



**RANCANG BANGUN APLIKASI REKOMENDASI PRODUK
MENGUNAKAN ALGORITMA APRIORI STUDI KASUS
TOKO AL-VEERA JEMBER**

TUGAS AKHIR

**Program Studi
S1 Sistem Informasi**

**INSTITUT BISNIS
DAN INFORMATIKA**

stikom
SURABAYA

Oleh:

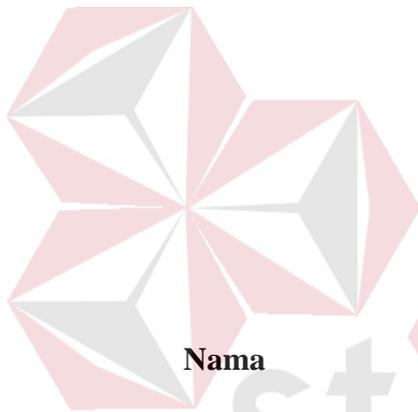
HAFIIDH MUHYIY

14410100159

**FAKULTAS TEKNOLOGI DAN INFORMATIKA
INSTITUT BISNIS DAN INFORMATIKA STIKOM SURABAYA
2019**

**RANCANG BANGUN APLIKASI REKOMENDASI PRODUK
MENGUNAKAN ALGORITMA APRIORI
STUDI KASUS TOKO AL-VEERA JEMBER
TUGAS AKHIR**

**Diajukan sebagai salah satu syarat untuk menyelesaikan
Program Sarjana Komputer**



Disusun Oleh :

Nama : Hafiidh Muhyiy

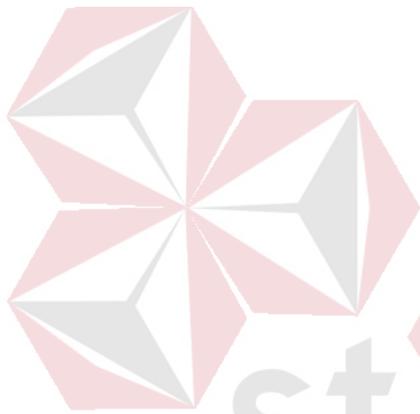
NIM : 14410100159

Program Studi : S1 (Strata Satu)

Jurusan : Sistem Informasi

**FAKULTAS TEKNOLOGI DAN INFORMATIKA
INSTITUT BISNIS DAN INFORMATIKA STIKOM SURABAYA**

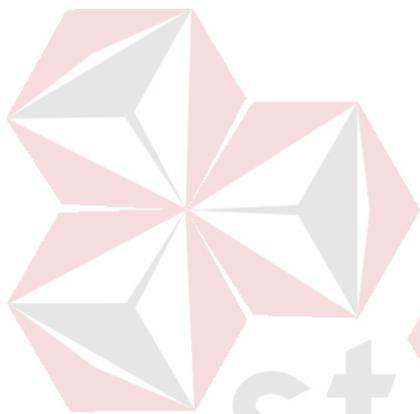
2019



INSTITUT BISNIS
DAN INFORMATIKA

stikom
SURABAYA

*Hidup hanyalah sebuah game yang tidak ada sistem checkpointnya
karena itu kita harus terus berusaha meskipun terkadang gagal*



INSTITUT BISNIS
DAN INFORMATIKA

stikom
SURABAYA

*Kupersembahkan Tugas Akhir ini untuk kedua orang tua saya yang
selalu mendukung dan memberikan semangat.*

TUGAS AKHIR
RANCANG BANGUN APLIKASI REKOMENDASI PRODUK
MENGGUNAKAN ALGORITMA APRIORI
STUDI KASUS TOKO AL-VEERA JEMBER

Dipersiapkan dan disusun oleh

Hafiidh Muhyiy

NIM: 14410100159

Telah diperiksa, diuji dan disetujui oleh Dewan Penguji

Pada : Februari 2019

Susunan Dewan Penguji

Pembimbing

I. Tri Sagirani, S.Kom., M.MT.

NIDN: 0731017601

II. Valentinus Roby Hananto, S.Kom., M.Sc.

NIDN: 0715028903

Pembahas

I. Sri Hariani Eko Wulandari S.Kom., M.MT.

NIDN: 0726017801

Tugas Akhir ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar sarjana



FAKULTAS TEKNOLOGI
DAN INFORMATIKA

stikom
SURABAYA

Dr. Jusak

Dekan Fakultas Teknologi dan Informatika

FAKULTAS TEKNOLOGI DAN INFORMATIKA
INSTITUT BISNIS DAN INFORMATIKA STIKOM SURABAYA

SURAT PERNYATAAN

PERSETUJUAN PUBLIKASI DAN KEASLIAN KARYA ILMIAH

Sebagai mahasiswa Institut Bisnis dan Informatika Stikom Surabaya, saya :

Nama : Hafidh Muhyiy

NIM : 14410100159

Program Studi : S1 Sistem Informasi

Fakultas : Fakultas Teknologi dan Informatika

Jenis Karya : Laporan Tugas Akhir

Judul Karya : **RANCANG BANGUN APLIKASI REKOMENDASI
PRODUK MENGGUNAKAN ALGORITMA APRIORI
STUDI KASUS TOKO AL-VEERA JEMBER**

Menyatakan dengan sesungguhnya bahwa:

1. Demi pengembangan Ilmu Pengetahuan, Teknologi dan Seni, saya menyetujui memberikan kepada Institut Bisnis dan Informatika Stikom Surabaya Hak Bebas Royalti Non-Eksklusif (*Non-Exclusive Royalti Free Right*) atas seluruh isi/ sebagian karya ilmiah saya tersebut di atas untuk disimpan, dialihmediakan dan dikelola dalam bentuk pangkalan data (*database*) untuk selanjutnya didistribusikan atau dipublikasikan demi kepentingan akademis dengan tetap mencantumkan nama saya sebagai penulis atau pencipta dan sebagai pemilik Hak Cipta
2. Karya tersebut di atas adalah karya asli saya, bukan plagiat baik sebagian maupun keseluruhan. Kutipan, karya atau pendapat orang lain yang ada dalam karya ilmiah ini adalah semata hanya rujukan yang dicantumkan dalam Daftar Pustaka saya
3. Apabila dikemudian hari ditemukan dan terbukti terdapat tindakan plagiat pada karya ilmiah ini, maka saya bersedia untuk menerima pencabutan terhadap gelar kesarjanaan yang telah diberikan kepada saya.

Demikian surat pernyataan ini saya buat dengan sebenarnya.

Surabaya, Februari 2019

Menyatakan
METERAI
MPEL
1000
97AFF640693449
6000
RUPIAH
Hafidh Muhyiy

NIM : 14410100159

ABSTRAK

Toko Online merupakan hal yang menjadi trend pada era saat ini karena dengan praktis dan mudahnya pelanggan dalam membeli produk yang diinginkan, tidak terkecuali dengan toko Al-Veera yang berada di Jember. Karena singkatnya jam operasional mereka dan kecilnya wilayah pemasaran mereka. Selain itu mereka membutuhkan aplikasi yang dapat memberikan rekomendasi produk berdasarkan frekuensi pembelian. Diharapkan dengan rekomendasi tersebut, pembeli dapat membeli lebih banyak.

Aplikasi rekomendasi produk menggunakan Algoritma Apriori dapat menjadi solusi untuk masalah tersebut. Algoritma Apriori merupakan suatu logika perhitungan dimana mengkombinasikan suatu produk tertentu dengan produk lainnya melalui data *history* transaksi. Dengan metode penelitian SDLC model *Waterfall*, dimana terdapat 5 tahap yaitu, *Communication*, *Planning*, *Modelling*, *Construction* dan *Deployment*.

Berdasarkan hasil uji coba Algoritma Apriori pada Aplikasi Rekomendasi Produk, yang dilakukan dalam 5 tahap dalam apriori diperoleh hasil bahwa, Algoritma Apriori berjalan sesuai skenario yang dihasilkan dari perhitungan manual. Dengan demikian dapat dikatakan bahwa aplikasi yang dibuat mampu membantu toko Al-Veera dalam mempromosikan produknya.

