang dan Struk Pembayaran 2 Rangkap |
| A-20 | Menyerahkan Struk Pembayaran |
| BW-01 | Penjual | A-21 | Menerima Struk Pembayaran |
| A-22 | Menyiapkan Pesanan | - | - |
| A-23 | Mengantar Pesanan |
| BA-01 | Pembeli | A-24 | Menerima Pesanan |

Tabel 2.4 Tabel Traceability

#### **Definisi Aktor**

Tabel 2.9 Tabel Definisi Aktor

|  |  |  |
| --- | --- | --- |
| **No. Aktor** | **Aktor** | **Deskripsi** |
| ACT-01 | Penjual | Aktor yang berperan sebagai penjual, dimana penjual akan mendata semua pesanan yang dipesan pelanggan |
| ACT-02 | Kasir | Aktor yang berperan sebagai kasir, dimana kasir akan menghitung total semua pembelian dan menangani pembayaran |

# **BAB III ANALIS DAN DESAIN**

## **ANALISIS SYSTEM**

## **SPECIFICATION REQUIREMENT SYSTEM**

Pada bagian ini akan dijelaskan tentang requirement yang ada pada perangkat lunak, termasuk User Requirement

### **User Requirement Perangkat Lunak**

Tabel 2.5 Tabel User Requirement

|  |  |
| --- | --- |
| **No Kebutuhan Fungsional** | **Deskripsi** |
| KF-001 | Sistem dapat melakukan mengelola pesanan |
| KF-002 | Sistem dapat menyediakan pembuatan nota |
| KF-003 | Sistem dapat memfasilitasi pengembalian barang |
| KF-004 | Sistem dapat menampilkan komplain pembeli |
| KF-005 | Sistem dapat menampilkan informasi stok barang tas kulit yang masuk(dari produksi) |
| KF-006 | Sistem dapat menampilkan informasi stok barang tas kulit yang keluar(dari penjualan) |
| KF-007 | Sistem dapat menambah informasi stok barang tas kulit |
| KF-008 | Sistem dapat mengurangi informasi stok barang tas kulit |
| KF-009 | Sistem dapat mengubah informasi stok barang tas kulit |
| KF-010 | Sistem dapat menghapus informasi stok barang tas kulit |
| KF-011 | Sistem dapat menampilkan inforamsi bahan baku tas kulit masuk |
| KF-012 | Sistem dapat menampilkan inforamsi bahan baku tas kulit keluar |
| KF-013 | Sistem dapat menambah informasi bahan baku |
| KF-014 | Sistem dapat mengurangi informasi bahan baku |
| KF-015 | Sistem dapat mengubah informasi bahan baku |
| KF-016 | Sistem dapat menghapus informasi bahan baku |
| KF-017 | Sistem dapat menampilkan data supplier |
| KF-018 | Sistem dapat menambah informasi data supplier |
| KF-019 | Sistem dapat mengurangi informasi data supplier |
| KF-020 | Sistem dapat mengubah informasi data supplier |
| KF-021 | Sistem dapat menghapus informasi data supplier |
| KF-022 | Sistem memfasilitasi fitur login |
| KF-023 | Sistem memfasilitasi fitur registerasi (oleh admin saja) |
| KF-024 | Sistem memfasilitasi fitur logout |

### **Daftar Kebutuhan Perangkat Lunak**

Tabel 2.6 Tabel Kebutuhan Perangkat Lunak Fungsional

|  |  |
| --- | --- |
| **No Kebutuhan Non Fungsional** | **Deskripsi** |
| SF-001 | Perangkat lunak harus dapat mengkalkulasi daftar pembelian |
| SF-002 | Perangkat lunak harus dapat mencetak daftar pembelian |
| SF-003 | Perangkat lunak harus dapat mengkalkulasi total pembelian |
| SF-004 | Perangkat lunak harus dapat mengkalkulasi transaksi pembayaran |
| SF-005 | Perangkat lunak harus dapat mencetak struk pembayaran |
| SF-006 | Perangkat lunak harus dapat mengelola daftar pembelian |
| SF-007 | Perangkat lunak harus dapat mengelola struk pembayaran |

Tabel 2.7 Tabel Kebutuhan Perangkat Lunak Non-Fungsional

|  |  |
| --- | --- |
| **No Kebutuhan Non- Fungsional** | **Deskripsi** |
| SNF-001 | Dapat digunakan pada perangkat non-mobile / pc |
| SNF-002 | Dapat digunakan pada perangkat yang menggunakan keyboard / numpad sebagai input |
| SNF-003 | Dapat digunakan pada perangkat berbasis Touch Screen |
| SNF-004 | Dapat mengkalkulasi total pembelian kurang dari 3 detik |
| SNF-005 | Memiliki warna untuk menandai pemesanan yang belum dibayar |

### **Model Use Case**

Model Use Case akan menjelaskan Diagram Use Case, Definisi Use Case dan Definisi aktor

#### **Diagram Use Case**

|  |
| --- |
| **Use Case Diagram SoftFood** |
| Gambar 2.8 Use Case Diagram – Activity Diagram |
| **Penjelasan** |
| Use case ini menjelaskan tentang penjual yang melakukan Kalkulasi Daftar Pembelian, Cetak Daftar Pembelian dan Mengelola Daftar Pembelian. Sementara Kasir akan melakukan Kalkulasi Total Pembelian, Kalkulasi Transaksi Pembayaran, Cetak Struk Pembayaran dan Mengelola Struk Pembelian |

#### **Definisi Use Case**

Tabel 2.8 Tabel Definisi Use-Case

| **No. Use Case** | **Use Case** | **Deskripsi** |
| --- | --- | --- |
| UC-01 | Kalkulasi Daftar Pembelian | Aktor menginputkan data pembelian pembeli ke aplikasi dan aplikas imengkalkulasi & menampilkan data daftar pembelian |
| UC-02 | Cetak Daftar Pembelian | Aktor mencetak daftar pembelian untuk diberikan kepada pembeli sebagai bukti pemesanan yang sah |
| UC-03 | Kelola Daftar Pembelian | Aktor menginputkan data pesanan pembeli ke aplikasi dan aplikasi mengelola data pemesanan yang nantinya dicetak menjadi daftar pembelian yang diserahkan ke pembeli sebagai bukti pemesanan yang sah |
| UC-04 | Kalkulasi Total Pembelian | Aktor menginputkan data pembelian pembeli ke aplikasi dan aplikasi mengkalkulasi & menampilkan total pembelian pesanan |
| UC-05 | Kalkulasi Transaksi Pembayaran | Aktor menginputkan uang pembayaran pesanan dari pembeli ke aplikasi dan aplikasi mengkalkulasi & menampilkan hasil kalkulasi pembayaran. |
| UC-06 | Cetak Struk Pembayaran | Aktor mencetak struk pembayaran untuk diberikan kepada pembeli sebagai bukti transaksi yang sah |
| UC-07 | Kelola Struk Pembayaran | Aktor menginputkan uang pembayaran pesanan dari pembeli ke aplikasi dan aplikasi mengkalkulasi & menampilkan hasil kalkulasi pembayaran yang dapat di cetak sebagai bukti pembayaran. |

* + - 1. ***Definisi Aktor***

***Tabel Definisi Aktor***

|  |  |  |
| --- | --- | --- |
| **No.Aktor** | **Nama Aktor** | **Deskripasi** |
| AC-01 | Penjual | ……………….. |
| AC-02 | Kasir | ……………………. |

#### **Skenario Use Case**

Dibawah ini akan dijelaskan mengenai scenario yang terdapat setiap use case pada aplikasi SoftFood

|  |  |  |
| --- | --- | --- |
| Identifikasi | | |
| Nomor | UC-01 | |
| Nama | Kalkulasi daftar pembelian | |
| Tujuan | Aplikasi dapat mengkalkulasi & menampilkan data pembelian yang diinputkan | |
| Deskripsi | Aktor menginputkan data pembelian pembeli ke aplikasi dan aplikas imengkalkulasi & menampilkan data daftar pembelian | |
| Aktor | Penjual | |
| Skenario Utama | | |
|  | | |
| Kondisi Awal : Halaman pemesanan tampil | | |
| Aksi Aktor | | Reaksi Sistem |
| 1. Menginputkan nama item & jumlah pesanan | |  |
| 2. Menekan tombol “Proses” | | 3. Menangkap nilai nama pemesanan dan jumlah pemesanan |
|  | | 4. Memanggil fungsi kalkulasiPemesanan |
|  | | 5. Memanggil fungsi tampilPesanan |
|  | | 6. Menampilkan data pemesanan yang  telah diinputkan |
| Kondisi Akhir : Daftar data pemesanan telah ditampilkan | | |
| Skenario Alternatif No.6: Menampilkan data pemesanan yang  telah diinputkan | | |

|  |  |  |
| --- | --- | --- |
| Identifikasi | | |
| Nomor | UC-02 | |
| Nama | Cetak daftar pembelian | |
| Tujuan | Aplikasi dapat mencetak daftar pembelian sebagai bukti pemesanan | |
| Deskripsi | Aktor mencetak daftar pembelian untuk diberikan kepada pembeli sebagai bukti pemesanan yang sah | |
| Aktor | Penjual | |
| Skenario Utama | | |
|  | | |
| Kondisi Awal : Halaman pemesanan sudah diisi | | |
| Aksi Aktor | | Reaksi Sistem |
| 1. Menekan tombol “Cetak Struk” | | 2. Menyimpan data pemesanan yang  diinputkan |
|  | | 3. Memanggil fungsi cetak |
|  | | 4. Memproses cetak daftar pembelian |
| Kondisi Akhir : Daftar pembelian telah dicetak | | |
| Skenario Alternatif No.4: Gagal cetak daftar pembelian | | |

