Software Requirements Specification

Sistem Point Of Sales

Retail Minimarket

Oleh :

Komang Arinanda

Nikolaus Aldo Halim

Blibli FUTURE Program Batch 2.0

**Agustus 2017**

**Daftar Isi**

Penjelasan sistem

Poin of Sales adalah sebuah sistem yang dapat mengelola transaksi jual beli, perhitungan akuntansi, manajemen barang dan stok, laporan laba rugi. Pada sistem ini terdapat 2 buah role user yaitu Admin dan Kasir. Fungsionalitas yang dapat dilakukan oleh masing - masing role digambarkan dalam use case diagram berikut.



Gambar : use case diagram sistem POS minimarket

Terdapat fungsi login, jika user login sebagai Kasir maka ia dapat melakukan beberapa fungsionalitas yaitu:

1. Mengelola Akun

Fungsionalitas ini meliputi ubah foto, nama, email, dan password.

1. Melakukan Transaksi

Fungsionalitas ini meliputi input data transaksi, proses pembayaran, dan mencetak receipt.

Jika user login sebagai Admin maka ia dapat melakukan semua fungsionalitas yang dapat dilakukan oleh role Kasir. Ini dikarenakan Kasir merupakan generalisasi dari Admin. Sebagai Admin terdapat beberapa fungsional lain yang dapat dilakukan yaitu:

1. Mengelola Informasi Minimarket

Fungsionalitas ini meliputi mengubah informasi nama, alamat, nomor telp, email minimarket, dan keterangan (berisi ucapan terimakasih atau informasi promo yang ditampilkan di receipt).

1. Mengelola Pegawai

Fungsionalitas ini meliputi tambah, ubah, hapus, tampil dan cari untuk data pegawai.

1. Mengelola Produk

Fungsionalitas ini meliputi tambah, ubah, hapus, tampil dan cari untuk data kategori dan data barang.

1. Mengelola Stok

Fungsionalitas ini digunakan untuk penambahan stok

1. Mengelola Promo

Fungsionalitas ini meliputi tambah, ubah, hapus, tampil dan cari untuk data Promo atau diskon.

1. Melihat Laporan

Fungsionalitas ini meliputi melihat laporan data penjualan dan pembelian dalam mingguan, bulanan, atau tahun.

Use Case Description

A. Mengelola Akun

1. Login

|  |  |
| --- | --- |
| **Nama Use Case** | Login |
| **Aktor** | User (Admin atau Kasir) |
| **Deskripsi Singkat** | Use case menggambarkan bagaimana user melakukan login ke sistem POS |
| **Pre Condition** | User harus telah memiliki ID Pegawai dan Password yang terdata dalam sistem |
| **Flow Of Event** | 1. Use case dimulai ketika user mengakses halaman login sistem |
| 2. Sistem menampilkan halaman login dan meminta user memasukkan ID Pegawai dan Password |
| 3. User memasukkan ID Pegawai dan password |
| 4. User menekan tombol login |
| 5. Sistem melakukan pengecekan ID Pegawai dan password, |
| 6. Sistem menampilkan pesan bahwa login berhasil |
| 7. Sistem mengecek role yang dimiliki user |
| 8. Sistem menampilkan halaman awal sesuai dengan role yang dimiliki user |
| 9. Use case selesai |
| **Post Condition** | Sistem menampilkan halaman awal untuk user sesuai dengan role yang dimiliki user |
| **Alternative Flow** | Pada langkah 3, jika user memasukkan ID Pegawai atau password yang salah, sistem akan meminta user memasukkan ID Pegawai dan password yang benar |