Keywords:

Aplikasi Rekomendasi, Algoritma Apriori

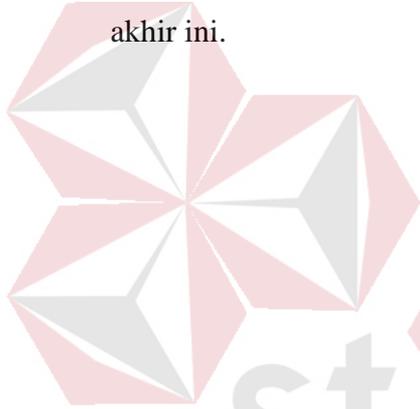
KATA PENGANTAR

Puji dan syukur atas kehadiran Allah SWT karena atas limpahan rahmat dan karunia-Nya, penulis dapat menyelesaikan laporan tugas akhir yang berjudul “Rancang Bangun Aplikasi Rekomendasi Produk Menggunakan Algoritma Apriori Studi Kasus Toko Al-veera Jember”.

Penyelesaian laporan tugas akhir ini tidak terlepas dari bantuan semua pihak yang telah memberikan banyak masukan, nasehat, kritik, saran, dan dukungan kepada penulis. Oleh karena itu, pada kesempatan ini penulis ingin menyampaikan rasa terimakasih kepada:

1. Orang tua dan Saudara-saudara yang selalu memberi dorongan motivasi dalam mengerjakan tugas akhir.
2. Bu Endah Selaku Penyelia toko busana muslim Al-Veera yang telah memberikan izin untuk melakukan penelitian disana.
3. Bapak Dr. Jusak selaku Dekan Fakultas Teknologi dan Informatika Institut Bisnis dan Informatika Stikom Surabaya.
4. Bapak Dr. Anjik Sukmaaji, S.Kom., M.Eng., selaku Ketua Program Studi S1 Sistem Informasi Institut Bisnis dan Informatika Stikom Surabaya.
5. Bu Tri Sagirani, S.Kom., M.MT. selaku pembimbing I yang telah membantu menjelaskan hal-hal terkait dengan segala yang dibutuhkan untuk tugas akhir ini.
6. Bapak Valentinus Roby Hananto, S.Kom., M.Sc. selaku pembimbing II yang juga telah membantu menjelaskan hal-hal terkait dengan segala yang dibutuhkan untuk tugas akhir ini.

7. Bu Sri Hariani Eko Wulandari S.Kom., M.MT. selaku Pembahas Tugas Akhir ini.
8. Randy Pramono yang telah membantu dalam pembangunan sistem informasi yang dibuat.
9. Teman-teman anggota grup Ra Jelas seperti Afif, Ferdi, Mamad, Rozaq, Santomo, Tama dan Reza
10. serta teman-teman lain yang tidak bisa disebut namanya yang terus membantu dalam memberi saran dan juga diskusi dalam pengerjaan tugas akhir ini.



Surabaya, Februari 2019

INSTITUT BISNIS
DAN INFORMATIKA

stikom
SURABAYA

Penulis

DAFTAR ISI

ABSTRAK	vii
KATA PENGANTAR	viii
DAFTAR ISI	x
DAFTAR GAMBAR	xiii
DAFTAR TABEL	xvi
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Tujuan	3
1.5 Manfaat	3
1.6 Sistematika Penulisan	3
BAB II LANDASAN TEORI	5
2.1 Penelitian Terdahulu	5
2.2 Aplikasi	6
2.3 <i>Online Shop</i>	7
2.3.1 Pengertian <i>Online Shop</i>	7
2.3.2 Karakteristik dan Faktor <i>Online Shop</i>	8
2.3.3 Kelebihan dan Kekurangan <i>Online Shop</i>	9
2.4 Algoritma <i>A priori</i>	12
2.5 <i>Website</i>	13
2.6 <i>Unified Modeling Language (UML)</i>	16
2.7 PHP	16
2.8 MySQL	16

2.9 HTML.....	17
2.10 <i>CodeIgniter</i>	17
2.11 Metode <i>Waterfall</i>	18
2.12 <i>Black Box Testing</i>	21
2.13 <i>User Acceptance Test (UAT)</i>	21
BAB III ANALISIS DAN PERANCANGAN SISTEM	23
3.1 <i>Communication</i>	24
3.1.1 Wawancara.....	24
3.1.2 Observasi.....	24
3.2 <i>Planning</i>	24
3.3 <i>Modelling</i>	25
3.3.1 Analisis.....	25
A. Proses Bisnis Saat Ini.....	26
B. Identifikasi Masalah.....	27
C. Analisis Kebutuhan Pengguna.....	28
D. Proses Bisnis Yang Akan Datang.....	29
E. Usecase Bisnis Diagram.....	30
F. Activity Diagram.....	31
3.3.2 Desain.....	34
A. UseCase System Diagram.....	34
B. Flow of Event.....	35
C. Desain Arsitektur.....	46
D. Sequence Diagram.....	47
E. Class Diagram.....	62
F. Desain Struktur <i>Database</i>	65
G. Desain Interface.....	71

BAB IV IMPLEMENTASI DAN EVALUASI SISTEM.....	78
4.1 Spesifikasi Sistem	78
4.2 Implementasi Apriori	79
4.2.1 Tahap 1	80
4.2.2 Tahap 2	82
4.2.3 Tahap 3	83
4.2.4 Tahap 4	84
4.2.5 Tahap 5	85
4.3 Implementasi Antarmuka	87
4.3.1 Admin	87
4.3.2 Pelanggan.....	92
4.4 Pengujian.....	97
4.4.1 Pengujian <i>Black-box</i>	97
4.4.2 Pengujian Apriori	101
4.4.3 Pengujian UAT.....	106
BAB V PENUTUP.....	112
5.1 Kesimpulan.....	112
5.2 Saran.....	112
DAFTAR PUSTAKA	113
BIODATA PENULIS	115

DAFTAR GAMBAR

Gambar 2.1 Model Waterfall	18
Gambar 3.1 Tahapan Model Penelitian.....	23
Gambar 3.2 Model Sistem Saat Ini	26
Gambar 3.3 Proses bisnis kedepannya	29
Gambar 3.4 Usecase bisnis diagram	30
Gambar 3.5 <i>Activity Diagram</i> Memesan Barang	31
Gambar 3.6 <i>Activity Diagram</i> Mengirim tagihan	32
Gambar 3.7 <i>Activity Diagram</i> Konfirmasi Pembayaran	32
Gambar 3.8 <i>Activity Diagram</i> Pembuatan Laporan.....	33
Gambar 3.9 <i>UseCase System Diagram</i>	34
Gambar 3.10 Arsitektur MVC	46
Gambar 3.11 Diagram <i>Sequence</i> Registrasi.....	47
Gambar 3.12 Diagram <i>Sequence</i> Lihat Barang.....	48
Gambar 3.13 Diagram <i>Sequence Order</i> Barang	49
Gambar 3.14 Diagram <i>Sequence</i> Konfirmasi Pembayaran.....	51
Gambar 3.15 Diagram <i>Sequence</i> Lihat <i>Order</i>	52
Gambar 3.16 Diagram <i>Sequence</i> Mengelola Data Barang.....	53
Gambar 3.17 Diagram <i>Sequence</i> Mengelola Kategori.....	56
Gambar 3.18 Diagram <i>Sequence</i> Mengelola <i>Brand</i>	57
Gambar 3.19 Diagram <i>Sequence</i> Mengelola <i>Order</i>	59
Gambar 3.20 Diagram <i>Sequence</i> Rekomendasi	60
Gambar 3.21 <i>Class Diagram Controller</i>	Error! Bookmark not defined.