|  |  |  |
| --- | --- | --- |
| Identifikasi | | |
| Nomor | UC-03 | |
| Nama | Kelola daftar pembelian | |
| Tujuan | Aplikasi dapat mengelola struk pembelian yang diinputkan aktor | |
| Deskripsi | Aktor menginputkan data pesanan pembeli ke aplikasi dan aplikasi mengelola data pemesanan yang nantinya dicetak menjadi daftar pembelian yang diserahkan ke pembeli sebagai bukti pemesanan yang sah | |
| Aktor | Penjual | |
| Isi Use-Case | 1. Create / Read 2. Update 3. Delete 4. Print | |
| Skenario Utama | | |
| 1. **Create / Read** | | |
| Kondisi Awal : Halaman pemesanan sudah dibuka | | |
| Aksi Aktor | | Reaksi Sistem |
| 1. Menginputkan nama pemesanan & jumlah pemesanan | |  |
| 2. Menekan tombol “proses” | | 3. Menangkap nilai nama pemesanan dan jumlah pemesanan |
|  | | 4. Memanggil fungsi kalkulasiPemesanan |
|  | | 5. Memanggil fungsi tampilPemesanan |
|  | | 6. Menampilkan data pemesanan yang telah diinputkan |
| Kondisi Akhir: Daftar data pemesanan ditampilkan | | |
| Skenario Alternatif No.6: Pesan error menampilkan data pemesanan yang telah diinputkan | | |

|  |  |
| --- | --- |
| **2. Update** | |
| Kondisi Awal : Daftar data pemesanan telah ditampilkan | |
| Aksi Aktor | Reaksi Sistem |
| 1. Menekan tombol “Edit Pesanan” | 2. Menampilkan halaman edit data pemesanan |
| 3. Menekan tombol “Update” |  |
| 4. Mengganti data pemesanan sesuai keinginan pembeli di tabel pemesanan |  |
| 5. Menekan tombol “Simpan” | 6. Menangkap nilai nama pemesanan dan jumlah pemesanan |
|  | 7. Memanggil fungsi kalkulasiPemesanan |
|  | 8. Memanggil fungsi tampilPesanan |
|  | 9. Menampilkan data edit terbaru di halaman pemesanan |
| Kondisi Akhir : Menampilkan data pemesanan pada halaman pemesanan yang sudah diubah | |
| Skenario Alternatif N0.9: Menampilkan pesan error | |

|  |  |
| --- | --- |
| **3. Delete** | |
| Kondisi Awal : Halaman edit pemesanan sudah ditampilkan | |
| Aksi Aktor | Reaksi Sistem |
| 1. Memilih data pemesanan yang ingin dihapus |  |
| 2. Menekan tombol “Delete” | 3. Menangkap nilai kode pesanan |
|  | 4. Memanggil fungsi delete |
|  | 5. Menghapus data pemesanan yang  dipilih |
| Kondisi Akhir : Menampilkan halaman pemesanan yang sudah dihapus | |
| Skenario Alternatif No.5: Menampilkan pesan error | |

|  |  |
| --- | --- |
| **4. Print** | |
| Kondisi Awal : Halaman pemesanan sudah diisi | |
| Aksi Aktor | Reaksi Sistem |
| 1. Menekan tombol “cetak struk” | 2. Memanggil fungsi print |
|  | 3. Menyimpan data transaksi yang diinputkan |
|  | 4. Memproses cetak daftar pembelian |
| Kondisi Akhir : Daftar pembelian telah dicetak | |
| Skenario Alternatif No.4: Menampilkan pesan error | |

|  |  |  |
| --- | --- | --- |
| Identifikasi | | |
| Nomor | UC-04 | |
| Nama | Kalkulasi total pembelian | |
| Tujuan | Aktor dapat melihat total biaya pembelian pesanan pembeli | |
| Deskripsi | Aktor menginputkan data pembelian pembeli ke aplikasi dan aplikasi mengkalkulasi & menampilkan total pembelian pesanan | |
| Aktor | Kasir | |
| Skenario Utama | | |
| Kondisi Awal : Halaman pembelian sudah dibuka | | |
| Aksi Aktor | | Reaksi Sistem |
| 1. Menginputkan kode pesanan | |  |
| 2. Menekan tombol “proses” | | 3. Menangkap nilai kode pesanan |
|  | | 4. Memanggil fungsi kalkulasiPembelian |
|  | | 5. Memanggil fungsi tampilPesanan |
|  | | 6. Menampilkan total pembelian harga |
| Kondisi Akhir : Menampilkan seluruh data & biaya pembelian pesanan yang diinputkan | | |
| Skenario Alternatif No.6: Menampilkan pesan error | | |

|  |  |  |
| --- | --- | --- |
| Identifikasi | | |
| Nomor | UC-05 | |
| Nama | Kalkulasi transaksi pembayaran | |
| Tujuan | Aktor dapat melihat total pembayaran yang harus dibayar pembeli | |
| Deskripsi | Aktor menginputkan uang pembayaran pesanan dari pembeli ke aplikasi dan aplikasi mengkalkulasi & menampilkan hasil kalkulasi pembayaran. | |
| Aktor | Kasir | |
| Skenario Utama | | |
| Kondisi Awal : Halaman pembayaran sudah dibuka | | |
| Aksi Aktor | | Reaksi Sistem |
| 1. Menginputkan uang pembayaran | |  |
| 2. Menekan tombol “OK” | | 3. Menangkap nilai pembayaran |
|  | | 4. Mengkalkulasi uang pembayaran yang telah diinputkan dengan fungsi kalkulasiPembayaran |
| Kondisi Akhir : Menampilkan uang pembayaran & kembalian pembayaran | | |
| Skenario Alternatif No.4: Menampilkan pesan error | | |

|  |  |  |
| --- | --- | --- |
| Identifikasi | | |
| Nomor | UC-06 | |
| Nama | Cetak struk pembayaran | |
| Tujuan | Aktor dapat mencetak struk pembayaran sebagai bukti transaksi | |
| Deskripsi | Aktor mencetak struk pembayaran untuk diberikan kepada pembeli sebagai bukti transaksi yang sah | |
| Aktor | Kasir | |
| Skenario Utama | | |
| Kondisi Awal : Halaman menampilkan pop-up cetak struk pembayaran | | |
| Aksi Aktor | | Reaksi Sistem |
| 1. Menekan tanda “Ya” di pop-up cetak struk pembayaran | | 2. Menyimpan data pembayaran yang diinputkan |
|  | | 3. Memproses cetak struk pembayaran |
|  | | 4. Melakukan cetak struk |
| Kondisi Akhir : Struk pembayaran telah dicetak | | |
| Skenario Alternatif No.4: Menampilkan pesan error | | |

|  |  |  |
| --- | --- | --- |
| Identifikasi | | |
| Nomor | UC-07 | |
| Nama | Kelola struk pembayaran | |
| Tujuan | Aplikasi dapat mengelola struk pembayaran yang diinputkan aktor | |
| Deskripsi | Aktor menginputkan uang pembayaran pesanan dari pembeli ke aplikasi dan aplikasi mengkalkulasi & menampilkan hasil kalkulasi pembayaran yang dapat di cetak sebagai bukti pembayaran. | |
| Aktor | Kasir | |
| Isi Use-Case | 1. Create / Read  2. Update  3. Print | |
| Skenario Utama | | |
| **1. Create / Read** | | |
| Kondisi Awal : Halaman pembelian sudah dibuka | | |
| Aksi Aktor | | Reaksi Sistem |
| 1. Menginputkan kode pesanan | |  |
| 2. Menekan tombol “proses” | | 3. Menangkap nilai kode pesanan |
|  | | 4. Memanggil fungsi kalkulasiPembelian |
|  | | 5. Memanggil fungsi tampilPesanan |
|  | | 6. Menampilkan data pemesanan yang  telah diinputkan |
| 7. Menekan tombol “Pembayaran” | | 8. Menampilkan halaman pembayaran |
| 9. Menginputkan uang pembayaran | |  |
| 10. Menekan tombol “OK” | | 11. Menangkap nilai pembayaran |
|  | | 12. Mengkalkulasi uang pembayaran yang telah diinputkan dengan fungsi kalkulasiPembayaran |
|  | | 13. Menampilkan hasil kalkulasi |
| Kondisi Akhir : Menampilkan uang pembayaran & kembalian pembayaran | | |
| Skenario Alternatif No.13: Menampilkan pesan error | | |

|  |  |
| --- | --- |
| **2. Update** | |
| Kondisi Awal : Daftar transaksi pembayaran telah ditampilkan | |
| Aksi Aktor | Reaksi Sistem |
| 1. Mengganti inputan uang pembayaran |  |
| 1. Menekan tombol OK | 3, Menangkap nilai pembayaran |
|  | 4. Mengkalkulasi uang pembayaran yang telah diinputkan dengan fungsi kalkulasiPembayaran |
|  | 1. Menampilkan hasil kalkulasi baru |
| Kondisi Akhir : Menampilkan halaman pembayaran yang telah diubah | |
| Skenario Alternatif No.5: Menampilkan pesan error | |

|  |  |
| --- | --- |
| **3. Print** | |
| Kondisi Awal : Halaman menampilkan pop-up cetak struk pembayaran | |
| Aksi Aktor | Reaksi Sistem |
| 1. Menekan tanda “Ya” di pop-up cetak struk pembayaran | 2. Memanggil fungsi print |
|  | 3. Menyimpan data transaksi yang diinputkan |
|  | 4. Memproses cetak struk pembayaran |
|  | 5. Menyimpan log transaksi dengan fungsi saveToLog |
| Kondisi Akhir : Struk pembayaran telah dicetak | |
| Skenario Alternatif No.5: Menampilkan pesan error | |