1. Mengelola Akun

|  |  |
| --- | --- |
| **Nama Use Case** | Mengelola Akun |
| **Aktor** | User (Admin atau Kasir) |
| **Deskripsi Singkat** | Use case menggambarkan bagaimana user mengubah foto, nama, dan email. |
| **Pre Condition** | User telah diautorisasi, memiliki data baru yang akan diubah |
| **Flow Of Event** | 1. Use case dimulai ketika user memilih halaman mengelola akun 2. Sistem menampilkan halaman mengelola akun yang berisi foto, nama, dan email lama user 3. User dapat memperbarui foto atau nama dengan data baru 4. User dapat menginputkan email baru 5. Sistem melakukan validasi email 6. User menekan tombol Simpan 7. Sistem menampilkan pesan konfirmasi 8. User menekan tombol Ya pada pilihan yang muncul 9. Sistem memperbarui data user 10. Sistem menampilkan pesan berhasil 11. Use case berhasil |
| **Post Condition** | Data akun berupa foto, nama, atau email diperbarui |
| **Alternative Flow** | Pada langkah 4, jika user menginputkan email yang tidak valid maka sistem meminta inputan email yang valid  Pada langkah 8, jika user memilih tombol tidak pada pilihan yang muncul maka proses pembaruan akun dibatalkan dan proses kembali ke langkah 1 |

1. Mengubah Password

|  |  |
| --- | --- |
| **Nama Use Case** | Mengubah password |
| **Aktor** | User (Admin atau Kasir) |
| **Deskripsi Singkat** | Use case menggambarkan bagaimana user mengubah password akun |
| **Pre Condition** | User mengakses halaman mengelola akun, dan mengklik tombol ubah akun pada tampilan yang muncul, user memiliki data password lama dan password baru |
| **Flow Of Event** | 1. Use case dimulai ketika user mengakses halaman mengelola akun dan mengklik tombol ubah password 2. Sistem menampilkan form ubah password 3. User menginputkan password lama, password baru, dan input ulang password baru 4. User mengklik tombol simpan 5. Sistem memvalidasi password lama 6. Sistem memvalidasi password baru dengan inputan ulang password 7. Sistem menampilkan pesan konvirmasi ubah password 8. User mengklik tombol Ya 9. Sistem memperbarui password 10. Sistem menampilkan pesan berhasil 11. Use case selesai |
| **Post Condition** | Password pada sistem diperbaharui |
| **Alternative Flow** | Pada langkah 3, jika user menginputkan password lama yang salah maka sistem meminta inputan password lama yang benar  Pada langkah 3, jika user menginputkan password baru dan inputan ulang password baru yang tidak sama persis maka sistem meminta password baru yang benar  Pada langkah 8 jika user mengklik tombol Tidak pada pilihan yang muncul maka proses perbarui password dibatalkan dan proses kembali ke langkah 1 |

B. Mengelola Informasi Minimarket

|  |  |
| --- | --- |
| **Nama Use Case** | Mengelola Informasi Minimarket |
| **Aktor** | Admin |
| **Deskripsi Singkat** | Use case menggambarkan bagaimana Admin mengelola informasi nama, alamat, nomor telp, email, dan keterangan (receipt) minimarket |
| **Pre Condition** | Admin harus sudah diotorisasi oleh sistem |
| **Flow Of Event** | 1. Use case dimulai ketika Admin masuk ke halaman Pengelolaan Informasi Minimarket  2. Sistem menampilkan informasi logo, nama, alamat, nomot telp, email dan keterangan (receipt) lama dan meminta inputan informasi logo, nama, alamat, nomot telp, email dan keterangan (receipt) yang baru  3. Admin menginputkan logo, nama, alamat, nomot telp, email dan keterangan (receipt) yang baru kemudian mengklik tombol Simpan  4. Sistem menampilkan pesan konfirmasi update  5. Admin melakukan konfirmasi update dengan memilik Ya pada sistem  6. Sistem memperbarui informasi nama, alamat, nomot telp, email dan keterangan (receipt)  7. Use case selesai |
| **Post Condition** | Sistem memperbarui informasi nama, alamat, nomot telp, email dan keterangan (receipt) |
| **Alternative Flow** | Pada langkah 5, jika Admin memilih Tidak mengkonfirmasi update maka proses update dibatalkan dan use case selesai |