Gambar 3.22 <i>Class Diagram Model</i>	65
Gambar 3.23 <i>Physical Data Model</i>	66
Gambar 3.24 Perancangan Antarmuka Beranda	71
Gambar 3.25 Perancangan Antarmuka Detail Barang	72
Gambar 3.25 Perancangan Antarmuka Keranjang Belanja.....	72
Gambar 3.27 Perancangan Antarmuka Billing Info.....	73
Gambar 3.28 Perancangan Antarmuka <i>Order</i> Sukses.....	74
Gambar 3.29 Perancangan Antarmuka Daftar <i>Order</i>	74
Gambar 3.30 Perancangan Antarmuka <i>Edit item</i> order	75
Gambar 3.31 Perancangan Antarmuka <i>Edit</i> Kategori.....	75
Gambar 3.32 Perancangan Antarmuka Tambah Kategori	76
Gambar 3.33 Perancangan Antarmuka Daftar <i>User</i>	77
Gambar 3.34 Perancangan Antarmuka Tambah <i>User</i>	77
Gambar 4.1 Hasil pencarian histori transaksi	80
Gambar 4.2 Hasil tahap 1	81
Gambar 4.3 Hasil tahap 2.....	82
Gambar 4.4 Hasil tahap 3.....	84
Gambar 4.5 Hasil Tahap 4	85
Gambar 4.6 Hasil Tahap 5	86
Gambar 4.7 Antarmuka Dashboard.....	87
Gambar 4.8 Antarmuka Daftar Order	88
Gambar 4.9 Antarmuka lihat order	89
Gambar 4.10 Antarmuka Daftar Produk	89
Gambar 4.11 Antarmuka Lihat Produk.....	90

Gambar 4.12 Antarmuka Daftar <i>Brand</i>	90
Gambar 4.13 Antarmuka Rekomendasi	91
Gambar 4.14 Antarmuka Daftar <i>User</i>	92
Gambar 4.15 Antarmuka Beranda.....	93
Gambar 4.16 Antarmuka Detail Barang.....	93
Gambar 4.17 Antarmuka <i>Order</i> Pelanggan	94
Gambar 4.18 Implementasi Antarmuka <i>Cart</i>	95
Gambar 4.19 Implementasi Antarmuka <i>Billing</i> Info.....	96
Gambar 4.20 Implementasi Antarmuka <i>Billing</i> Berhasil.....	96



DAFTAR TABEL

Tabel 3.1 Jadwal Penelitian.....	25
Tabel 3.2 Mapping Masalah, Dampak, dan Solusi	27
Tabel 3.3 Kebutuhan pengguna.....	28
Tabel 3.4 Pengelompokan <i>usecase</i>	35
Tabel 3.5 <i>Flow of event Registrasi</i>	36
Tabel 3.6 <i>Flow of event</i> Lihat Barang.....	37
Tabel 3.7 <i>Flow of event</i> Order barang.....	37
Tabel 3.8 <i>Flow of event</i> Konfirmasi Pembayaran.....	39
Tabel 3.9 <i>Flow of event</i> Lihat Order	40
Tabel 3.10 <i>Flow of event</i> Mengelola Data Barang.....	40
Tabel 3.11 <i>Flow of event</i> Mengelola Kategori.....	42
Tabel 3.12 <i>Flow of event</i> Mengelola Brand.....	43
Tabel 3.13 <i>Flow of event</i> Mengelola Order	44
Tabel 3.14 Tabel Kategori.....	66
Tabel 3.15 Tabel <i>Brand</i>	67
Tabel 3.16 Tabel Barang	67
Tabel 3.17 Tabel <i>Order Status</i>	68
Tabel 3.18 Tabel Order	69
Tabel 3.19 Tabel Status <i>Order</i>	69
Tabel 3.20 Tabel <i>User</i>	70
Tabel 4.1 Spesifikasi Perangkat Keras.....	78
Tabel 4.2 Spesifikasi Perangkat Lunak.....	79

Tabel 4.3 <i>Validation Testing</i>	97
Tabel 4.4 Tabel Data Transaksi	101
Tabel 4.5 Tabel Perbandingan tahap 1	102
Tabel 4.6 Tabel Perbandingan tahap 2	103
Tabel 4.7 Tabel Perbandingan tahap 3	104
Tabel 4.8 Tabel perbandingan tahap 4	105
Tabel 4.9 Tabel perbandingan tahap 5	105
Tabel 4.10 Kriteria dan Pertanyaan UAT	107
Tabel 4.11 Hasil Wawancara <i>User Acceptance Testing</i>	108
Tabel 4.12 Tabel Hasil Skala <i>Likert</i>	110



BAB I

PENDAHULUAN

1.1 Latar Belakang

Toko Online merupakan hal yang menjadi *trend* pada era saat ini karena dengan praktis dan mudahnya pelanggan dalam membeli produk yang diinginkan. Selain itu tanpa dikekangnya waktu transaksi atau dapat dilakukan 24 jam perhari selama 7 hari dalam seminggu dan dapat diakses dimanapun, sehingga membuat toko-toko ingin membuat toko *online*. Tidak terkecuali dengan toko Al-Veera yang berada di Jember. Toko Busana Muslim Al-Veera merupakan toko yang menyediakan busana muslim yang berlokasi di Jalan Koptu Berlian Kecamatan Sumbersari Kabupaten Jember. Al-Veera melayani pembelian pakaian busana muslim baik dalam satuan atau eceran ataupun grosir. Jam Operasional toko dimulai pukul 12.00 wib hingga pukul 22.00 wib.

Transaksi pada Al-Veera dapat dilakukan dengan dua cara yaitu dengan cara mendatangi toko dan memesan lewat chat. Setiap harinya Al-Veera melayani kurang lebih 50 pengunjung yang mendatangi toko dan 28 pembeli yang memesan via chat. Karena jangkauan pemasaran yang kurang rata-rata pengunjung toko adalah wanita yang berusia diatas 25 tahun dan berdomisili didaerah sekitar Jember, Lumajang dan Bondowoso. Adapun produk yang banyak dibeli adalah berbagai model jilbab dan busana muslim perempuan. Dengan hasil tersebut tiap bulannya toko meraih omset kurang lebih 50 juta rupiah.

Pada perkembangannya pihak toko merasa perlu untuk membangun toko online dengan prioritas pada proses memberikan rekomendasi produk kepada

pelanggan, dengan cara memanfaatkan data transaksi yang ada dan menganalisis kebiasaan pengguna dalam melakukan pembelian. Hal ini dilakukan dalam usaha meningkatkan penjualan produk pada toko Al-Veera dan merupakan salah satu layanan untuk memperkenalkan produk lain yang dimiliki pihak toko pada pelanggan. Dengan memberikan rekomendasi produk berdasarkan frekuensi pembelian, pembeli lebih tertarik berbelanja.

Dari uraian di atas Algoritma Apriori menjadi solusi terkait permasalahan yang terjadi. Algoritma Apriori merupakan aturan asosiasi data *mining* yang bertujuan untuk menemukan aturan suatu kombinasi suatu *item* dengan menghitung nilai *support* dan *confidencenya*. Dengan menggunakan Algoritma Apriori aplikasi yang dibuat akan dapat memunculkan rekomendasi produk berdasarkan perhitungan nilai *frequent* suatu produk dengan produk lainnya yang dilakukan oleh pelanggan. Hasil akhir dari pemanfaatan Algoritma Apriori menjadi suatu asosiasi produk-produk tertentu berdasarkan riwayat pembelian seluruh pelanggan.

1.2 Rumusan Masalah

Berdasarkan uraian latar belakang, maka didapatkan rumusan masalah yaitu bagaimana membuat rancang bangun aplikasi rekomendasi produk menggunakan Algoritma Apriori studi kasus toko Al-Veera Jember.

1.3 Batasan Masalah

Setelah dirumuskan permasalahannya, maka ditentukan batasan permasalahan sebagai berikut :

1. Aplikasi disusun berbasis *website* untuk mendukung toko online yang dibuat.

2. Pembayaran yang dapat dilakukan adalah dengan cara semi-manual (*Cash On Delivery* (COD) ataupun transfer via bank).
3. Tidak mencakup proses retur barang.
4. Perhitungan hanya sampai 2-*itemset*.

1.4 Tujuan

Adapun tujuan yang ingin dicapai adalah menghasilkan aplikasi rekomendasi produk menggunakan Algoritma Apriori studi kasus toko Al-Veera Jember.

1.5 Manfaat

Manfaat yang diharapkan dari aplikasi ini bagi pihak organisasi adalah sebagai berikut :

1. Mempermudah pemilik dalam melakukan merekomendasikan produknya.
2. Meningkatkan keuntungan (*profit*) toko.
3. Membantu pelanggan dalam mencari informasi barang beserta rekomendasi barang yang cocok menjadi pasangannya.

1.6 Sistematika Penulisan

BAB I : Pendahuluan

Pada bab ini akan dibahas mengenai latar belakang permasalahan yang melandasi studi kasus ini, perumusan masalah berdasarkan tujuan dan batasan masalah yang akan di bahas (dalam hal ini adalah studi kasus toko busana muslim Al-Veera, Jember), tujuan dan manfaat dari pembuatan sistem aplikasi, serta sistematika penulisan yang mendeskripsikan semuanya menjadi pengantar.

BAB II : LANDASAN TEORI

Pada bab ini berisi tentang teori-teori pendukung dalam proses pembuatan sistem aplikasi serta teori-teori pendukung dalam operasional perusahaan.

BAB III : ANALISIS DAN PERANCANGAN SISTEM

Pada bab ini berisi penjelasan mengenai penerapan solusi yang dibuat dari rancangan sistem informasi dan evaluasi terhadap sistem yang akan dirancang.