#### **Diagram Sekuen (Sequence Diagram)**

|  |
| --- |
| **Sequence Diagram** |
| Gambar 3.40 Sequence Diagram – Kalkulasi Daftar Pembelian |
| **Penjelasan** |
| Sequence Diagram ini menjelaskan bagaimana penjual melakukan Kalkulasi Daftar Pembelian dengan memberikan input sampai halaman pemesanan menerima |

|  |
| --- |
| **Sequence Diagram** |
| Gambar 3.41 Sequence Diagram Cetak Daftar Pembelian |
| **Penjelasan** |
| Sequence Diagram ini menjelaskan bagaimana penjual melakukan Cetak daftar pembelian dari penjual menekan tombol cetak sampai print sukses dan keluar struk |

|  |
| --- |
| **Sequence Diagram** |
| Gambar 3.42 Sequence Diagram – Mengelola Daftar Pembelian – Create / Read |
| **Penjelasan** |
| Sequence Diagram ini menjelaskan tentang penjual yang mengelola Daftar Pembelian dan melakukan create / read data, sampai show data pesanan |

|  |
| --- |
| **Sequence Diagram** |
| Gambar 3.43 Sequence Diagram – Mengelola Daftar Pembelian - Update |
| **Penjelasan** |
| Sequence Diagram ini akan menjelaskan seorang penjual meng-update atau mengubah data yang sudah ada sesuai permintaan pelanggan sampai data ditampilkan |

|  |
| --- |
| **Sequence Diagram** |
| Gambar 3.44 Sequence Diagram – Mengelola Daftar Pembelian – Delete |
| **Penjelasan** |
| Sequence Diagram ini akan menjelaskan bagaimana penjual menghapus data yang sudah ada sampai saat penampilan data, data tersebut sudah tidak ada |

|  |
| --- |
| **Sequence Diagram** |
| Gambar 3.45 Sequence Diagram – Mengelola Daftar Pembelian – Print |
| **Penjelasan** |
| Sequence Diagram ini akan menjelaskan bagaimana penjual melakukan cetak daftar pesanan yang sudah dipesan, yang nantinya akan diteruskan kepada kasir untuk melakukan pembayaran |

|  |
| --- |
| **Sequence Diagram** |
| Gambar 3.46 Sequence Diagram Kalkulasi Total Pembelian |
| **Penjelasan** |
| Sequence Diagram ini akan menjelaskan seorang Kasir yang melakukan Kalkulasi Total Pembelian, dari inputKodePesanan sampai menampilkan pesanan yang diinput. |

|  |
| --- |
| **Sequence Diagram** |
| Gambar 3.47 Sequence Diagram Kalkulasi Transaksi Pembayaran |
| **Penjelasan** |
| Sequence Diagram ini akan menjelaskan tentang kasir yang melakukan Kalkulasi Transaksi Pembayaran yang mana kasir akan melakukan input pembayaran sampai menampilkan pembayaran & kembalian |

|  |
| --- |
| **Sequence Diagram** |
| Gambar 3.48 Sequence Diagram - Cetak Struk Pembayaran |
| **Penjelasan** |
| Sequence Diagram ini akan menjelaskan tentang kasir yang melakukan Cetak Struk Pembayaran yang mana kasir menekan tombol ya dan keluar hasil print tersebut |

|  |
| --- |
| **Sequence Diagram** |
| Gambar 3.49 Sequence Diagram – Mengelola Daftar Pembayaran – Create / Read |
| **Penjelasan** |
| Sequence Diagram ini akan menjelaskan seorang kasir yang sedang melakukan create data dari kode pesanan yang diberikan pelanggan. Kode tersebut akan otomatis generate semua hal yang dipesan sebelumnya |

|  |
| --- |
| **Sequence Diagram** |
| Gambar 3.50 Sequence Diagram – Mengelola Daftar Pembayaran - Update |
| **Penjelasan** |
| Sequence Diagram ini akan menjelaskan bagaimana kasir mengubah data yang sudah ada. |

|  |
| --- |
| **Sequence Diagram** |
| Gambar 3.51 Sequence Diagram – Mengelola Daftar Pembayaran – Print |
| **Penjelasan** |
| Sequence Diagram ini akan menjelaskan seorang kasir yang akan mencetak semua pembayaran, dimana pembayaran akan otomatis tersimpan di database dan mengeluarkan hasil cetakan tersebut untuk dijadikan bukti transaksi |

#### **Diagram Kelas Analisis**

|  |
| --- |
| **Diagram Kelas Analisis** |
| Gambar 3.52 Kalkulasi Daftar Pembeli |
| **Penjelasan** |
| Diagram Kelas Analis ini akan menjelaskan Penjual melakukan Kalkulasi Daftar Pembelian |

|  |
| --- |
| **Diagram Kelas Analisis** |
| Gambar 3.53 Cetak Daftar Pembeli |
| **Penjelasan** |
| Diagram Kelas Analis ini akan menjelaskan Penjual melakukan Cetak Daftar Pembelian |

|  |
| --- |
| **Diagram Kelas Analisis** |
| Gambar 3.54 Mengelola Daftar Pembelian – Create/Read |
| **Penjelasan** |
| Diagram Kelas Analis ini akan menjelaskan Penjual melakukan Kelola Daftar Pembelian – Create / Read |

|  |
| --- |
| **Diagram Kelas Analisis** |
| Gambar 3.55 Mengelola Daftar Pembelian – Update |
| **Penjelasan** |
| Diagram Kelas Analis ini akan menjelaskan Penjual melakukan Kelola Daftar Pembelian – Update |

|  |
| --- |
| **Diagram Kelas Analisis** |
| Gambar 3.56 Mengelola Daftar Pembelian – Delete |
| **Penjelasan** |
| Diagram Kelas Analis ini akan menjelaskan Penjual melakukan Kelola Daftar Pembelian – Delete |

|  |
| --- |
| **Diagram Kelas Analisis** |
| Gambar 3.57 Mengelola Daftar Pembelian – Print |
| **Penjelasan** |
| Diagram Kelas Analis ini akan menjelaskan Penjual melakukan Kelola Daftar Pembelian – Print |

|  |
| --- |
| **Diagram Kelas Analisis** |
| Gambar 3.58 Kalkulasi Total Pembelian |
| **Penjelasan** |
| Diagram Kelas Analis ini akan menjelaskan Kasir melakukan Kalkulasi Total Pembelian |

|  |
| --- |
| **Diagram Kelas Analisis** |
| Gambar 3.59 Kalkulasi Transaksi Pembayaran |
| **Penjelasan** |
| Diagram Kelas Analis ini akan menjelaskan Kasir melakukan Kalkulasi Transaksi Pembayaran |

|  |
| --- |
| **Diagram Kelas Analisis** |
| Gambar 3.60 Cetak Struk Pembayaran |
| **Penjelasan** |
| Diagram Kelas Analis ini akan menjelaskan Kasir melakukan Cetak Struk Pembayaran |

|  |
| --- |
| **Diagram Kelas Analisis** |
| Gambar 3.61 Mengelola Daftar Pembayaran – Create/Read |
| **Penjelasan** |
| Diagram Kelas Analis ini akan menjelaskan Kasir melakukan Kelola Daftar Pembayaran – Create / Read |

|  |
| --- |
| **Diagram Kelas Analisis** |
| Gambar 3.62 Mengelola Daftar Pembayaran – Update |
| **Penjelasan** |
| Diagram Kelas Analis ini akan menjelaskan Kasir melakukan Kelola Daftar Pembayaran – Update |

|  |
| --- |
| **Diagram Kelas Analisis** |
| Gambar 3.63 Mengelola Daftar Pembayaran – Print |
| **Penjelasan** |
| Diagram Kelas Analis ini akan menjelaskan Kasir melakukan Kelola Daftar Pembayaran – Print |

|  |
| --- |
| **Diagram Kelas Analisis** |
| Gambar 3.64 Diagram Kelas Analisis Gabungan – Penjual |
| **Penjelasan** |
| Diagram Kelas Analis ini akan menjelaskan semua aktivitas yang dilakukan oleh Penjual |

|  |
| --- |
| **Diagram Kelas Analisis** |
| Gambar 3.65 Diagram Kelas Analisis Gabungan – Kasir |
| **Penjelasan** |
| Diagram Kelas Analis ini akan menjelaskan semua aktivitas yang dilakukan oleh Kasir |

#### **Deskripsi Kelas Analisis**

Pada bagian ini akan menjelaskan tentang semua Kelas analisis berdasarkan Diagram Kelas Analisis.

Tabel 3.1 Tabel Diagram Kelas Analisis – Kalkulasi Daftar Pembelian

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Nama Kelas** | **Jenis Kelas** | **Daftar Tanggung Jawab** | **Daftar Atribut** |
| 1 | Halaman Pemesanan | Boundary | * inputNamaJumlahPesanan() * pressBtnProses() * showDataPesanan() | * textFieldNamaItem * textFieldJumlahItem * buttonProses * buttonCetakStruk * buttonEditPesanan * tabelPemesanan |
| 2 | Controller Pemesanan | Controller | * getInput() * tampilPesanan() | * Model Pemesanan |
| 3 | Model Pemesanan | Model/Entity | * kalkulasiPemesanan() | * Kode Pesanan * Nama Pesanan * Jumlah * Harga * Total Pembelian * Kode Pembelian |

Tabel 3.2 Tabel Diagram Kelas Analisis – Cetak Daftar Pembelian

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Nama Kelas** | **Jenis Kelas** | **Daftar Tanggung Jawab** | **Daftar Atribut** |
| 1 | Halaman Pemesanan | Boundary | * pressBtnCetak() * printSuccess() | * textFieldNamaItem * textFieldJumlahItem * buttonProses * buttonCetakStruk * buttonEditPesanan * tabelTotalPembelian * tabelPemesanan |
| 2 | Controller Pemesanan | Controller | * getPesanan() * doPrint(pesanan) | * Model Pemesanan |
| 3 | Model Pemesanan | Model/Entity | * setPesanan() * doSave(pesanan) | * Kode Pesanan * Nama Pesanan * Jumlah * Harga * Total Pembelian * Kode Pembelian |

Tabel 3.3 Tabel Diagram Kelas Analisis – Kelola Daftar Pembelian – Create/Read

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Nama Kelas** | **Jenis Kelas** | **Daftar Tanggung Jawab** | **Daftar Atribut** |
| 1 | Halaman Pemesanan | Boundary | * inputNamaJumlahPesanan() * pressBtnProses() * showDataPesanan() | * textFieldNamaItem * textFieldJumlahItem * buttonProses * buttonCetakStruk * buttonEditPesanan * tabelPemesanan |
| 2 | Controller Pemesanan | Controller | * getInput() * tampilPesanan(pesanan) | * Model Pemesanan |
| 3 | Model Pemesanan | Model/Entity | * kalkulasiPemesanan() | * Kode Pesanan * Nama Pesanan * Jumlah * Harga * Total Pembelian * Kode Pembelian |