C. Mengelola Pegawai

1. Tampil Pegawai

|  |  |
| --- | --- |
| **Nama Use Case** | Tampil Pegawai |
| **Aktor** | Admin |
| **Deskripsi Singkat** | Use case menggambarkan bagaimana Admin melakukan proses mencari dan menampilkan pegawai |
| **Pre Condition** | Admin memilih halaman mengelola pegawai, memiliki ID atau nama pegawai yang akan dicari dan tampilkan |
| **Flow Of Event** | 1. Use case dimulai ketika Admin akan melihat, mengubah, ataupun menghapus pegawai tertentu dan mengakses halaman mengelola pegawai  2. Sistem meminta inputan berupa ID atau Nama pegawai yang dicari  3. Admin menginputkan ID atau Nama pegawai yang dicari  4. Sistem melakukan pencarian pegawai  5. Sistem menampilkan ID, Nama, Role, dan email pegawai  6. Use case selesai |
| **Post Condition** | Sistem menampilkan ID, Nama, Role, dan email pegawai |
| **Alternative Flow** | Admin dapat melakukan Export hasil kedalam bentuk file dokumen dengan mengklik button Export, Sistem akan mengeksport data kategori kedalam file dokumen.  Pada langkah 4, jika sistem tidak menemukan ID atau nama pegawai yang dicari maka sistem akan menampilkan pesan kesalahan bahwa ID atau nama pegawai yang dicari tidak ditemukan |

2. Tambah Pegawai

|  |  |
| --- | --- |
| **Nama Use Case** | Tambah Pegawai |
| **Aktor** | Admin |
| **Deskripsi Singkat** | Use case menggambarkan bagaimana Admin menambahkan data pegawai baru |
| **Pre Condition** | Admin memilih halaman mengelola pegawai dan memiliki data pegawai baru yang akan ditambahkan |
| **Flow Of Event** | 1. Use case dimulai ketika Admin memilih button tambah pada halaman mengelola pegawai  2. Sistem meminta inputan ID pegawai baru  3. Sistem melakukan pengecekan keunikan kode pegawai baru  4. Admin menginputkan nama pegawai  5. Admin menginputkan role pegawai  6. Sistem menset password dengan “password”  7. Admin menginputkan email pegawai  8. Admin mengklik button Tambah  9. Sistem menampilkan pesan konfirmasi tambah pegawai  10. Admin memilih Ya pada pilihan yang muncul  11. Sistem mencatat data pegawai baru  12. Use case selesai |
| **Post Condition** | Sistem mencatat data kategori baru |
| **Alternative Flow** | Pada langkah 3, jika Admin menginputkan ID pegawai yang tidak unik maka sistem menampilkan pesan kesalahan dan kemudian meminta Admin menginputkan ID yang baru dan unik  Pada langkah nomor 9, jika Admin memilih “Tidak” pada pilihan yang muncul maka sistem membatalkan penambahan pegawai dan proses kembali ke langkah 1 |

3. Mengubah Pegawai

|  |  |
| --- | --- |
| **Nama Use Case** | Mengubah Pegawai |
| **Aktor** | Admin |
| **Deskripsi Singkat** | Use case menggambarkan bagaimana Admin mengubah data pegawai tertentu |
| **Pre Condition** | Admin memilih halaman mengelola pegawai dan memiliki ID atau nama pegawai yang akan diubah serta data yang akan diubah |
| **Flow Of Event** | * + - 1. Use case dimulai ketika Admin memilih pegawai yang akan diubah (dapat melakukan pencarian terlebih dahulu)  1. Sistem melakukan pencarian data pegawai 2. Admin mengklik button Ubah 3. Sistem menampilkan form ubah pegawai 4. Admin menginputkan nama baru 5. Admin menginputkan role baru 6. Admin menginputkan email baru 7. Sistem menampilkan pesan konfirmasi update 8. Admin memilih “Ya” pada pilihan yang ada 9. Sistem memperbarui nama, role, dan email 10. Use case selesai |
| **Post Condition** | Sistem memperbarui nama, role, dan email |
| **Alternative Flow** | Pada langkah 9, jika Admin memilih “Tidak” pada pilihan yang muncul maka proses mengubah data pegawai dibatalkan dan proses kembali ke langkah 1 |

4. Menghapus Pegawai

|  |  |
| --- | --- |
| **Nama Use Case** | Menghapus Pegawai |
| **Aktor** | Admin |
| **Deskripsi Singkat** | Use case menggambarkan bagaimana Admin menghapus data pegawai tertentu |
| **Pre Condition** | Admin memilih halaman mengelola pegawai dan memiliki ID atau nama pegawai yang akan dihapus |
| **Flow Of Event** | 1. Use case dimulai ketika Admin mengklik button hapus pada halaman mengelola pegawai  2. Admin memilih data pegawai yang akan dihapus (dapat dilakukan dengan pencarian)  3. Sistem melakukan pencarian data pegawai  4. Sistem menampilkan pesan konfirmasi hapus  5. Admin memilih “Ya” pada pilihan yang ada  6. Sistem menghapus data pegawai  7. Use case selesai |
| **Post Condition** | Sistem menghapus data pegawai |
| **Alternative Flow** | Pada langkah 5, jika Admin memilih “Tidak” pada pilihan yang muncul maka proses menghapus kategori dibatalkan dan proses kembali ke langkah 1 |