BAB IV : HASIL DAN PEMBAHASAN

Pada bab ini berisi penjelasan mengenai penerapan solusi yang dibuat dari rancangan sistem aplikasi dan evaluasi terhadap sistem yang telah dirancang serta diimplementasikan.

BAB V : PENUTUP

Pada bab ini berisi tentang kesimpulan dari pembahasan yang terkait dengan tujuan dan permasalahan yang ada, serta saran bagi pengembangnya.

BAB II

LANDASAN TEORI

2.1 Penelitian Terdahulu

Berikut adalah beberapa penelitian terdahulu terkait dengan rancang bangun aplikasi penjualan berbasis web.

- a. Penelitian dengan judul “Pembangunan Sistem Penjualan Online Pada Toko Indah Jaya Furniture Surakarta” dilakukan oleh Arip Aryanto dkk (2012) memiliki hasil sebagai berikut:
 1. Pemodelan Aplikasi belum berorientasi objek.
 2. Pembuatan aplikasi belum menggunakan bantuan framework.
 3. Web masih belum menggunakan sistem login untuk pembeli sehingga rentan terjadi keisengan pihak yang tidak bertanggung jawab.
 4. Detail produk yang ditampilkan terlalu singkat.
 5. Tampilan *website* penjualan online sudah bagus sesuai standar web *e-commerce*.
- b. Penelitian dengan judul “*Market Basket Analysis* pada perusahaan retail menggunakan algoritma apriori dan sales forecasting” dilakukan oleh Nursanti Irliana dkk (2013) memiliki hasil sebagai berikut:
 1. Peneliti melakukan perhitungan algoritma apriori menggunakan data penjualan yang hanya mencantumkan beberapa item saja.
 2. Perhitungan belum dimasukkan dalam sebuah aplikasi atau masih dilakukan dengan menggunakan *ms. Excel*

3. Algoritma apriori mampu memberikan rekomendasi yang tepat agar dapat meningkatkan penjualan perusahaan retail tersebut.
- c. Penelitian dengan judul “*Finding Frequent Item Set using Apriori Algorithm for Online Shopping (Ekart)*” dilakukan oleh K. Uma Maheswari (2018) memiliki hasil sebagai berikut:
1. Algoritma Apriori dapat diterapkan para awam untuk mengatur strateginya dalam *online shopping*.
 2. Penelitian masih dilakukan secara manual atau tidak dilakukan secara langsung kedalam suatu aplikasi toko online.

Perbedaan dari penelitian terdahulu jika dibandingkan dengan penelitian ini yaitu:

- a. Perancangan sistem akan menggunakan metode UML agar pembangunan sistem berjalan lebih mudah.
- b. Aplikasi akan menggunakan sistem login agar pembeli tidak perlu memasukkan infonya berkali-kali saat akan bertransaksi, selain itu toko akan setidaknya merasa aman dari gangguan pihak-pihak yang tidak bertanggung jawab.
- c. Menggunakan Algoritma Apriori secara langsung kedalam aplikasi agar dapat membantu memberikan rekomendasi produk.

2.2 Aplikasi

Menurut Nazrudin Safaat H (2012) Perangkat lunak aplikasi adalah suatu subkelas perangkat lunak komputer yang memanfaatkan kemampuan komputer langsung untuk melakukan suatu tugas yang diinginkan pengguna. Biasanya dibandingkan dengan perangkat lunak sistem yang mengintegrasikan berbagai

kemampuan komputer, tapi tidak secara langsung menerapkan kemampuan tersebut untuk mengerjakan suatu tugas yang menguntungkan pengguna. Contoh aplikasi diantaranya *Microsoft Powerpoint*, *Mozilla Firefox*, dll.

Aplikasi adalah program siap pakai yang dapat digunakan untuk menjalankan perintah-perintah dari pengguna aplikasi tersebut dengan tujuan mendapatkan hasil yang lebih akurat sesuai dengan tujuan pembuatan aplikasi tersebut, aplikasi mempunyai arti yaitu pemecahan masalah yang menggunakan salah satu tehnik pemrosesan data aplikasi yang biasanya berpacu pada sebuah komputansi yang diinginkan atau diharapkan maupun pemrosesan data yang diharapkan. Pengertian aplikasi secara umum adalah alat terapan yang difungsikan secara khusus dan terpadu sesuai kemampuan yang dimilikinya aplikasi merupakan suatu perangkat komputer yang siap pakai bagi user.

2.3 Online Shop

2.3.1 Pengertian Online Shop

Belanja *online* (*online shop*) merupakan proses pembelian barang/jasa oleh konsumen ke penjual *realtime*, tanpa pelayan, dan melalui internet. Toko virtual ini mengubah paradigma proses membeli barang/jasa dibatasi oleh tembok, pengecer, atau mall (Granito, 2012). Maksudnya, tak perlu harus bertemu penjual/pembeli secara langsung, tak perlu menemukan wujud 'pasar' secara fisik, namun hanya dengan menghadap layar monitor komputer, dengan koneksi internet tersambung, kita dapat melakukan transaksi jual/beli secara cepat dan nyaman. Dalam *Online shop* terdapat tingkatan-tingkatan dalam pengoperasian *online shop*, yaitu :

1. *Sebagai Supplier: yaitu sebagai* pihak yang menyediakan, menyalurkan dan memasarkan suatu produk tertentu.
2. *Sebagai Reseller: yaitu sebagai* pihak yang menjual kembali produk orang lain atau supplier dimana reseller ini berdiri sendiri atau bukan merupakan pegawai dari si supplier.
3. *Sebagai Dropshipper: yaitu sebagai* pihak yang menjadi perantara pemilik produk/barang dengan pembeli tanpa menyimpan barang terlebih dahulu.

2.3.2 Karakteristik dan Faktor *Online Shop*

Adapun karakteristik yang dimiliki oleh belanja *online (online shop)* adalah sebagai berikut:

1. Transparan dan simultan
2. Interaktif
3. Cepat
4. Terjadinya interaksi antar dua belah pihak
5. Adanya pertukaran barang, jasa, dan informasi
6. Internet merupakan media utama dalam proses perdagangan tersebut
7. Transaksi bersifat tanpa batas
8. Transaksi bersifat anonim
9. Produk digital dan non digital
10. Produk barang tak berwujud

Sedangkan faktor-faktor penting yang harus diperhatikan dalam belanja *online (online shop)* adalah :

- Kelengkapan Informasi : Dalam hal ini kelengkapan situs penjual terkait dengan detail barang yang dijual belikan harus benar-benar lengkap, sehingga konsumen mudah dalam melihat dan memilih barang dan jasa yang ditawarkan.
- Kenyamanan Pelanggan : Faktor kedua adalah kenyamanan pelanggan, dengan melakukan kegiatan online shop konsumen tidak perlu terjun langsung ke lapangan, artinya tidak perlu mencari langsung barang yang dibeli ditempat itu juga. Tidak perlu ribet kesana kemari, kepanasan, hujan, hanya duduk manis memilih-milih barang melalui smartphonenya.
- Waktu Tak Terbatas : Pembeli (konsumen) bisa melihat harga dan setelah cocok harganya tinggal memesan barang dagangan dengan waktu yang tak terbatas, kapan saja di mana saja full 24 jam.
- Kepercayaan Pelanggan : Nah faktor keempat ini benar-benar harus diperhatikan saat anda akan memulai bisnis online shop, kenyamanan dan kekecewaan akan mempengaruhi usaha anda. Semakin percaya konsumen kepada anda merupakan poin penting dalam online shop, sebaliknya penyesalan dan kekecewaan konsumen akan membuat online shop anda sepi dan semakin terpuruk.

2.3.3 Kelebihan dan Kekurangan *Online Shop*

Ada beberapa keunggulan / alasan kenapa konsumen lebih memilih belanja *online* dari pada belanja secara *offline* yaitu :

1. Mudah

Mudah karena dapat dilakukan dimana saja & kapan saja, melalui perangkat computer dimana saja yang terkoneksi dengan *internet*, termasuk dari perangkat *mobile* pribadi.

2. Murah

Alasan lain adalah belanja lewat *online* lebih murah. Ini karena biaya untuk menjual produk di internet lebih murah sehingga secara adil konsumen juga berhak atas biaya yang lebih murah jika dibandingkan dengan harga di pasar *offline*. Bahkan bila harga barang di *online* sama dengan harga di *offline* itupun masih bisa dihitung lebih murah karena ada beberapa komponen biaya lain yang sebenarnya tidak perlu dikeluarkan seperti biaya bensin, parkir, tol, makan, dll.