Tabel 3.4 Tabel Diagram Kelas Analisis – Kelola Daftar Pembelian – Update

| **No.** | **Nama Kelas** | **Jenis Kelas** | **Daftar Tanggung Jawab** | **Daftar Atribut** |
| --- | --- | --- | --- | --- |
| 1 | Halaman Pemesanan | Boundary | * pressBtnEditPesanan() * showDataBiayaPesanan() | * textFieldNamaItem * textFieldJumlahItem * buttonProses * buttonCetakStruk * buttonEditPesanan * tabelTotalPembelian * tabelPemesanan |
| 2 | Halaman Edit Pemesanan | Boundary | * editPesananForm() * pressBtnUpdate() * changeInputPesanan() * pressBtnSimpan() | * textFieldNamaItem * textFieldKodeItem * textFieldJumlahItem * buttonProses * buttonSimpan * tabelPemesanan * ButtonUpdate * ButtonDelete |
| 3 | Controller Pemesanan | Controller | * editForm() * getInput() * tampilPesanan(pesanan) | * Model Pemesanan |
| 4 | Model Pemesanan | Model/Entity | * update() * doUpdate(nama, jumlah) * kalkulasiPemesanan(nama, jumlah) | * Kode Pesanan * Nama Pesanan * Jumlah * Harga * Total Pembelian * Kode Pembelian |

Tabel 3.5 Tabel Diagram Kelas Analisis – Kelola Daftar Pembelian – Delete

| **No.** | **Nama Kelas** | **Jenis Kelas** | **Daftar Tanggung Jawab** | **Daftar Atribut** |
| --- | --- | --- | --- | --- |
| 1 | Halaman Edit Pemesanan | Boundary | * selectData() * pressBtnDelete() | * textFieldNamaItem * textFieldKodeItem * textFieldJumlahItem * buttonProses * buttonSimpan * tabelPemesanan * buttonUpdate * buttonDelete |
| 2 | Halaman Pemesanan | Boundary | * showDataBiayaPesanan() | * textFieldNamaItem * textFieldJumlahItem * buttonProses * buttonCetakStruk * buttonEditPesanan * tabelTotalPembelian * tabelPemesanan |
| 3 | Controller Pemesanan | Controller | * getKode() * tampilPesanan(pesanan) | * Model Pemesanan |
| 4 | Model Pemesanan | Model/Entity | * delete() * doDelete(kode) | * Kode Pesanan * Nama Pesanan * Jumlah * Harga * Total Pembelian * Kode Pembelian |

Tabel 3.6 Tabel Diagram Kelas Analisis – Kelola Daftar Pembelian – Print

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Nama Kelas** | **Jenis Kelas** | **Daftar Tanggung Jawab** | **Daftar Atribut** |
| 1 | Halaman Pemesanan | Boundary | * pressBtnCetak() * printSuccess() | * textFieldNamaItem * textFieldJumlahItem * buttonProses * buttonCetakStruk * buttonEditPesanan * tabelTotalPembelian * tabelPemesanan |
| 2 | Controller Pemesanan | Controller | * getPesanan() * doPrint(pesanan) | * Model Pemesanan |
| 3 | Model Pemesanan | Model/Entity | * setPesanan() * doSave(pesanan) | * Kode Pesanan * Nama Pesanan * Jumlah * Harga * Total Pembelian * Kode Pembelian |

Tabel 3.7 Tabel Diagram Kelas Analisis – Kalkulasi Total Pembelian

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Nama Kelas** | **Jenis Kelas** | **Daftar Tanggung Jawab** | **Daftar Atribut** |
| 1 | Halaman Pembelian | Boundary | * inputKodePesanan() * presssBtnProses() * showDataPesanan() | * textFieldKodeItem * buttonProses * buttonPembayaran * buttonBatalPesanan * tabelPembelian |
| 2 | Controller Pembayaran | Controller | * getInput() * tampilPesanan(pesanan) | * Model Pemesanan |
| 3 | Model Pemesanan | Model/Entity | * kalkulasiPembelian(pesanan) | * Kode Pesanan * Nama Pesanan * Jumlah * Harga * Total Pembelian * Kode Pembelian |

Tabel 3.8 Tabel Diagram Kelas Analisis – Kalkulasi Transaksi Pembayaran

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Nama Kelas** | **Jenis Kelas** | **Daftar Tanggung Jawab** | **Daftar Atribut** |
| 1 | Halaman Pembayaran | Boundary | * inputPembayaran() * pressBtnOK() * showPembayaranKembalian() | * textFieldBayar * buttonOK * buttonEditPembayaran * tabelPembelian * tabelTransaksi * buttonCetakStruk |
| 2 | Controller Pembayaran | Controller | * getInput() * tampilTransaksi(pembayaran, kembalian) | * Model Pembayaran |
| 3 | Model Pembayaran | Model/Entity | * kalkulasiPembayaran(pembelian, pembayaran) | * Kode Pembelian * Uang Pembayaran * Tanggal Pembelian |

Tabel 3.9 Tabel Diagram Kelas Analisis – Cetak Struk Pembayaran

| **No.** | **Nama Kelas** | **Jenis Kelas** | **Daftar Tanggung Jawab** | **Daftar Atribut** |
| --- | --- | --- | --- | --- |
| 1 | Halaman Pemesanan | Boundary | * pressBtnCetak() * printSuccess() | * textFieldNamaItem * textFieldJumlahItem * buttonProses * buttonCetakStruk * buttonEditPesanan * tabelTotalPembelian * tabelPemesanan |
| 2 | Controller Pemesanan | Controller | * getPesanan() * doPrint(pesanan) | * Model Pemesanan |
| 3 | Model Pemesanan | Model/Entity | * setPesanan() * doSave(pesanan) | * Kode Pesanan * Nama Pesanan * Jumlah * Harga * Total Pembelian * Kode Pembelian |

Tabel 3.10 Tabel Diagram Kelas Analisis – Kelola Daftar Pembayaran – Create / Read

| **No.** | **Nama Kelas** | **Jenis Kelas** | **Daftar Tanggung Jawab** | **Daftar Atribut** |
| --- | --- | --- | --- | --- |
| 1 | Halaman Pembelian | Boundary | * inputKodePesanan() * pressBtnProses() * showDataPesanan() * pressBtnPembayaran() | * textFieldKodeItem * buttonProses * buttonPembayaran * buttonBatalPesanan * tabelPembelian |
| 2 | Halaman Pembayaran | Boundary | * inputUangPembayaran() * pressBtnOK() * showPembayaranForm() * showUangPembayaranKembalian() | * textFieldbayar * buttonOK * buttonEditPembayaran * tabelPembelian * tabelTransaksi * buttonCetakStruk |
| 3 | Controller Pemesanan | Controller | * getInput() * tampilPesanan(pesanan) | * Model Pemesanan |
| 4 | Controller Pembayaran | Controller | * getInput() * showPembayaranForm() * tampilPembayaran(pembayaran) | * Model Pembayaran |
| 5 | Model Pemesanan | Model/Entity | * kalkulasiPembelian(input) | * Kode Pesanan * Nama Pesanan * Jumlah * Harga * Total Pembelian * Kode Pembelian |
| 6 | Model Pembayaran | Model/Entity | * kalkulasiPembayaran(input) | * Kode Pembelian * Uang Pembayaran * Tanggal Pembelian |

Tabel 3.11 Tabel Diagram Kelas Analisis – Kelola Daftar Pembayaran – Update

| **No.** | **Nama Kelas** | **Jenis Kelas** | **Daftar Tanggung Jawab** | **Daftar Atribut** |
| --- | --- | --- | --- | --- |
| 1 | Halaman Pembayaran | Boundary | * changeInputUangPembayaran() * pressBtnOK() * pembayaranEdit() | * textFieldbayar * buttonOK * buttonEditPembayaran * tabelPembelian * tabelTransaksi * buttonCetakStruk |
| 2 | Controller Pembayaran | Controller | * getInput() * tampilPembayaran(pembayaran) | * Model Pembayaran |
| 3 | Model Pembayaran | Model/Entity | * kalkulasiPembayaran(input) | * Kode Pembelian * Uang Pembayaran * Tanggal Pembelian |

Tabel 3.12 Tabel Diagram Kelas Analisis – Kelola Daftar Pembayaran – Print

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Nama Kelas** | **Jenis Kelas** | **Daftar Tanggung Jawab** | **Daftar Atribut** |
| 1 | Halaman Pembayaran | Boundary | * pressBtnYA() * printSuccess() | * buttonYa * buttonTidak * buttonCetakStruk |
| 2 | Controller Pembayaran | Controller | * savePembayaran() * doPrint(pembayaran) | * Model Pembayaran |
| 3 | Model Pembayaran | Model/Entity | * savePembayaran() * doSave(pembayaran) * saveToLog() | * Kode Pembelian * Uang Pembayaran * Tanggal Pembelian |

Tabel 3.13 Tabel Diagram Kelas Analisis – Kelola Daftar Pembayaran – Gabungan

| **No.** | **Nama Kelas** | **Jenis Kelas** | **Daftar Tanggung Jawab** | **Daftar Atribut** |
| --- | --- | --- | --- | --- |
| 1 | Halaman Pemesanan | Boundary | * inputNamaJumlahesanan() * pressBtnProses() * showDataPesanan() * pressBtnCetak() * printSuccess() * pressBtnEditPesanan() * showDataBiayaPesanan() | * textFieldNamaItem * textFieldJumlahItem * buttonProses * buttonCetakStruk * buttonEditPesanan * tabelPemesanan |
| 2 | Halaman Edit Pemesanan | Boundary | * editPesananForm() * pressBtnUpdate() * changeInputPesanan() * pressBtnSimpan() * selectData() * pressBtnDelete() | * textFieldNamaItem * textFieldKodeItem * textFieldJumlahItem * buttonProses * buttonSimpan * tabelPemesanan * buttonUpdate * buttonDelete |
| 3 | Halaman Pembelian | Boundary | * inputKodePesanan() * pressBtnProses() * showDataPesanan() * pressBtnPembayaran() | * textFieldKodeItem * buttonProses * buttonPembayaran * buttonBatalPesanan * tabelPembelian |
| 4 | Halaman Pembayaran | Boundary | * inputPembayaran() * pressBtnOK() * showPembayaranKembalian() * pressBtnYA() * printSuccess() * showPembayaranForm() * showUangPembayaranKembalian() * pembayaranEdit() * changeInputUangPembayaran() | * textFieldBayar * buttonOK * buttonEditPembayaran * tabelPembelian * tabelTransaksi * buttonCetakStruk |
| 5 | Controller Pemesanan | Controller | * getInput() * tampilPesanan() * getPesanan() * doPrint(pesanan) * editForm() * tampilPesanan(pesanan) * getKode() | * Model Pemesanan |
| 6 | Controller Pembayaran | Controller | * getInput() * tampilPesanan(pesanan) * tampilTransaksi(pembayaran, kembalian) * savePembayaran() * doPrint(pembayaran) * showPembayaranForm() * tampilPembayaran(pembayaran) | * Model Pembayaran |
| 7 | Model Pemesanan | Model/Entity | * kalkulasiPemesanan() * kalkulasiPembelian(pesanan) * update() * doUpdate(nama, jumlah) * kalkulasiPemesanan(nama, jumlah) * delete() * doDelete(kode) * setPesanan() * doSave(pesanan) | * Kode Pesanan * Nama Pesanan * Jumlah * Harga * Total Pembelian * Kode Pembelian |
| 8 | Model Pembayaran | Model/Entity | * kalkulasiPembayaran(pembelian, pembayaran) * savePembayaran() * doSave(pembayaran) * kalkulasiPembayaran(input) * saveToLog() | * Kode Pembelian * Uang Pembayaran * Tanggal Pembelian |