D. Mengelola Produk

1. Mengelola Kategori

a. Tampil Kategori

|  |  |
| --- | --- |
| **Nama Use Case** | Tampil Kategori |
| **Aktor** | Admin |
| **Deskripsi Singkat** | Use case menggambarkan bagaimana Admin melakukan proses mencari dan menampilkan kategori |
| **Pre Condition** | Admin memilih halaman mengelola kategori, memiliki kode atau nama kategori yang akan dicari dan tampilkan |
| **Flow Of Event** | 1. Use case dimulai ketika Admin akan melihat, melakukan pengubahan pada data kategori ataupun menghapus kategori tertentu  2. Sistem meminta inputan berupa Kode atau Nama kategori yang dicari  3. Admin menginputkan Kode atau Nama Kategori yang dicari  4. Sistem melakukan pencarian kategori  5. Sistem menampilkan Kode, Nama, dan deskripsi kategori  6. Use case selesai |
| **Post Condition** | Sistem menampilkan kode, nama, dan deskripsi kategori |
| **Alternative Flow** | Admin dapat melakukan Export tampilan kedalam bentuk file dokumen dengan mengklik button Export, Sistem akan mengeksport data kategori kedalam file dokumen.  Pada langkah 4, jika sistem tidak menemukan kode atau nama kategori yang dicari maka sistem akan menampilkan pesan kesalahan bahwa kode atau nama kategori yang dicari tidak ditemukan |

b. Tambah kategori

|  |  |
| --- | --- |
| **Nama Use Case** | Tambah Kategori |
| **Aktor** | Admin |
| **Deskripsi Singkat** | Use case menggambarkan bagaimana Admin menambahkan sebuah kategori baru |
| **Pre Condition** | Admin memilih halaman mengelola kategori dan memiliki data kategori baru yang akan ditambahkan |
| **Flow Of Event** | 1. Use case dimulai ketika Admin memilih tampilan tambah kategori  2. Sistem meminta inputan Kode Kategori baru  3. Admin menginputkan Kode Kategori Baru  4. Sistem melakukan pengecekan keunikan kode kategori baru  5. Admin menginputkan nama kategori baru  6. Admin menginputkan deskripsi kategori  7. Sistem menampilkan pesan konfirmasi tambah kategori  8. User memilih Ya pada pilihan yang muncul  9. Sistem mencatat data kategori baru  10. Use case selesai |
| **Post Condition** | Sistem mencatat data kategori baru |
| **Alternative Flow** | Pada langkah 3, jika Admin menginputkan kode kategori yang tidak unik maka sistem menampilkan pesan kesalahan dan kemudian meminta Admin menginputkan kode kategori yang baru dan unik  Pada langkah nomor 8, jika Admin memilih “Tidak” pada pilihan yang muncul maka sistem membatalkan penambahan kategori dan proses kembali ke langkah 1 |

c. Mengubah Kategori

|  |  |
| --- | --- |
| **Nama Use Case** | Mengubah Kategori |
| **Aktor** | Admin |
| **Deskripsi Singkat** | Use case menggambarkan bagaimana Admin mengubah data kategori tertentu |
| **Pre Condition** | Admin memilih halaman mengelola kategori dan memiliki kode atau nama kategori yang akan diubah serta nama atau deskripsi baru |
| **Flow Of Event** | * + - 1. Use case dimulai ketika Admin memilih kotegori yang akan diubah, juga dapat melakukan pencarian terlebih dahulu  1. Sistem melakukan pencarian data kategori 2. Admin mengklik button Ubah 3. Sistem menampilkan form ubah kategori 4. Admin menginputkan nama kategori yang baru 5. Admin menginputkan deskripsi kategori yang baru 6. Sistem menampilkan pesan konfirmasi update 7. Admin memilih “Ya” pada pilihan yang ada 8. Sistem memperbarui nama dan deskripsi kategori 9. Use case selesai |
| **Post Condition** | Sistem memperbarui nama dan deskripsi kategori |
| **Alternative Flow** | Pada langkah 6, jika Admin memilih “Tidak” pada pilihan yang muncul maka proses mengubah kategori dibatalkan dan proses kembali ke langkah 1 |