3. Praktis

Praktis karena tidak perlu mendorong trolley, tidak perlu antri di kasir, dan tidak perlu marah-marah dengan layanan SPG yang lama dan tidak memuaskan. Belanja online bisa dilakukan dimana saja, kapan saja, dengan alamat pengiriman yang boleh ditentukan sendiri misalnya untuk ke rumah sendiri, ke kantor atau ke rumah teman atau saudara misalnya untuk kado,

4. Efisien

Konsumen tidak perlu berlama-lama menghabiskan waktu mencari produk yang di butuhkan. Waktu menjadi lebih efisien dari pada harus berjam-jam menembus kemacetan lalu lintas.

5. Pribadi

Dunia belanja online sangat menghargai hal-hal yang bersifat pribadi. Mulai dari data pribadi yang tidak akan di-*share* ke siapapun hingga kebutuhan

produk yang di beli. Dengan belanja lewat internet konsumen tidak perlu sungkan atau tidak enak hati dilihat orang lain atau ketahuan dalam membeli produk tersebut. Produk-produk tertentu yang bersifat sangat pribadi misalnya pakaian dalam, produk kesehatan/kecantikan/kebugaran, CD, DVD, bahkan mobil mewah bisa anda dapatkan di web. Termasuk pula buku yang sangat sukses dijual secara *online*.

6. Tanpa Batas

Tidak ada batas apapun terutama dalam hal jarak ataupun waktu. Toko online melayani dalam 24 jam sehari, 7 hari seminggu, dan 365 hari setahun. Konsumen bisa membeli produk di belahan bumi manapun dan kapanpun asalkan harganya cocok sesuai dengan kondisi keuangan pribadi.

Selain keunggulan, belanja online juga mempunyai beberapa kelemahan-kelemahan, yaitu :

1. Kualitas barang yang diinginkan kadang-kadang berbeda kualitasnya dengan yang tercantum di *website*.
2. Rentan Penipuan
3. Rentan rusak atau pecah karena media pengiriman adalah pos.
4. Rentan aksi pembobolan rekening karena pembayaran dilakukan melalui Internet.
5. Marak aksi spamming karena setelah pembeli melakukan registrasi, penjual cenderung selalu mengirimkan katalog online melalui email pembeli dan hal ini cukup mengganggu *privacy*.

2.4 Algoritma *A priori*

Menurut Triyanto(2014) Algoritma *A priori* adalah suatu algoritma dasar yang diusulkan oleh Agrawal & Srikant pada tahun 1994 untuk penentuan *frequent itemsets* untuk aturan asosiasi *Boolean*. Algoritma *a priori* termasuk jenis aturan asosiasi data mining. Asosiasi data mining ini bertujuan untuk menemukan aturan suatu kombinasi item. Penting tidaknya suatu asosiasi dapat diketahui dengan dua tolak ukur, yaitu *support* dan *confidence*. *Support* adalah presentase item tersebut dalam *database*, sedangkan *confidence* adalah kuatnya hubungan antar item dalam aturan asosiasi. Terdapat tahap perhitungan dalam analisis asosiasi yaitu:

- a. Analisa pola frekuensi tinggi

Tahapan ini mencari kombinasi item yang memenuhi syarat minimum dari nilai *support* sebuah item dengan rumus:

$$\text{Support (A)} = \frac{\text{Jumlah Transaksi mengandung A}}{\text{Total Transaksi}} \dots\dots(1)$$

Sedangkan nilai support 2 item diperoleh dari rumus:

$$\text{Support (A} \cap \text{B)} = \frac{\text{Jumlah Transaksi mengandung A dan B}}{\text{Total Transaksi}} \dots\dots(2)$$

- b. Pembentukan aturan asosiatif

Setelah semua pola frekuensi tinggi ditentukan, barulah dicari aturan asosiatif yang memenuhi syarat minimum untuk *confidence* dengan menghitung aturan asosiatif A terhadap B dengan rumus :

$$\text{Confidence} = P(B | A) = \frac{\text{Jumlah Transaksi mengandung A dan B}}{\text{Jumlah Transaksi mengandung A}} \dots\dots(3)$$

Langkah-langkah dalam algoritma *a priori* sebagai berikut:

1. Menguraikan masing-masing item dalam transaksi yang ada.
2. Menghitung jumlah banyaknya pembelian untuk setiap item.
3. Menentukan batas minimum transaksi.
4. Mengkombinasikan item-item, menghitung jumlah frekuensi transaksi kombinasi tersebut dan membandingkan dengan batas minimum transaksi.
5. Setelah menemukan kombinasi item yang memenuhi syarat minimum, kemudian menghitung nilai *support* dan *confidence* dengan rumus-rumus sebagai berikut:

$$SUPPORT = \frac{\Sigma \text{item.yang.dibeli.sekaligus}}{\Sigma \text{jumlah.seluruh.transaksi}} \times 100\%$$

$$CONFIDENCE = \frac{\Sigma \text{item.yang.dibeli.sekaligus}}{\Sigma \text{jumlah.transaksi.pada.bagian.anteecedent}} \times 100\%$$

2.5 Website

Website adalah suatu sistem yang berkaitan dengan dokumen digunakan sebagai media untuk menampilkan teks, gambar, multimedia dan lainnya pada jaringan internet (Sibero, 2011).

Website dapat diartikan suatu kumpulan-kumpulan halaman yang menampilkan berbagai macam informasi teks, data, gambar diam ataupun bergerak, data animasi, suara, video maupun gabungan dari semuanya, baik itu yang bersifat statis maupun yang dinamis, yang dimana membentuk satu rangkaian bangunan yang saling berkaitan dimana masing-masing dihubungkan dengan jaringan halaman atau *hyperlink*. Jenis-jenis *website* ada 3 (tiga) macam diantaranya

1. *Website* statis adalah suatu *website* yang mempunyai halaman yang tidak berubah, yang artinya adalah untuk melakukan sebuah perubahan pada suatu halaman hanya bisa dilakukan secara manual yaitu dengan cara mengedit kode-kode yang menjadi struktur dari *website* itu sendiri.
2. *Website* dinamis adalah merupakan suatu *website* yang secara strukturnya diperuntukan untuk update sesering mungkin. Biasanya selain dimana utamanya yang bisa diakses oleh para pengguna (*user*) pada umumnya, juga telah disediakan halaman *backend* yaitu untuk mengedit konten dari *website* tersebut, contoh dari *website* dinamis seperti web berita yang didalamnya terdapat fasilitas berita, dsb.
3. *Website* Interaktif adalah suatu *website* yang didalamnya banyak terjadi interaksi antara pengunjung dengan pemilik ataupun pengunjung dengan pengunjung, contohnya *website* interaktif seperti forum dan blog. Di *website* ini para pengguna bisa berinteraksi dan juga beradu argumen mengenai apa yang menjadi pemikiran mereka.

Menurut CNET/Builder (Shawn, 2012) ada kriteria-kriteria tertentu yang menentukan *website* tersebut termasuk *website* yang baik:

1. *Usability*

Maksud dari kriteria ini adalah dapatkah seorang *user* menggunakan *website* tersebut secara efektif (*doing right things*). *Usability* sendiri terdapat 5 karakteristik:

- a. Mudah dipelajari
- b. Efisien dalam penggunaan
- c. Gampang/mudah diingat

d. Tingkat frekuensi kesalahan

e. Tingkat kepuasan pemakai

2. Sistem Navigasi

Navigasi atau alur yang mudah dipahami oleh pengunjung secara keseluruhan

3. *Graphic Design*

Pemilihan grafis, layout, warna, bentuk maupun typografi yang menarik visual pengunjung untuk menjelajahi website.

4. *Content*

Isi/konten yang bermanfaat (*content is king but without good design content is a naked king*) kecuali website tersebut adalah website eksperimental/*show off*.

5. *Compatibility*

Seberapa luas sebuah website didukung kompatibilitas peralatan yang ada, misalnya browser dengan berbagai plug-in nya (IE, Mozilla, Opera, Netscape, Lynx, Avant, Maxthon dan masih banyak lagi dengan berbagai versi dan plugin nya).

6. *Loading Time*

Waktu panggil (*loading time*), walaupun ada banyak faktor yang akan mempengaruhi waktu panggil (*loading time*) website yang akan dibuka, diantaranya: besar bandwidth/koneksi pengakses, kondisi webserver pada saat diakses, aplikasi yang digunakan dalam membangun website dan sebagainya.