#### **Relationship antar Kelas Analisis**

Tabel 3.14 Tabel Relationship Antar Kelas Analisis

| **NO** | **From Class** | **To Class** | **Relationship** | **Keterangan** |
| --- | --- | --- | --- | --- |
| 1 | Pembayaran | ModelPembayaran | Association | ModelPembayaran membutuhkan kelas Pembayaran untuk mendapatkan atribut-atribut yang ada pada kelas pembayaran. |
| 2 | ModelPembayaran | ControllerPemesanan | Association | ModelPembayaran akan dibutuhkan pada ControllerPemesanan untuk segala hal yang berurusan mengenai data. |
| 3 | Pemesanan | ModelPemesanan | Association | ModelPemesanan membutuhkan Pemesanan untuk mendapatkan atribut-atribut yang ada di kelas pembayaran. |
| 4 | ModelPemesanan | ControllerPemesanan, ControllerPembayaran | Association | ModelPemesanan dibutuhkan pada ControllerPemesanan dan ControllerPembayaran ketika akan melakukan transaksi. |
| 5 | ControllerPemesanan | ViewPemesanan, ViewEditPemesanan | Association | ControllerPemesanan dibutuhkan pada ViewPemesanan dan ViewEditPemesanan sebagai jembatan antara View dan Model. |
| 6 | ControllerPembayaran | ViewPembelian, ViewPembayaran | Association | ControllerPembayaran dibutuhkan pada ViewPembelian dan ViewPembayaran sebagai jembatan antara View dan Model. |
| 7 | ViewPembayaran | MainActivity | Association | Kelas yang bisa dieksekusi salah satunya ViewPembayaran. |
| 8 | ViewPembelian | MainActivity | Association | Kelas yang bisa dieksekusi salah satunya ViewPembelian |
| 9 | ViewPemesanan | MainActivity | Association | Kelas yang bisa dieksekusi salah satunya ViewPemesanan |
| 10 | ViewEditPemesanan | MainActivity | Association | Kelas yang bisa di eksekusi salah satunya ViewEditPemesanan. |
| 11 | Penjual | ModelPemesanan | Association | Kelas yang dibutuhkan untuk login sebagai penjual. |
| 12 | Kasir | Pembayaran | Association | Kelas yang dibutuhkan untuk login sebagai kasir. |

#### **Paket Analisis**

##### ***Identifikasi Paket Analisis***

Tabel 3.15 Tabel Identifikasi Paket Analisis

| **NO** | **Nama Paket** | **Use Case Terkait** | **Keterangan** |
| --- | --- | --- | --- |
| 1 | Model | Mengkalkulasi Daftar Pembelian. | Untuk mengkalkulasi pembelian. |
| Cetak Daftar Pembelian. | Untuk mencetak pembelian. |
| Kalkulasi Total Pembelian. | Untuk mengkalkulasi grand total dari pembelian. |
| Kalkulasi Transaksi Pembayaran. | Untuk mengkalkulasi transaksi pembayaran. |
| Cetak Struk Pembayaran. | Untuk mencetak struk pembayaran. |
| Mengelola Daftar Pembelian. | Untuk mengelola data-data pada daftar pembelian. |
| Mengelola Struk Pembayaran. | Untuk mengelola struk pembayaran. |
| 2 | View | Mengkalkulasi Daftar Pembelian.  Mengelola Struk Pembayaran. | Untuk mengkalkulasi pembelian. |
| Cetak Daftar Pembelian. | Untuk mencetak pembelian. |
| Kalkulasi Total Pembelian. | Untuk mengkalkulasi grand total dari pembelian. |
| Kalkulasi Transaksi Pembayaran. | Untuk mengkalkulasi transaksi pembayaran. |
| Cetak Struk Pembayaran. | Untuk mencetak struk pembayaran. |
| Mengelola Daftar Pembelian. | Untuk mengelola data-data pada daftar pembelian. |
| Mengelola Daftar Pembelian. | Untuk mengelola struk pembayaran. |
| 3 | Controller | Mengkalkulasi Daftar Pembelian. | Untuk mengkalkulasi pembelian. |
| Cetak Daftar Pembelian. | Untuk mencetak pembelian. |
| Kalkulasi Total Pembelian. | Untuk mengkalkulasi grand total dari pembelian. |
| Kalkulasi Transaksi Pembayaran. | Untuk mengkalkulasi transaksi pembayaran. |
| Cetak Struk Pembayaran. | Untuk mencetak struk pembayaran. |
| Mengelola Daftar Pembelian. | Untuk mengelola data-data pada daftar pembelian. |
| Mengelola Struk Pembayaran. | Untuk mengelola struk pembayaran. |

##### ***Identifikasi Kelas Analisis Tiap Paket***

Tabel 3.16 Tabel Identifikasi Kelas Ananalisis Tiap Paket

|  |  |  |  |
| --- | --- | --- | --- |
| **NO** | **Nama Paket** | **Nama Kelas Analisis** | **Jenis Kelas** |
| 1 | Model | Model Pemesanan | Entity |
| Model Pembayaran | Entity |
| 2 | View | Halaman Pemesanan | Boundary |
| Halaman Edit Pemesanan | Boundary |
| Halaman Pembelian | Boundary |
| Halaman Pembayaran | Boundary |
| 3 | Controller | Controller Pembayaran | Control |
| Controller Pemesanan | Control |

#### **Prototype/Mock-Up Perangkat Lunak**

Dibawah ini akan dijelaskan mengenai prorotype yang terdapat pada aplikasi SoftFood

|  |
| --- |
| **Prototype Perangkat Lunak** |
| Gambar 3.1 Halaman Awal |
| **Penjelasan** |
| Halaman awal yang terdapat pada aplikasi SoftFood. Halaman ini akan menerima input Nama Item dan Jumlah Item yang nantinya akan ditampilkan di tabel setelah menekan tombol Proses. Halaman ini ditangani oleh seorang Penjual. |

|  |
| --- |
| **Prototype Perangkat Lunak** |
| Gambar 3.2 Halaman Pop-up Proses |
| **Penjelasan** |
| Halaman awal yang berisi pop-up, dimana pop-up ini akan memberitahukan proses yang sedang berjalan saat tombol Proses ditekan. Halaman ini ditangani oleh penjual. |

|  |
| --- |
| **Prototype Perangkat Lunak** |
| Gambar 3.3 Halaman Edit Pesanan |
| **Penjelasan** |
| Halaman Edit Pesanan yang fungsinya mengedit pesanaan yang sudah diinput, tetapi ingin ada perubahan karena kesalahan input atau keinginan konsumen. Halaman ini ditangani oleh penjual. |

|  |
| --- |
| **Prototype Perangkat Lunak** |
| Gambar 3.4 Halaman pop-up data pesanan disimpan |
| **Penjelasan** |
| Halaman awal yang berisi pop-up, dimana pop-up ini akan memberitahukan ketika tombol Cetak Struk ditekan, data pesanan sudah disimpan ke Database dan dicetak. |

|  |
| --- |
| **Prototype Perangkat Lunak** |
| Gambar 3.5 Struk Pembayaran |
| **Penjelasan** |
| Struk daftar pembayaran sudah dicetak. Struk ini berisi Atas Nama pemesan dan Kode Pesanan yang nantinya Kode Pesanan ini akan diproses kepada kasir untuk pembayaran |

|  |
| --- |
| **Prototype Perangkat Lunak** |
| Gambar 3.6 Halaman Awal Pembayaran |
| **Penjelasan** |
| Halaman awal pembayaran yang ditangani kasir. Halaman ini akan meminta Kode Pembelian untuk diproses dan ditampilkan ke tabel yang tersedia. Halaman ini juga akan otomatis mengkalkulasi semua yang ada pada pembelian yang sesuai dengan Kode Pembelian. Kemudian menekan tombol Pembayaran untuk meneruskan Pembayaran. Halaman ini ditangani oleh Kasir. |

|  |
| --- |
| **Prototype Perangkat Lunak** |
| Gambar 3.7 Halaman Pembayaran |
| **Penjelasan** |
| Halaman pembayaran yang menampilkan semua total pembayaran beserta pajak nya. Halaman ini bisa menangani pembayaran dari konsumen, memasukan nominal pembayaran dan mengkalkulasi sesuai dengan uang yang diberikan. Akan langsung dikalkulasikan sesuai dengan input uang yang dimasukkan. |

|  |
| --- |
| **Prototype Perangkat Lunak** |
| Gambar 3.8 Halaman pop-up cetak struk |
| **Penjelasan** |
| Halaman pembayaran pop-up yang dimana pop-up ini berisi pertanyaan apakah ingin dicetak atau tidak. Jika Ya, struk akan langsung dicetak. Jika Tidak, akan kembali ke halaman sebelumnya dan memeriksa ulang apakah pesanan yang dipesan sudah benar atau belum. |

|  |
| --- |
| **Prototype Perangkat Lunak** |
| Gambar 3.9 Struk Pembelian |
| **Penjelasan** |
| Struk pembelian yang didapat setelah membayar tagihan di kasir. Struk ini akan memperlihatkan semua pembayaran yang sudah dibayar beserta pajak yang dibayar. |

## **DESAIN**

### **Coding Standard and Naming Convention**

Pada bagian ini diisi dengan standar kode program dan penamaan – penamaan identifier yang digunakan pada kode program. Uraikan tujuan dari sub bab ini dan lengkapi dengan uraian yang menjelaskan coding standar dan naming convention tersebut.