d. Menghapus Kategori

|  |  |
| --- | --- |
| **Nama Use Case** | Menghapus Kategori |
| **Aktor** | Admin |
| **Deskripsi Singkat** | Use case menggambarkan bagaimana Admin menghapus data kategori tertentu |
| **Pre Condition** | Admin memilih halaman mengelola kategori dan memiliki kode atau nama kategori yang akan dihapus, tidak ada data barang pada kategori yang akan dihapus |
| **Flow Of Event** | 1. Use case dimulai ketika Admin menginputkan kode atau nama kategori yang akan dihapus  2. Sistem melakukan pencarian data kategori  3. Sistem memeriksa keberadaan data barang pada kategori yang akan dihapus  4. Sistem menampilkan pesan konfirmasi hapus  5. Admin memilih “Ya” pada pilihan yang ada  6. Sistem menghapus data kategori  7. Use case selesai |
| **Post Condition** | Sistem menghapus data kategori |
| **Alternative Flow** | Pada langkah 3, jika ada data barang dalam kategori tersebut maka proses hapus kategori gagal dan proses kembali ke langkah 1  Pada langkah 5, jika Admin memilih “Tidak” pada pilihan yang muncul maka proses menghapus kategori dibatalkan dan proses kembali ke langkah 1 |

2. Mengelola Produk

a. Tampil Produk

|  |  |
| --- | --- |
| **Nama Use Case** | Tampil Produk |
| **Aktor** | Admin |
| **Deskripsi Singkat** | Use case menggambarkan bagaimana Admin melakukan proses pencarian dan menampilkan barang |
| **Pre Condition** | Admin memilih halaman mengelola barang dan memiliki kode atau nama barang yang akan dicari dan tampilkan |
| **Flow Of Event** | 1. Use case dimulai ketika Admin akan melihat, mengubah, atau menghapus barang tertentu  2. Sistem meminta inputan berupa Kode atau Nama kategori yang dicari  3. Admin menginputkan Kode atau Nama Barang yang dicari  4. Sistem melakukan pencarian barang  5. Sistem menampilkan Kode, Nama Kategori, Nama Barang, deskripsi barang, stok, dan harga barang  6. Use case selesai |
| **Post Condition** | Sistem menampilkan kode, nama kategori, nama barang,deskripsi, stok, dan harga barang |
| **Alternative Flow** | Admin dapat melakukan Export hasil kedalam bentuk file dokumen dengan mengklik buttn Export. Sistem akan mengeksport data kategori kedalam file dokumen.  Pada langkah 4, jika sistem tidak menemukan kode atau nama barang yang dicari maka sistem akan menampilkan pesan kesalahan bahwa kode atau nama barang yang dicari tidak ditemukan |

b. Tambah Produk

|  |  |
| --- | --- |
| **Nama Use Case** | Tambah Produk |
| **Aktor** | Admin |
| **Deskripsi Singkat** | Use case menggambarkan bagaimana Admin menambahkan sebuah barang baru |
| **Pre Condition** | Admin memilih halaman mengelola barang dan memiliki data barang baru yang akan ditambahkan |
| **Flow Of Event** | 1. Use case dimulai ketika sistem meminta input kode barang baru  2. Sistem mengecek keunikan kode barang  3. Admin menginputkan nama kategori barang  4. Admin menginputkan nama barang  5. Admin menginputkan deskripsi barang  6. Sistem menginisialisasi stok dengan nilai 0  7. Sistem menginisialisasi harga dengan nilai 0  8. Sistem menampilkan pesan konfirmasi tambah barang  9. Admin memilih Ya pada pilihan yang muncul  10. Sistem mencatat data barang baru  11. Use case selesai |
| **Post Condition** | Sistem mencatat data barang baru |
| **Alternative Flow** | Pada langkah 1, jika Admin menginputkan kode barang yang tidak unik atau sudah terdata dalam sistem maka sistem menampilkan pesan kesalahan dan kemudian meminta Admin menginputkan kode barang yang baru dan unik  Pada langkah nomor 9, jika Admin memilih “Tidak” pada pilihan yang muncul maka sistem membatalkan penambahan barang dan proses kembali ke langkah 1 |