7. *Functionality*

Ini akan melibatkan programmer dengan script-scriptnya, misal PHP, ASP, Java, CGI dan sebagainya, untuk menciptakan sebuah website yang dinamis, interaktif dan 'hidup' yang bisa mengajak pengunjung berkomunikasi secara langsung.

2.6 Unified Modeling Language (UML)

UML adalah ‘bahasa’ pemodelan untuk sistem atau perangkat lunak yang berparadigma ‘berorientasi objek’. Pemodelan sesungguhnya digunakan untuk penyederhaan permasalahan-permasalahan yang kompleks sehingga lebih mudah dipahami (Nugroho, 2010). Adapun Langkah-langkah dalam pemodelan UML ini adalah:

1. Membuat daftar bisnis proses yang ada.
2. Memetakan usecase untuk setiap bisnis proses.
3. Membuat *activity* diagram berdasarkan use case diagram yang telah dipetakan.
4. Membuat *sequence* diagram berdasarkan activity diagram.
5. Membuat rancangan *user interface* berdasarkan skenario usecase yang telah dibuat.
6. Membuat *class* diagram.
7. Mulai membangun sistem.

2.7 PHP

PHP (*PHP: Hypertext Preprocessor*) adalah bahasa *server-side scripting* yang menyatu dengan HTML untuk membuat halaman web yang dinamis. Karena merupakan *server-side scripting* maka sintaks dan perintah-perintah PHP akan dieksekusi di server kemudian hasilnya dikirimkan ke browser dalam format HTML (Arief, 2012).

2.8 MySQL

MySQL adalah sebuah perangkat lunak system manajemen basis data SQL atau dikenal dengan DBMS(*Database Management System*), database ini

multithread dan *multiuser*. MySQL AB membuat MySQL tersedia perangkat lunak gratis dibawah lisesni GNU *General Public Licensed* (GPL), tetapi mereka menjual dibawah lisensi komersial untuk kasus-kasus tertentu.

2.9 HTML

HTML adalah bahasa markup untuk menyebarkan informasi pada Web. Ketika merancang HTML, ide ini diambil dari *Standard Generalized Markup Language* (SGML). Walaupun HTML tidak dengan mudah dapat dipahami kebanyakan orang, ketika diterbitkan penggunaannya menjadi jelas. HTTP adalah protokol komunikasi stateless yang berbasiskan pada TCP yang awalnya digunakan untuk mengambil kembali file-file HTML dari server Web ketika dirancang pada tahun 1991. *Uniform Resource Locator* (URL). URL tersusun atas tiga bagian yaitu format *transfer*, nama *host*, *path* berkas dokumen (Simarmata, 2010).

2.10 CodeIgniter

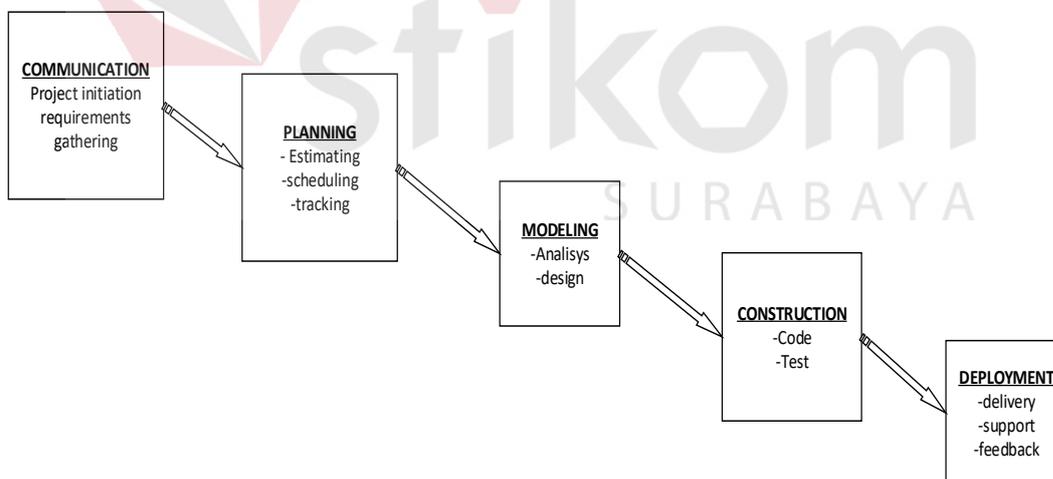
CodeIgniter adalah sebuah web application network yang bersifat open source yang digunakan untuk membangun aplikasi php dinamis. CodeIgniter menjadi sebuah *framework* PHP dengan model MVC (*Model*, *View*, *Controller*) untuk membangun website dinamis dengan menggunakan PHP yang dapat mempercepat pengembang untuk membuat sebuah aplikasi web. Selain ringan dan cepat, *CodeIgniter* juga memiliki dokumentasi yang super lengkap disertai dengan contoh implementasi kodenya. Dokumentasi yang lengkap inilah yang menjadi salah satu alasan kuat mengapa banyak orang memilih CodeIgniter sebagai framework pilihannya. Karena kelebihan-kelebihan yang dimiliki oleh *CodeIgniter*, pembuat PHP Rasmus Lerdorf memuji *CodeIgniter* di frOSCon

dengan mengatakan bahwa dia menyukai CodeIgniter karena “*it is faster, lighter and the least like a framework.*”.

2.11 Metode Waterfall

Model *waterfall* adalah model klasik yang bersifat sistematis, berurutan dalam membangun *software*. Nama model ini sebenarnya adalah “*Linear Sequential Model*”. Dalam buku yang ditulis Pressman, ia menjelaskan Model ini termasuk ke dalam model generik pada rekayasa perangkat lunak dan pertama kali diperkenalkan oleh Winston Royce sekitar tahun 1970 sehingga sering dianggap kuno, tetapi merupakan model yang paling banyak dipakai dalam *Software Engineering* (SE). Model ini melakukan pendekatan secara sistematis dan berurutan. Disebut dengan *waterfall* karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan (Pressman, 2015).

Fase-fase dalam *Waterfall Model* menurut referensi Pressman :



Gambar 2.1 Model Waterfall

A. *Communication (Project Initiation & Requirements Gathering)*

Sebelum memulai pekerjaan yang bersifat teknis, sangat diperlukan adanya komunikasi dengan customer demi memahami dan mencapai tujuan yang

ingin dicapai. Hasil dari komunikasi tersebut adalah inisialisasi proyek, seperti menganalisis permasalahan yang dihadapi dan mengumpulkan data-data yang diperlukan, serta membantu mendefinisikan fitur dan fungsi *software*. Pengumpulan data-data tambahan bisa juga diambil dari jurnal, artikel, dan internet.

B. *Planning (Estimating, Scheduling, Tracking)*

Tahap berikutnya adalah tahapan perencanaan yang menjelaskan tentang estimasi tugas-tugas teknis yang akan dilakukan, resiko resiko yang dapat terjadi, sumber daya yang diperlukan dalam membuat sistem, produk kerja yang ingin dihasilkan, penjadwalan kerja yang akan dilaksanakan, dan *tracking* proses pengerjaan sistem.

C. *Modeling (Analysis & Design)*

Tahapan ini adalah tahap perancangan dan permodelan arsitektur sistem yang berfokus pada perancangan struktur data, arsitektur *software*, tampilan *interface*, dan algoritma program. Tujuannya untuk lebih memahami gambaran besar dari apa yang akan dikerjakan.

D. *Construction (Code & Test)*

Tahapan *Construction* ini merupakan proses penerjemahan bentuk desain menjadi kode atau bentuk/bahasa yang dapat dibaca oleh mesin. Setelah pengkodean selesai, dilakukan pengujian terhadap sistem dan juga kode yang sudah dibuat. Tujuannya untuk menemukan kesalahan yang mungkin terjadi untuk nantinya diperbaiki.

E. *Deployment (Delivery, Support, Feedback)*

Tahapan *Deployment* merupakan tahapan implementasi *software* ke *customer*, pemeliharaan *software* secara berkala, perbaikan *software*, evaluasi *software*, dan pengembangan *software* berdasarkan umpan balik yang diberikan agar sistem dapat tetap berjalan dan berkembang sesuai dengan fungsinya. (Pressman, 2015).