1. **Kelas dan Interface**

Nama kelas dan interface dimulai dengan huruf kapital pada huruf pertamanya selanjutnya menggunakan huruf kecil.

Uraikan dengan nama – nama kelas dan interface yang digunakan pada kode program yang dibuat.

**Kelas & Interface**

Tabel 3.17 Tabel Kelas & Interface

| **No** | **Nama Kelas dan Interface** | **Deskripsi** |
| --- | --- | --- |
| 1 | public class ModelPemesanan {  ...  } | Kelas model ini digunakan untuk mengelola data-data mengenai pemesanan. |
| 2 | public class ModelPembayaran {  ...  } | Kelas model ini digunakan untuk mengelola data-data mengenai pembayaran. |
| 3 | public class ControllerPembayaran {  ...  } | Kelas controller ini digunakan untuk menjembatani antara antarmuka dengan model pembayaran. |
| 4 | public class ControllerPemesanan {  ...  } | Kelas controller ini digunakan untuk menjembatani antara antarmuka dengan model pemesanan. |
| 5 | public class Pemesanan {  ...  } | Kelas yang berisi atribut-atribut yang dibutuhkan oleh kelas model ModelPemesanan. |
| 6 | public class Pembayaran {  ...  } | Kelas yang berisi atribut-atribut yang dibutuhkan oleh kelas model ModelPembayaran. |
| 7 | public class ViewPemesanan extends JFrame{  ...  } | Kelas view ini digunakan untuk interaksi antara aktor dan sistem pada saat pemesanan. |
| 8 | public class ViewEditPemesanan extends JFrame {  ...  } | Kelas view ini digunakan untuk interaksi antara aktor dan sistem saat terjadi kesalahan dan ingin meng-edit pemesanan. |
| 9 | public class ViewPembelian extends JFrame {  ...  } | Kelas view ini digunakan untuk melakukan pembelian antara aktor dan sistem. |
| 10 | public class ViewPembayaran extends JFrame {  ...  } | Kelas view ini digunakan ketika customer dan kasir melakukan pembayaran. |
| 11 | public class MainActivity {  ...  } | Kelas main yang berguna untuk mengeksekusi kelas yang akan di *running* pertama kali. |
| 12 | public class Kasir {  ...  } | Kelas yang berguna untuk menyimpan data-data mengenai kasir untuk perihal login sebagai pembeda hak akses. |
| 13 | public class Penjual {  ...  } | Kelas yang berguna menyimpan data-data mengenai penjual untuk perihal login sebagai pembeda hak akses. |

1. **Data**

Nama data selalu diawali dengan huruf kecil.

Uraikan dengan nama – nama data yang digunakan pada kode program yang dibuat.

Tabel 3.18 Tabel Data

|  |  |  |
| --- | --- | --- |
| **No** | **Nama Data** | **Deskripsi** |
| 1 | private String kodePembelian, kodePesanan, namaPesanan;  private int jumlah, harga; | Data / Variable ini dibutuhkan pada kelas ModelPemesanan. |
| 2 | private String kodePembelian;  private int totalPembelian, uangPembayaran; | Data / Variable yang dibutuhkan pada kelas ModelPembayaran. |
| 3 | ModelPemesanan pesanan; | Data / Variable Reference yang dibutuhkan pada ControllerPemesanan. |

1. **Konstanta**

Di Java, untuk menunjukan konstanta menggunakan kata kunci (keyword) final. Keyword final mengindikasikan bahwa nilai dari variabel tidak dapat dirubah. Penamaan konstanta seluruhnya menggunakan huruf kapital, jika lebih dari satu kata maka dipisahkan menggunakan karakter underscore ( ’\_’ ).

Uraikan dengan nama – nama konstanta yang digunakan pada kode program yang dibuat.

Tabel 3.19 Tabel Konstanta

|  |  |  |
| --- | --- | --- |
| **No** | **Nama Konstanta** | **Deskripsi** |
| 1 | public static final double TAX = 0.1; | Konstanta pada kelas ModelPembayaran yang digunakan pada saat kalkulasi yaitu pajak sebesar 10%. |

1. **Objek**

Penamaan suatu objek (variabel) yang merupakan instansiasi dari kelas sama dengan nama kelasnya atau tipe datanya. Selanjutnya diawal ditambahkan dengan a atau an atau the.

Uraikan dengan nama – nama objek yang digunakan pada kode program yang dibuat.

Tabel 3.20 Tabel Objek

|  |  |  |
| --- | --- | --- |
| **No** | **Nama Objek** | **Deskripsi** |
| 1 | pesanan = new ModelPemesanan(); | Object ini di instansiasi pada kelas ControllerPemesanan dan ControllerPembayaran. |

1. **Metode**

Penamaan Metode (prosedur dan fungsi) sama dengan penamaan pada Data.Dimulai dengan huruf kecil, jika terdiri lebih dari satu kata maka tiap – tiap kata huruf pertamanya menggunakan huruf kapital.

Uraikan dengan nama – nama metode yang digunakan pada kode program yang dibuat.

Tabel 3.21 Tabel Kelas ControllerPembayaran

| **No** | **Nama Kelas dan Interface** | **Deskripsi** |
| --- | --- | --- |
| 1 | public String getInput() {  return null;  } | Fungsi ini digunakan untuk menangkap inputan user. |
| 2 | public void tampilPesanan(ModelPemesanan pesanan) {  } | Method ini digunakan untuk menampilkan pesanan ke layar. |
| 3 | public void tampilTransaksi(ModelPembayaran pembayaran, int kembalian) {  } | Method ini digunakan untuk menampilkan transaksi ke layar. |
| 4 | public void tampilTotal() {    } | Method ini digunakan untuk menampilkan total ke layar. |
| 5 | public void savePembayaran() {  } | Method ini digunakan untuk menyimpan pembayaran ke database. |
| 6 | public void doPrint(ModelPembayaran pembayaran) {  } | Method ini digunakan untuk print struk pembayaran. |
| 7 | public void editPembayaranForm() {    } | Method ini digunakan untuk menampilkan form / halaman edit pembayaran. |

Tabel 3.22 Kelas ControllerPemesanan

| **No** | **Nama Kelas dan Interface** | **Deskripsi** |
| --- | --- | --- |
| 1 | public String getInput() {  return null;  } | Fungsi ini untuk menangkap input dari user. |
| 2 | public String getKode() {  return null;  } | Fungsi ini digunakan untuk menangkap kode. |
| 3 | public void tampilPesanan() {  } | Method ini digunakan untuk menampilkan pesanan ke layar. |
| 4 | public void tampilPesanan(ModelPemesanan pesanan) {  } | Method ini digunakan untuk menampilkan pesanan berdasarkan parameter pesanan ke layar. |
| 5 | public void doPrint(ModelPemesanan pesanan) {  } | Method ini digunakan untuk memcetak pesanan ke layar. |
| 6 | public ModelPemesanan getPesanan() {  return pesanan;  } | Fungsi ini digunakan untuk menangkap pesanan. |
| 7 | public void editForm() {  } | Method ini digunakan untuk menampilkan form edit. |

Tabel 3.23 Kelas ModelPembayaran

|  |  |  |
| --- | --- | --- |
| **No** | **Nama Kelas dan Interface** | **Deskripsi** |
| 1 | public int kalkulasiPembayaran(ModelPemesanan pembelian, ModelPembayaran pembayaran) {  int grandTotal = 0;  return grandTotal;  } | Fungsi ini digunakan untuk mengkalkulasi pembayaran berdasarkan dari pembelian dan pembayaran. |
| 2 | public int kalkulasiPembelian(String input, ModelPemesanan pesanan) {  return 0;  } | Fungsi ini digunakan untuk mengkalkulasi pembelian berdasarkan input dan pesanan. |
| 3 | public void savePembayaran() {  } | Method ini digunakan untuk menyimpan data pembayaran ke database. |
| 4 | public void doSave(ModelPembayaran pembayaran) {  } | Method ini digunakan untuk melakukan eksekusi query pada database. |

Tabel 3.24 Kelas ModelPemesanan

| **No** | **Nama Kelas dan Interface** | **Deskripsi** |
| --- | --- | --- |
| 1 | public int kalkulasiPemesanan() {  int jumlah = 0;    return jumlah;  } | Fungsi ini digunakan untuk mengkalkulasi pemesanan. |
| 2 | public int kalkulasiPembelian(ModelPemesanan pesanan) {  int jumlah = 0;    return jumlah;} | Fungsi ini digunakan untuk mengkalkulasi pembelian berdasarkan pesanan yang di inputkan. |
| 3 | public int kalkulasiPembelian() {  return 0;  } | Fungsi ini digunakan untuk mengkalkulasi pembelian. |
| 4 | public int kalkulasiPembayaran(String input, ModelPemesanan pesanan) {  return 0;  } | Fungsi ini digunakan untuk mengkalkulasi pembayaran berdasarkan input dan pesanan. |
| 5 | public void setPesanan() {    } | Method ini digunakan untuk menyimpan pesanan sementara. |
| 6 | public ModelPemesanan getPesanan() {  return null;  } | Fungsi ini digunakan untuk menangkap pesanan sementara yang sudah di simpan sebelumnya. |
| 7 | public ModelPemesanan doSave() {  return pesanan;  } | Fungsi ini digunakan untuk menyimpan data pesanan ke database. |
| 8 | public void update() {  String nama = null;  int jumlah = 0;  doUpdate(nama, jumlah);  } | Method ini digunakan untuk memperbarui data ke database. |
| 9 | public void doUpdate(String nama, int jumlah) {  kalkulasiPemesanan(nama, jumlah);  } | Method ini digunakan ketika method update() dipanggil dan melakukan eksekusi query di database. |
| 10 | public int kalkulasiPemesanan(String nama, int jumlah) {  return 0;  } | Fungsi ini digunakan untuk mengkalkulasi pesmesanan berdasarkan nama dan jumlah pesanan. |
| 11 | public void delete() {  String kode = null;  doDelete(kode);  } | Method ini digunakan untuk menghapus data dari database. |
| 12 | public void doDelete(String kode) {    } | Fungsi ini digunakan untuk melakukan penghapusan dan melakukan eksekusi query pada database. |

1. **Komponen**

Penamaan komponen untuk kata yang pertama merupakan tipe dari komponen diikuti dengan fungsi dari komponen.