c. Mengubah Produk

|  |  |
| --- | --- |
| **Nama Use Case** | Mengubah Produk |
| **Aktor** | Admin |
| **Deskripsi Singkat** | Use case menggambarkan bagaimana Admin mengubah data barang tertentu |
| **Pre Condition** | Admin memilih halaman mengelola barang dan memiliki kode atau nama barang yang akan diubah serta nama kategori, nama barang, atau deskripsi baru |
| **Flow Of Event** | 1. Use case dimulai ketika Admin memilih barang yang akan diubah yang dapat dilakukan dengan melakukan pencarian terlebihdahulu  2. Sistem melakukan pencarian data barang  3. Admin mengklik button Ubah  4. Sistem menampilkan form ubah barang  4. Admin menginputkan nama kategori yang baru  5. Admin menginputkan nama barang yang baru  6. Admin menginputkan deskripsi barang yang baru  8. Sistem menampilkan pesan konfirmasi update  9. Admin memilih “Ya” pada pilihan yang ada  10. Sistem memperbarui nama kategori, nama barang, deskripsi barang.  11. Use case selesai |
| **Post Condition** | Sistem memperbarui nama kategori, nama barang dan deskripsi barang |
| **Alternative Flow** | Pada langkah 6, jika Admin memilih “Tidak” pada pilihan yang muncul maka proses mengubah barang dibatalkan dan proses kembali ke langkah 1 |

d. Menghapus Produk

|  |  |
| --- | --- |
| **Nama Use Case** | Menghapus Produk |
| **Aktor** | Admin |
| **Deskripsi Singkat** | Use case menggambarkan bagaimana Admin menghapus data barang tertentu |
| **Pre Condition** | Admin memilih halaman mengelola barang dan memiliki kode atau nama barang yang akan dihapus |
| **Flow Of Event** | 1. Use case dimulai ketika Admin menginputkan kode atau nama barang yang akan dihapus  2. Sistem melakukan pencarian data barang  3. Sistem memastikan bahwa stok barang kosong  4. Sistem menampilkan pesan konfirmasi hapus  5. Admin memilih “Ya” pada pilihan yang ada  6. Sistem menghapus data barang  7. Use case selesai |
| **Post Condition** | Sistem menghapus data barang |
| **Alternative Flow** | Pada langkah 3, jika barang yang akan dihapus memiliki stok maka proses hapus kategori gagal dan proses kembali ke langkah 1  Pada langkah 5, jika Admin memilih “Tidak” pada pilihan yang muncul maka proses menghapus barang dibatalkan dan proses kembali ke langkah 1 |

E. Mengelola Stok

F. Melakukan Transaksi

G. Mengelola Promo

1. Tampil Promo

|  |  |
| --- | --- |
| **Nama Use Case** | Tampil Promo |
| **Aktor** | Admin |
| **Deskripsi Singkat** | Use case menggambarkan bagaimana Admin melakukan proses pencarian dan menampilkan promo ataupun export hasil |
| **Pre Condition** | Admin memilih halaman promo dan memiliki kode promo yang akan dicari dan tampilkan |
| **Flow Of Event** | 1. Use case dimulai ketika Admin akan melihat, mengubah, atau menghapus promo tertentu  2. Sistem meminta inputan berupa Kode dan jenis promo  3. Admin menginputkan Kode promo  4. Admin memilih jenis promo Diskon Barang  5. Sistem melakukan pencarian pencarian  6. Sistem menampilkan Kode, nama barang, tanggal mulai, berakhir, diskon, dan status  7. Use case selesai |
| **Post Condition** | Sistem menampilkan Kode, nama barang, tanggal mulai, berakhir, diskon, dan status |
| **Alternative Flow** | Admin dapat melakukan Export hasil kedalam bentuk file dokumen dengan mengklik buttn Export. Sistem akan mengeksport data kategori kedalam file dokumen.  Pada langkah 5, jika sistem tidak menemukan kode atau nama barang yang dicari maka sistem akan menampilkan pesan kesalahan bahwa kode atau nama barang yang dicari tidak ditemukan  Pada langkah 4 jika admin memilih jenis Diskon Total Belanja maka data yang ditampilkan adalah kode, tanggal mulai, berakhir, belanja minimal, diskon, dan status.  Pada langkah 4 jika admin memilih jenis Diskon Buy x get y maka data yang ditampilkan adalah kode, tanggal mulai, berakhir, nama barang x, jumlah min, nama barang y, jumlah, dan status |