Kapan sebaiknya metode *waterfall* digunakan? Ada teori-teori yang menyimpulkan beberapa hal, yaitu :

- a. Ketika semua persyaratan yang diajukan sudah dipahami dengan baik pada awal pengembangan program.
- b. Definisi produk bersifat stabil dan tidak ada perubahan yang dilakukan saat pengembangan untuk alasan apapun. Oleh karena itu, teknologi yang digunakan juga harus sudah dipahami dengan baik.
- c. Menghasilkan produk baru, atau produk dengan versi baru. Sebenarnya, jika menghasilkan produk dengan versi baru maka itu sudah termasuk *incremental development*, yang setiap tahapannya sama dengan metode *waterfall* kemudian diulang-ulang.
- d. Pembentukan ulang produk yang sudah ada ke dalam platform baru.

Dengan demikian, metode *waterfall* dianggap pendekatan yang lebih cocok digunakan untuk proyek pembuatan sistem baru dan juga pengembangan *software* dengan tingkat resiko yang kecil serta waktu pengembangan yang cukup lama. Tetapi salah satu kelemahan paling mendasar adalah menyamakan pengembangan *hardware* dan *software* dengan meniadakan perubahan saat pengembangan.

2.12 *Black Box Testing*

Menurut Roger S. Pressman (2015) *Black Box Testing* atau Pengujian Kotak Hitam atau juga disebut *Behavioral Testing*, berfokus pada persyaratan fungsional dari perangkat lunak. Artinya, teknik *Black Box Testing* memungkinkan untuk mendapatkan set kondisi masukan yang sepenuhnya akan melaksanakan semua persyaratan fungsional untuk suatu program. *Black Box Testing* bukan merupakan alternatif dari pengujian *White Box Testing*. Sebaliknya, *Black Box Testing* adalah pendekatan komplementer yang mungkin untuk mengungkap kelas yang berbeda dari kesalahan daripada metode *White Box Testing*. *Black Box Testing* mencoba untuk menemukan kesalahan dalam kategori berikut.

- a. Fungsi tidak benar atau hilang.
- b. Kesalahan *interface* atau antarmuka.
- c. Kesalahan dalam struktur data atau akses *database* eksternal.
- d. Kesalahan kinerja atau perilaku.
- e. Kesalahan inisialisasi dan terminasi.

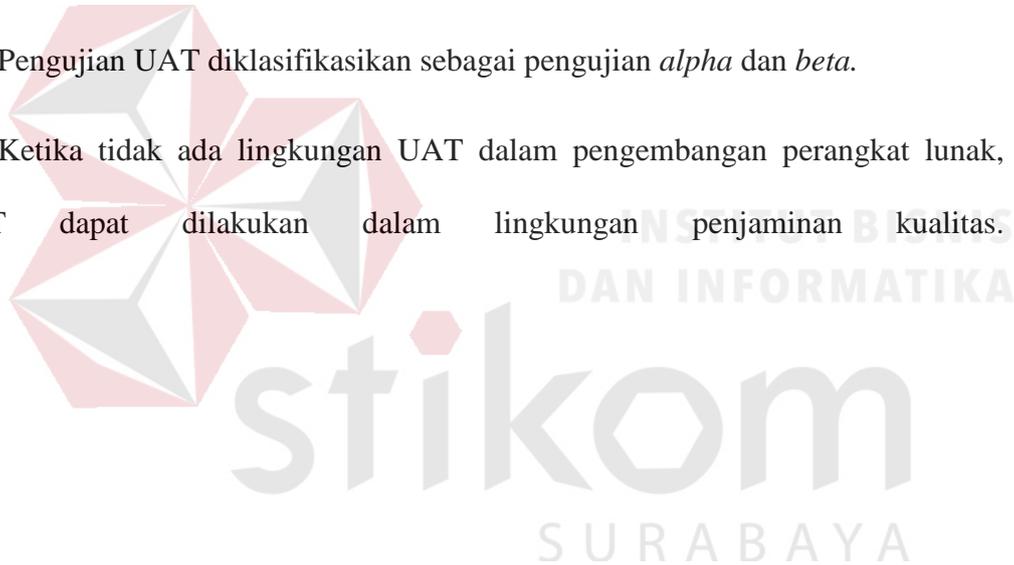
2.13 *User Acceptance Test (UAT)*

UAT atau *user acceptance test* biasa dikaitkan dengan pengujian pada pengguna terkait adanya sebuah sistem informasi baru yang tengah di perkenalkan kedalam suatu perusahaan atau organisasi. Tujuan utama dari UAT adalah memastikan apakah sistem baru yang dibuat dapat melakukan fungsinya dengan baik dan apakah sudah memenuhi kebutuhan pengguna (Hambling dan Goethem, 2013). UAT biasa dilakukan setelah pengujian sistem. Pengujian UAT dilakukan

oleh tiap pengguna yang terlibat dengan sistem yang dibuat. Ada beberapa poin yang harus diperhatikan dalam membuat UAT (softwaretestinghelp, 2018), yaitu:

1. UAT tidak sekedar menguji halaman, kolom atau tombol. Karena semua hal itu seharusnya sudah di uji sebelum melakukan pengujian UAT.
2. Pengujian ini adalah tentang entitas yang merupakan elemen utama dalam suatu bisnis.
3. UAT juga dapat menjadi sebuah pengujian inti, yang berarti dengan pengujian ini kita juga dapat menemukan *bug* ketika melakukan pengujian.
4. Pengujian UAT diklasifikasikan sebagai pengujian *alpha* dan *beta*.

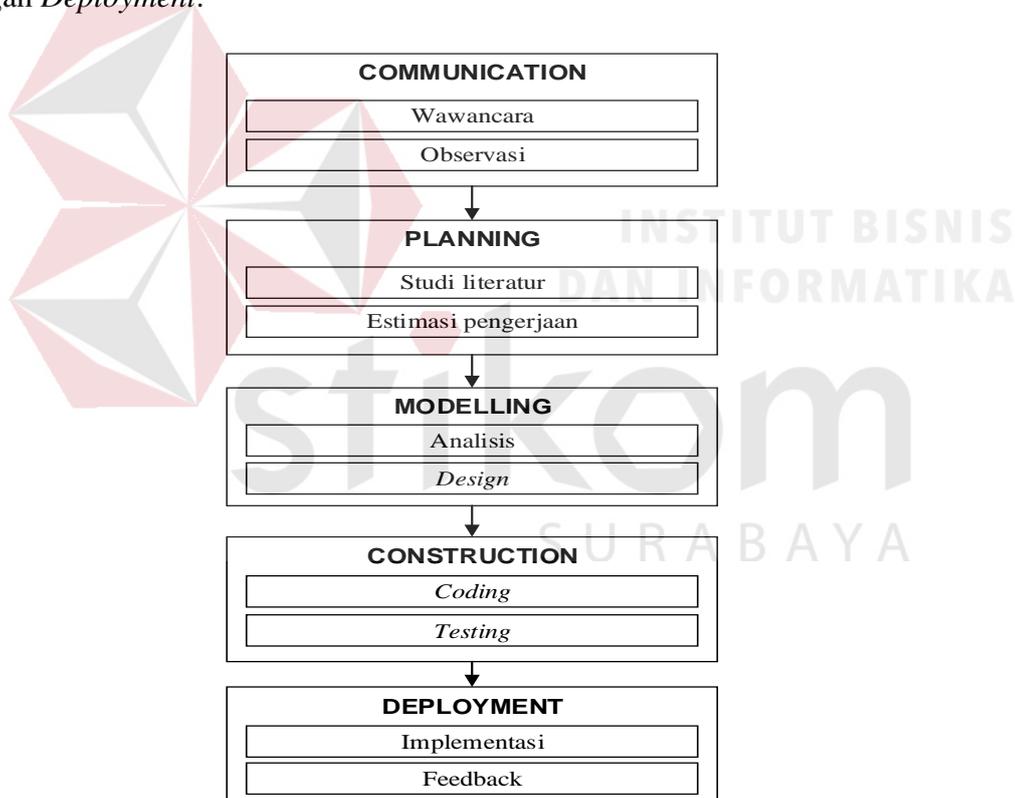
Ketika tidak ada lingkungan UAT dalam pengembangan perangkat lunak, UAT dapat dilakukan dalam lingkungan penjaminan kualitas.



BAB III

ANALISIS DAN PERANCANGAN SISTEM

Pada bab ini berisi penjelasan mengenai penerapan solusi yang dibuat dari rancang bangun sistem informasi pemasaran menggunakan algoritma apriori pada Toko Busana Muslim Al-Veera di Jember Jawa Timur. Metode penelitian pada pengembangan sistem ini adalah menggunakan metode *System Development Life Cycle* (SDLC) dengan model *Waterfall* yaitu dimulai dari *Communication* sampai dengan *Deployment*.