Uraikan dengan nama – nama komponen yang digunakan pada kode program yang dibuat.

Tabel 3.25 Komponen

| **No** | **Nama Komponen** | **Deskripsi** |
| --- | --- | --- |
| - | - | - |

### **Diagram Kelas Perancangan**

|  |
| --- |
| **Diagram Kelas** |
| Gambar 3.66 Diagram Kelas Perancangan |

#### **Deskripsi Kelas Perancangan**

Tabel 3.26 Deskripsi Kelas Perancangan

|  |  |  |
| --- | --- | --- |
| **No.** | **Nama Kelas** | **Deskripsi** |
| 1 | Pemesanan | Merupakan sebuah kelas yang berisi variable-variable yang dibutuhkan oleh ModelPemesanan. |
| 2 | Pembayaran | Merupakan sebuah kelas yang berisi variable-variable yang dibutuhkan oleh ModelPembayaran. |
| 3 | ModelPemesanan | Merupakan sebuah kelas yang mengatur data terkait pemesanan. |
| 4 | ModelPembayaran | Merupakan sebuah kelas yang mengatur data terkait pembayaran. |
| 5 | ControllerPemesanan | Merupakan sebuah kelas penengah yang menjembatani antara ModelPemesanan, ViewPemesanan, dan ViewEditPemesanan. |
| 6 | ControllerPembayaran | Merupakan sebuah kelas penengah yang menjembatani antara ModelPemesanan, ViewPembelian, dan ViewPembayaran. |
| 7 | ViewPemesanan | Halaman interaksi antara system dan actor untuk pemesanan. |
| 8 | ViewEditPemesanan | Halaman interaksi antara system dan actor ketika terjadi kesalahan pada pemesanan. |
| 9 | ViewPembelian | Halaman interaksi untuk pembelian. |
| 10 | ViewPembayaran | Halaman interaksi untuk pembayaran pesanan. |
| 11 | MainActivity | Main Class untuk mengeksekusi system. |
| 12 | Penjual | Merupakan sebuah kelas yang berisi variable-variable yang dibutuhkan oleh ModelPemesanan untuk melakukan login sebagai pembeda hak akses. |
| 13 | Kasir | Merupakan sebuah kelas yang berisi variable-variable yang dibutuhkan oleh ModelPembayaran untuk melakukan login sebagai pembeda hak akses. |

#### **Deskripsi Operasi Kelas**

Tabel 3.27 Deskripsi Operasi Kelas

| **No.** | **Nama Kelas** | **Operasi** | **Visibility** |
| --- | --- | --- | --- |
| **1** | Pemesanan | getKodePembelian() | Public |
| setKodePembelian(kodePembelian) | Public |
| getKodePesanan() | Public |
| setKodePesanan(kodePesanan) | Public |
| getNamaPesanan() | Public |
| setNamaPesanan(namaPesanan) | Public |
| getJumlah() | Public |
| setJumlah(jumlah) | Public |
| getHarga() | Public |
| setHarga(harga) | Public |
| **2** | Pembayaran | getKodePembelian() | Public |
| setKodePembelian(kodePembelian) | Public |
| getTotalPembayaran() | Public |
| setTotalPembayaran(totalPembayaran) | Public |
| getUangPembayaran() | Public |
| setUangPembayaran(uangPembayaran) | Public |
| **3** | ModelPemesanan | kalkulasiPemesanan() | Public |
| kalkulasiPembelian() | Public |
| kalkulasiPembelian(pesanan) | Public |
| kalkulasiPembayaran(input, pesanan) | Public |
| setPesanan() | Public |
| getPesanan() | Public |
| doSave() | Public |
| Update() | Public |
| doUpdate() | Public |
| kalkulasiPemesanan(nam, jumlah) | Public |
| Delete() | Public |
| doDelete() | Public |
| **4** | ModelPembayaran | kalkulasiPembayaran(pembelian, pembayaran) | Public |
| kalkulasiPembelian(input, pesanan) | Public |
| savePembayaran() | Public |
| doSave(pembayaran) | Public |
| **5** | ControllerPemesanan | getInput() | Public |
| getKode() | Public |
| tampilPesanan() | Public |
| tampilPesanan(pesanan) | Public |
| doPrint() | Public |
| getPesanan() | Public |
| editForm() | Public |
| **6** | ControllerPembayaran | getInput() | Public |
| tampilPesanan() | Public |
| tampilTransaksi(pembayaran, kembalian) | Public |
| tampilTotal() | Public |
| savePembayaran() | Public |
| doPrint() | Public |
| editPembayaran() | Public |
| **7** | ViewPemesanan | - | Public |
| **8** | ViewEditPemesanan | - | Public |
| **9** | ViewPembelian | - | Public |
| **10** | ViewPembayaran | - | Public |
| **11** | MainActivity | Main(args) | Public |
| **12** | Penjual | getID() | Public |
| setID(id) | Public |
| getNama() | Public |
| setNama(nama) | Public |
| **13** | Kasir | getID() | Public |
| setID(id) | Public |
| getNama() | Public |
| setNama(nama) | Public |

#### **Deskripsi Atribut Kelas**

Tabel 3.28 Deskripsi Atribut Kelas

| **No.** | **Nama Kelas** | **Atribut** | **Visibility** |
| --- | --- | --- | --- |
| **1** | Pemesanan | kodePembelian: String | Private |
| kodePesanan: String | Private |
| namaPesanan: String | Private |
| Jumlah: Integer | Private |
| Harga: Integer | Private |
| **2** | Pembayaran | kodePembelian: String | Private |
| totalPembayaran: Integer | Private |
| uangPembayaran: Integer | Private |
| **3** | ModelPemesanan | Pesanan: Pesanan | Public |
| **4** | ModelPembayaran | Pembayaran: Pembayaran | Public |
| **5** | ControllerPemesanan | Pesanan: ModelPemesanan | Public |
| TAX: double | Public |
| **6** | ControllerPembayaran | Pesanan: ModelPemesanan | Public |
| **7** | ViewPemesanan | textFieldNamaItem: TextField | Private |
| textFieldJumlahItem: TextField | Private |
| buttonProses: Button | Private |
| buttonCetakStruk: Button | Private |
| buttonEditPesanan: Button | Private |
| tabelPemesanan: Table | Private |
| **8** | ViewEditPemesanan | textFieldNamaItem: TextField | Private |
| textFieldJumlahItem: TextField | Private |
| textFieldKodeItem: TextField | Private |
| buttonProses: Button | Private |
| buttonCetakStruk: Button | Private |
| buttonEditPesanan: Button | Private |
| **9** | ViewPembayaran | textFieldBayar: TextField | Private |
| buttonOK: Button | Private |
| buttonEditPembayaran: Button | Private |
| tablePembelian: Table | Private |
| tableTransaksi: Table | Private |
| buttonCetakStruk: Button | Private |
| **10** | ViewPembelian | textFieldKodeItem: TextField | Private |
| buttonProses: Button | Private |
| buttonPembayaran: Button | Private |
| buttonBatalPesanan: Button | Private |
| tablePembelian: Table | Private |
| **11** | MainActivity | pesananForm: ViewPemesanan | Public |
| editPesananForm: ViewEditPemesanan | Public |
| pembelianForm: ViewPembelian | Public |
| PembayaranForm: ViewPembayaran | Public |
| **12** | Penjual | id\_penjual: String | Private |
| nama\_penjual: String | Private |
| **13** | Kasir | id\_kasir: String | Private |
| nama\_kasir: String | Private |

#### **Deskripsi Relasi Antar Kelas**

Tabel 3.29 Deskripsi Relasi Antar Kelas

| **NO** | **Kelas Asal** | **Kelas Tujuan** | **Relationship** | **Keterangan** |
| --- | --- | --- | --- | --- |
| 1 | Pembayaran | ModelPembayaran | Association | ModelPembayaran membutuhkan kelas Pembayaran untuk mendapatkan atribut-atribut yang ada pada kelas pembayaran. |
| 2 | ModelPembayaran | ControllerPemesanan | Association | ModelPembayaran akan dibutuhkan pada ControllerPemesanan untuk segala hal yang berurusan mengenai data. |
| 3 | Pemesanan | ModelPemesanan | Association | ModelPemesanan membutuhkan Pemesanan untuk mendapatkan atribut-atribut yang ada di kelas pembayaran. |
| 4 | ModelPemesanan | ControllerPemesanan, ControllerPembayaran | Association | ModelPemesanan dibutuhkan pada ControllerPemesanan dan ControllerPembayaran ketika akan melakukan transaksi. |
| 5 | ControllerPemesanan | ViewPemesanan, ViewEditPemesanan | Association | ControllerPemesanan dibutuhkan pada ViewPemesanan dan ViewEditPemesanan sebagai jembatan antara View dan Model. |
| 6 | ControllerPembayaran | ViewPembelian, ViewPembayaran | Association | ControllerPembayaran dibutuhkan pada ViewPembelian dan ViewPembayaran sebagai jembatan antara View dan Model. |
| 7 | ViewPembayaran | MainActivity | Association | Kelas yang bisa dieksekusi salah satunya ViewPembayaran. |
| 8 | ViewPembelian | MainActivity | Association | Kelas yang bisa dieksekusi salah satunya ViewPembelian |
| 9 | ViewPemesanan | MainActivity | Association | Kelas yang bisa dieksekusi salah satunya ViewPemesanan |
| 10 | ViewEditPemesanan | MainActivity | Association | Kelas yang bisa di eksekusi salah satunya ViewEditPemesanan. |
| 11 | Penjual | ModelPemesanan | Association | Kelas yang dibutuhkan pada ModelPemesanan untuk perihal login sebagai pembeda Hak Akses. |
| 12 | Kasir | ModelPembayaran | Association | Kelas yang dibutuhkan pada ModelPembayaran untuk perihal login sebagai pembeda Hak Akses. |

### **Perancangan Data dengan ORM**

|  |
| --- |
| **Diagram ORM** |
| Gambar 3.67 Diagram ORM Tabel Pemesanan    Gambar 3.68 Diagram ORM Tabel Pembayaran    Gambar 3.69 Diagram ORM Tabel Penjual    Gambar 3.70 Diagram ORM Tabel Kasir |