2. Tambah Promo

a. Tambah promo diskon barang

|  |  |
| --- | --- |
| **Nama Use Case** | Tambah Promo diskon barang |
| **Aktor** | Admin |
| **Deskripsi Singkat** | Use case menggambarkan bagaimana Admin menambahkan promo diskon barang |
| **Pre Condition** | Admin memilih halaman promo dan mengklik button Tambah, Admin memiliki data diskon yang akan ditambahkan |
| **Flow Of Event** | 1. Use case dimulai ketika sistem menampilkan form tambah diskon 2. Admin menginputkan Kode Promo 3. Sistem memvalidasi keunikan kode 4. Admin menginputkan tanggal mulainya promo 5. Admin menginputkan tanggal berakhir promo 6. Sistem memvalidasi tanggal mulai dan berakhir 7. Admin memilih jenis promo Diskon Barang 8. Admin menginput kode barang promo 9. Sistem menampilkan nama barang promo 10. Admin menginput persentase diskon 11. Sistem memvalidasi persentase diskon 12. Sistem menampilkan pesan konfirmasi tambah diskon 13. Admin memilih Ya pada pilihan yang muncul 14. Sistem mencatat data diskon 15. Use case selesai |
| **Post Condition** | Sistem mencatat data promo Diskon Barang |
| **Alternative Flow** | Pada langkah 3, 6, dan 11 jika sistem menilai bahwa inputan tidak valid mana sistem menampilkan pesan kesalahan dan meminta input ulang  Pada langkah nomor 13, jika Admin memilih “Tidak” pada pilihan yang muncul maka sistem membatalkan penambahan data promo dan proses kembali ke langkah 1 |

b. Tambah Promo Diskon Total Belanja

|  |  |
| --- | --- |
| **Nama Use Case** | Tambah Promo diskon total belanja |
| **Aktor** | Admin |
| **Deskripsi Singkat** | Use case menggambarkan bagaimana Admin menambahkan promo total belanja |
| **Pre Condition** | Admin memilih halaman promo dan mengklik button Tambah, Admin memiliki data diskon yang akan ditambahkan |
| **Flow Of Event** | 1. Use case dimulai ketika sistem menampilkan form tambah diskon 2. Admin menginputkan Kode Promo 3. Sistem memvalidasi keunikan kode 4. Admin menginputkan tanggal mulainya promo 5. Admin menginputkan tanggal berakhir promo 6. Sistem memvalidasi tanggal mulai dan berakhir 7. Admin memilih jenis promo Diskon Total Belanja 8. Admin menginput total belanja minimal 9. Admin menginput persentase diskon 10. Sistem memvalidasi persentase diskon 11. Sistem menampilkan pesan konfirmasi tambah diskon 12. Admin memilih Ya pada pilihan yang muncul 13. Sistem mencatat data diskon 14. Use case selesai |
| **Post Condition** | Sistem mencatat data promo Diskon Barang |
| **Alternative Flow** | Pada langkah 3, 6, dan 10 jika sistem menilai bahwa inputan tidak valid mana sistem menampilkan pesan kesalahan dan meminta input ulang  Pada langkah nomor 13, jika Admin memilih “Tidak” pada pilihan yang muncul maka sistem membatalkan penambahan data promo dan proses kembali ke langkah 1 |

c. Tambah Promo Buy x Get y

|  |  |
| --- | --- |
| **Nama Use Case** | Tambah Promo Buy x Get y |
| **Aktor** | Admin |
| **Deskripsi Singkat** | Use case menggambarkan bagaimana Admin menambahkan promo Buy x Get y |
| **Pre Condition** | Admin memilih halaman promo dan mengklik button Tambah, Admin memiliki data diskon yang akan ditambahkan |
| **Flow Of Event** | 1. Use case dimulai ketika sistem menampilkan form tambah diskon 2. Admin menginputkan Kode Promo 3. Sistem memvalidasi keunikan kode 4. Admin menginputkan tanggal mulainya promo 5. Admin menginputkan tanggal berakhir promo 6. Sistem memvalidasi tanggal mulai dan berakhir 7. Admin memilih jenis promo Buy x Get y 8. Admin menginput kode barang x 9. Sistem menampilkan nama barang x 10. Admin menginputkan jumlah barang minimum 11. Sistem memvalidasi jumlah 12. Admin menginputkan kode barang y 13. Sistem menampilkan nama barang y 14. Admin menginputkan jumlah barang y 15. Sistem melakukan validasi 16. Sistem menampilkan pesan konfirmasi tambah diskon 17. Admin memilih Ya pada pilihan yang muncul 18. Sistem mencatat data diskon 19. Use case selesai |
| **Post Condition** | Sistem mencatat data promo Buy x Get y |
| **Alternative Flow** | Pada langkah 3, 6, 11, dan 15 jika sistem menilai bahwa inputan tidak valid mana sistem menampilkan pesan kesalahan dan meminta input ulang  Pada langkah nomor 17, jika Admin memilih “Tidak” pada pilihan yang muncul maka sistem membatalkan penambahan data promo dan proses kembali ke langkah 1 |

3. Mengubah Promo

|  |  |
| --- | --- |
| **Nama Use Case** | Mengubah Promo |
| **Aktor** | Admin |
| **Deskripsi Singkat** | Use case menggambarkan bagaimana Admin mengubah data promo tertentu |
| **Pre Condition** | Admin berada pada halaman promo memiliki kode promo yang akan diubah |
| **Flow Of Event** | 1. Use case dimulai ketika Admin memilih promo yang akan diubah yang dapat dilakukan dengan melakukan pencarian terlebih dahulu(misal promo Diskon Barang) 2. Admin mengklik button Ubah 3. Sistem menampilkan form ubah promo beserta data lama 4. Admin dapat mengubah tanggal berakhir, dan persentase diskon 5. Sistem menampilkan pesan konfirmasi update 6. Admin memilih “Ya” pada pilihan yang ada 7. Sistem memperbarui tanggal berakhir dan atau persentase diskon dan menampilkan pesan berhasil 8. Use case selesai |
| **Post Condition** | Sistem memperbarui tanggal berakhir dan atau persentase diskon dan menampilkan pesan berhasil |
| **Alternative Flow** | Pada langkah 6, jika Admin memilih “Tidak” pada pilihan yang muncul maka proses mengubah barang dibatalkan dan proses kembali ke langkah 1  Apabila data yang diubah adalah jenis promo Diskon total belanja maka data yang dapat diubah adalah tanggal berakhir, total belanja minimal, dan persentase diskon  Apabila daya yang diubah adalah jenis promo Buy x get y maka data yang dapat diubah adalah tanggal berakhir, jumlah minumum barang x, jumlah barang y |

4. Menghapus Promo

|  |  |
| --- | --- |
| **Nama Use Case** | Menghapus Promo |
| **Aktor** | Admin |
| **Deskripsi Singkat** | Use case menggambarkan bagaimana Admin menghapus data promo tertentu |
| **Pre Condition** | Admin berada pada halaman promo dan memiliki kode promo yang akan dihapus |
| **Flow Of Event** | 1. Use case dimulai ketika Admin memilih data promo yang akan dihapus 2. Admin mengklik button Hapus 3. Sistem memastikan bahwa status promo tidak berlaku 4. Sistem menampilkan pesan konfirmasi hapus 5. Admin memilih “Ya” pada pilihan yang ada 6. Sistem menghapus data promo 7. Use case selesai |
| **Post Condition** | Sistem menghapus data barang |
| **Alternative Flow** | Pada langkah 3, jika status promo yang akan dihapus adalah berlaku maka proses hapus promo gagal dan proses kembali ke langkah 1  Pada langkah 5, jika Admin memilih “Tidak” pada pilihan yang muncul maka proses menghapus promo dibatalkan dan proses kembali ke langkah 1 |