### **Deskripsi Algoritma / Query Operasi**

Tabel 3.30 Deskripsi Algoritma / Query Operasi

| **No.** | **Nama Kelas** | **Operasi** | **Algoritma / Query** |
| --- | --- | --- | --- |
| **1** | Pemesanan | getKodePembelian() | return kodePembelian; |
| setKodePembelian(kodePembelian) | this.kodePembelian = kodePembelian; |
| getKodePesanan() | return kodePesanan; |
| setKodePesanan(kodePesanan) | this.kodePesanan = kodePesanan; |
| getNamaPesanan() | return namaPesanan; |
| setNamaPesanan(namaPesanan) | this.namaPesanan = namaPesanan; |
| getJumlah() | return jumlah; |
| setJumlah(jumlah) | this.jumlah = jumlah; |
| getHarga() | return harga; |
| setHarga(harga) | this.harga = harga; |
| **2** | Pembayaran | getKodePembelian() | return kodePembelian; |
| setKodePembelian(kodePembelian) | this.kodePembelian = kodePembelian; |
| getTotalPembayaran() | return totalPembayaran; |
| setTotalPembayaran(totalPembayaran) | this.totalPembayaran = totalPembayaran; |
| getUangPembayaran() | return uangPembayaran; |
| setUangPembayaran(uangPembayaran) | this.uangPembayaran = uangPembayaran; |
| **3** | ModelPemesanan | kalkulasiPemesanan() | pemesanan = jumlah \* harga; |
| kalkulasiPembelian() | total = (pemesanan \* jumlah); |
| kalkulasiPembelian(pesanan) | total = (pesanan \* jumlah); |
| kalkulasiPembayaran(input, pesanan) | bayar = input \* pesanan; |
| setPesanan() | this.pesanan = pesanan; |
| getPesanan() | return pesanan; |
| doSave() | INSERT INTO tbl\_pemesanan values (pesanan); |
| Update() | doUpdate(); |
| doUpdate() | UPDATE tbl\_pemesanan SET pesanan; |
| kalkulasiPemesanan(nama, jumlah) | pemesanan = (nama as harga) \* jumlah |
| Delete() | doDelete(); |
| doDelete() | DELETE FROM tbl\_pemesanan; |
| **4** | ModelPembayaran | kalkulasiPembayaran(pembelian, pembayaran) | bayar = pembayaran - pembelian; |
| kalkulasiPembelian(input, pesanan) | pembelian = pesanan \* input; |
| savePembayaran() | doSave(); |
| doSave(pembayaran) | INSERT INTO tbl\_pembayaran values (pembayaran); |
| **5** | ControllerPemesanan | getInput() | return input; |
| getKode() | return kode; |
| tampilPesanan() | this.pesanan.show(); |
| tampilPesanan(pesanan) | pesanan.show(); |
| doPrint() | print(); |
| getPesanan() | return pesanan; |
| editForm() | editForm.show() |
| **6** | ControllerPembelian | getKode() | return kode; |
| tampilPesanan(pesanan) | pesanan.show(); |
| pembayaranForm() | pembayaran.show(); |
| getInput() | return input; |
| tampilTotal(pembayaran, kembalian) | total.show(pembayaran, kembalian); |
| **7** | ControllerPembayaran | getInput() | return input; |
| tampilPesanan() | pesanan.show(); |
| tampilTransaksi(pembayaran, kembalian) | transaksi.show(pembayaran, kembalian); |
| tampilTotal() | total.show(); |
| savePembayaran() | INSERT INTO tbl\_pembayaran values (pembayaran); |
| doPrint() | print(); |
| editPembayaran() | pembayaran.show(); |
| **8** | ViewPemesanan | - | - |
| **9** | ViewEditPemesanan | - | - |
| **10** | ViewPembelian | - | - |
| **11** | ViewPembayaran | - | - |
| **12** | MainActivity | Main(args) | exec.view(); |
| **13** | Penjual | getID() | return id\_penjual; |
| setID(id) | this.id\_penjual = id; |
| getNama() | return nama\_penjual; |
| setNama(nama) | this.nama\_penjual = nama; |
| **14** | Kasir | getID() | return id\_kasir; |
| setID(id) | this.id\_kasir = id; |
| getNama() | return nama\_kasir; |
| setNama(nama) | this.nama\_kasir = nama; |

### **Antarmuka Perangkat Lunak**

#### **Perancangan Antarmuka Aplikasi**

Bagian ini akan mejelaskan tentang semua perancangan antarmuka yang sudah di screenshot

|  |
| --- |
| **Prototipe Perangkat Lunak** |
| Gambar 3.71 Gambar Halaman Awal Pemesanan |

|  |
| --- |
| **Prototipe Perangkat Lunak** |
| Gambar 3.72 Gambar pop-up pemesanan diproses |

|  |
| --- |
| **Prototipe Perangkat Lunak** |
| Gambar 3.73 Gambar isi halaman pemesanan |

|  |
| --- |
| **Prototipe Perangkat Lunak** |
| Gambar 3.74 Gambar isi halaman yang sudah dicetak |

|  |
| --- |
| **Prototipe Perangkat Lunak** |
| Gambar 3.75 Gambar Struk Pembelian |

|  |
| --- |
| **Prototipe Perangkat Lunak** |
| Gambar 3.76 Gambar awal Pembelian |

|  |
| --- |
| **Prototipe Perangkat Lunak** |
| Gambar 3.77 Gambar Halaman Pembelian yang sudah diisi Kode Pembelian |

|  |
| --- |
| **Prototipe Perangkat Lunak** |
| Gambar 3.78 Gambar Halaman Pembayaran |

|  |
| --- |
| **Prototipe Perangkat Lunak** |
| Gambar 3.79 Gambar Halaman Pembayaran yang sudah diisi input uang |

|  |
| --- |
| **Prototipe Perangkat Lunak** |
| Gambar 3.80 Gambar Halaman Pembayaran pop-up print |

|  |
| --- |
| **Prototipe Perangkat Lunak** |
| Gambar 3.81 Gambar Struk Transaksi |

#### **Deskripsi Prototipe Perangkat Lunak**

1. Pada bagian ini diisi dengan daftar objek dan deskripsi singkat mengenai objek – objek yang ada pada prototype perangkat lunak tersebut. Uraikan tujuan dari sub bab ini dan deskripsikan dengan menggunakan tabel seperti di bawah ini.

Tabel 3.31 Tabel Deskripsi Prototipe Perangkat Lunak

| **NO** | **Nama Objek** | **Metode** | **Event** | **Algoritma / Query** |
| --- | --- | --- | --- | --- |
| 1 | Button Cetak Struk | doPrint() | onClick() | print(); |
| 2 | Button Edit Pesanan | editPesananForm() | onClick() | editPesananForm.show(); |
| 3 | Tombol Proses | tampilPesanan() | onClick() | pesanan.show() |
| 4 | Tombol Pembayaran | PembayaranForm() | onClick() | PembayaranForm.Show(); |
| 5 | Tombol Batal Pesanan | cancel() | onClick() | table.setNull(); |
| 6 | Tombol OK Pembayaran | kalkulasiPembayaran | onClick() | bayar = input \* pesanan; |
| 7 | Tombol Cetak Struk Pembayaran | doPrint() | onClick() | popup.show() |
| 8 | Popup Mencetak Struk YA | print() | onClick() | print() |
| 9 | Popup Mencetak Struk TIDAK | cancel() | onClick | this.popup.close() |

### **Perancangan Arsitektur / MVC**

Tabel 3.32 Perancangan Arsitektur – Model

|  |  |  |
| --- | --- | --- |
| **No.** | **Nama Kelas** | **Derkripsi** |
| **1** | ModelPemesanan | Kelas model ini digunakan untuk mengelola data-data mengenai pemesanan. |
| **2** | ModelPembayaran | Kelas model ini digunakan untuk mengelola data-data mengenai pembayaran. |

Tabel 3.33 Perancangan Arsitektur – View

|  |  |  |
| --- | --- | --- |
| **No.** | **Nama Kelas** | **Derkripsi** |
| **1** | ViewPemesanan | Kelas view ini digunakan untuk interaksi antara aktor dan sistem pada saat pemesanan. |
| **2** | ViewEditPemesanan | Kelas view ini digunakan untuk interaksi antara aktor dan sistem saat terjadi kesalahan dan ingin meng-edit pemesanan. |
| **3** | ViewPembelian | Kelas view ini digunakan untuk melakukan pembelian antara aktor dan sistem. |
| **4** | ViewPembayaran | Kelas view ini digunakan ketika customer dan kasir melakukan pembayaran. |

Tabel 3.34 Perancangan Arsitektur – Controller

|  |  |  |
| --- | --- | --- |
| **No.** | **Nama Kelas** | **Derkripsi** |
| **1** | ControllerPembayaran | Kelas controller ini digunakan untuk menjembatani antara antarmuka dengan model pembayaran. |
| **2** | ControllerPemesanan | Kelas controller ini digunakan untuk menjembatani antara antarmuka dengan model pemesanan. |

### **Daftar File**

Tabel 3.35 Daftar FIle

| **No** | **Kelas Analis** | **Kelas Diagram** | **Koding** |
| --- | --- | --- | --- |
| 1 |  |  | ViewPemesanan.java |
| 2 |  |  | ModelPemesanan.java |
| 3 |  |  | ControllerPemesanan.java |
| 4 |  |  | ViewEditPemesanan.java |
| 5 |  |  | ViewPembelian.java |
| 6 |  |  | ViewPembayaran.java |
| 7 |  |  | ModelPembayaran.java |
| 8 |  |  | ControllerPembayaran.java |

### **Daftar Data**

Tabel 3.36 Daftar Data

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Kelas design** | **Nama tabel** | **Atribut** |
| 1 |  | tbl\_pemesanan | - KodePesanan varchar(10)  - NamaPesanan varchar(30)  - Jumlah int(11)  - Harga int(11) |
| 2 |  | tbl\_pembayaran | - KodePembelian varchar(10)  - UangPembayaran int(11)  - TanggalPembelian Date |
| 3 |  | tbl\_penjual | - Id\_Penjual varchar(10)  - NamaPenjual varchar(30) |
| 4 |  | tbl\_kasir | - Id\_Kasir varchar(10)  - NamaKasir varchar(30) |