Gambar 3.1 Tahapan Model Penelitian

3.1 Communication

Tahapan awal pada pengembangan perangkat lunak ini yaitu berfokus pada proses pengumpulan informasi dari masing-masing pihak yang terlibat (*stakeholder*). Berikut merupakan metode pengumpulan data yang digunakan:

3.1.1 Wawancara

Wawancara pada Toko Busana muslim Al-veera ini dilakukan untuk memperoleh informasi mengenai sistem yang berlaku ataupun informasi lain yang sekiranya dapat membantu dalam pengembangan program. Hasil dari wawancara dengan bu Endah selaku pemilik toko adalah permasalahan pemasaran yang dihadapi karena masih bersifat manual sehingga jangkauan pemasaran tidak lebih dari kota Jember, Lumajang dan sekitarnya. Selain itu kurang efektifnya jam kerja toko yang jangka waktunya lebih cepat daripada toko yang biasanya. Dan terakhir pemilik membutuhkan sebuah katalog yang dapat membantu dalam proses bisnis dalam transaksi tanpa ke toko.

3.1.2 Observasi

Observasi dilakukan untuk pengembangan program dengan mengumpulkan dan mengamati secara langsung terhadap data yang akan di gunakan serta dapat mengetahui lebih jelas proses bisnis yang ada pada Toko busana muslim al-veera. Dengan dilakukannya observasi maka dapat mengetahui lebih jelas lagi proses bisnis dari toko busana muslim al-veera dalam transaksi pembelian.

3.2 Planning

Pada tahapan *planning* akan dilakukan proses studi literatur yang dilakukan untuk mencari solusi terkait hasil wawancara dan observasi dan

membuat *jobdesk* mengenai aktivitas-aktivitas yang dilakukan dalam mencapai tujuan penelitian ini. Aktivitas yang dimaksud adalah *estimating* (estimasi waktu), *scheduling* (penjadwalan), dan *tracking* (penjabaran tugas) sehingga dapat dihasilkan *roadmap* yang dapat dijadikan panduan dalam aktivitas pengembangan perangkat lunak. *Roadmap* tersebut dapat dilihat pada Tabel 3.1 berikut.

Tabel 3.1 Jadwal Penelitian

No	Kegiatan	Bulan 1				Bulan 2				Bulan 3				Bulan 4			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1.	Wawancara	■															
2.	Observasi		■														
3.	Studi Literatur			■													
4.	Estimasi Pengerjaan				■												
5.	Analisis					■	■										
6.	Desain							■	■								
7.	Coding									■	■	■					
8.	Testing												■	■			
9.	Deployment															■	■

3.3 Modelling

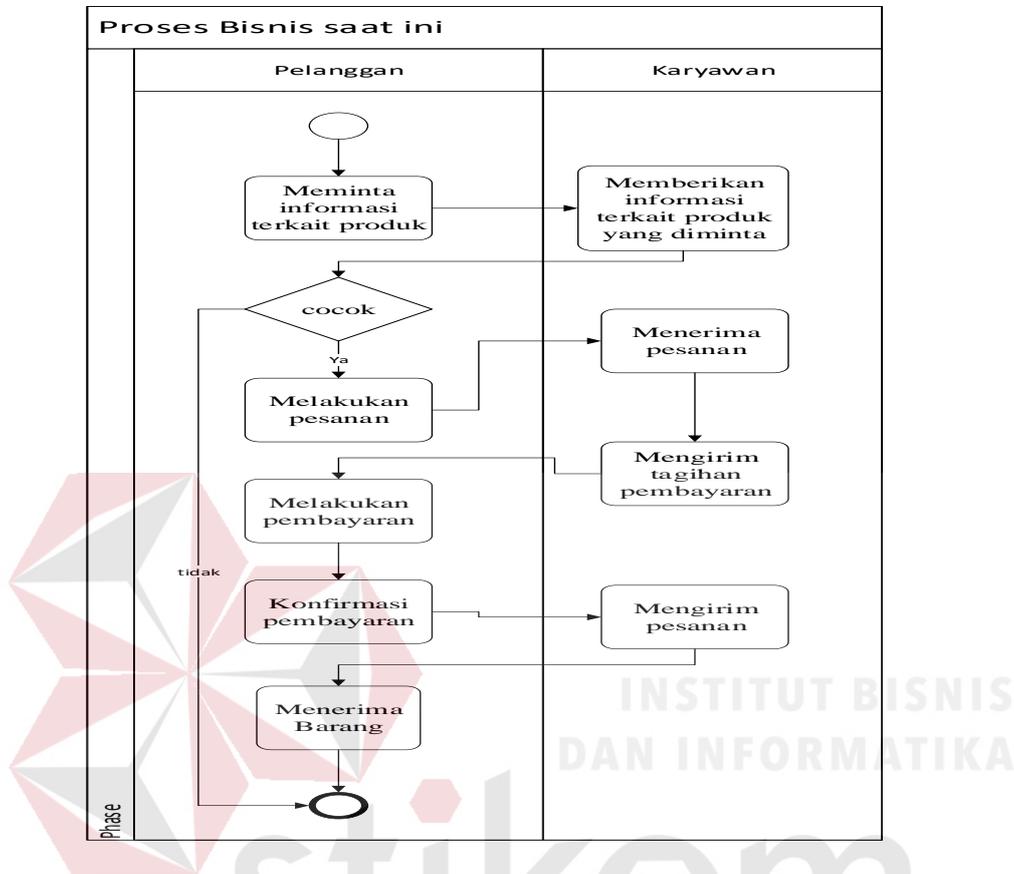
Pada tahapan ini terdiri dari dua aktivitas yaitu analisis dan desain dimana kedua aktivitas tersebut dilakukan pada komponen-komponen sistem informasi sesuai kebutuhan fungsional. Proses analisis menjadi langkah awal sebelum dilakukannya proses desain dalam merencanakan model sistem.

3.3.1 Analisis

Analisis yang dilakukan meliputi analisis proses bisnis saat ini, analisis masalah, usulan proses bisnis kedepannya, *usecase* bisnis, *activity* diagram, dan analisis kebutuhan pengguna. Berikut adalah hasil analisis yang telah dilakukan.

A. Proses Bisnis Saat Ini

1. Proses Pendaftaran



Gambar 3.2 Model Sistem Saat Ini

Proses bisnis saat ini dimulai dari pelanggan melakukan *chat* untuk menanyakan apakah terdapat produk baru yang bagus, stok dan juga harga barang. Apabila sesuai dengan keinginannya pelanggan melakukan pemesanan barang apabila tidak maka transaksi selesai. Setelah menerima pemesanan karyawan mengirim ulang detail pesanan beserta total harga yang harus dibayar dan nomer rekening tujuan agar pelanggan dapat mengkonfirmasi pesanan telah dibayar dan dapat segera membayar pesanan. Setelah pembayaran selesai pelanggan mengirim bukti pembayaran ke

karyawan agar karyawan dapat mengkonfirmasi pembayaran dan segera mengirim pesan. Transaksi selesai saat pelanggan telah menerima barang.

B. Identifikasi Masalah

Berdasarkan hasil wawancara dan analisis proses bisnis yang saat ini dilakukan, ditemukan bahwa Toko Al-veera belum sepenuhnya memanfaatkan teknologi informasi sehingga terjadi permasalahan-permasalahan yang ditunjukkan pada Tabel 3.2 berikut:

Tabel 3.2 Mapping Masalah, Dampak, dan Solusi

Masalah	Dampak	Solusi
<ul style="list-style-type: none"> - Pelanggan harus menanyakan detail produk. - karyawan harus segera mengecek stok barang agar segera dapat membalas pertanyaan pelanggan 	<ul style="list-style-type: none"> transaksi yang berjalan menjadi tidak efisien 	<ul style="list-style-type: none"> Menyediakan katalog yang berisi detail produk secara <i>real-time</i>.
Tidak adanya sarana pemasaran yang dapat membantu toko dalam memasarkan produk.	Pembeli hanya berasal didaerah yang dekat toko saja.	Menyediakan sarana pemasaran yang dapat diakses pelanggan dimana saja.

Masalah	Dampak	Solusi
Rekapitulasi tidak dapat langsung diberikan kepada pimpinan agar segera disimpan	Rekap penjualan sering hilang atau terselip sehingga karyawan harus membuat ulang	Memberikan sistem yang dapat langsung mengirimkan hasil rekap langsung setelah transaksi selesai.

C. Analisis Kebutuhan Pengguna

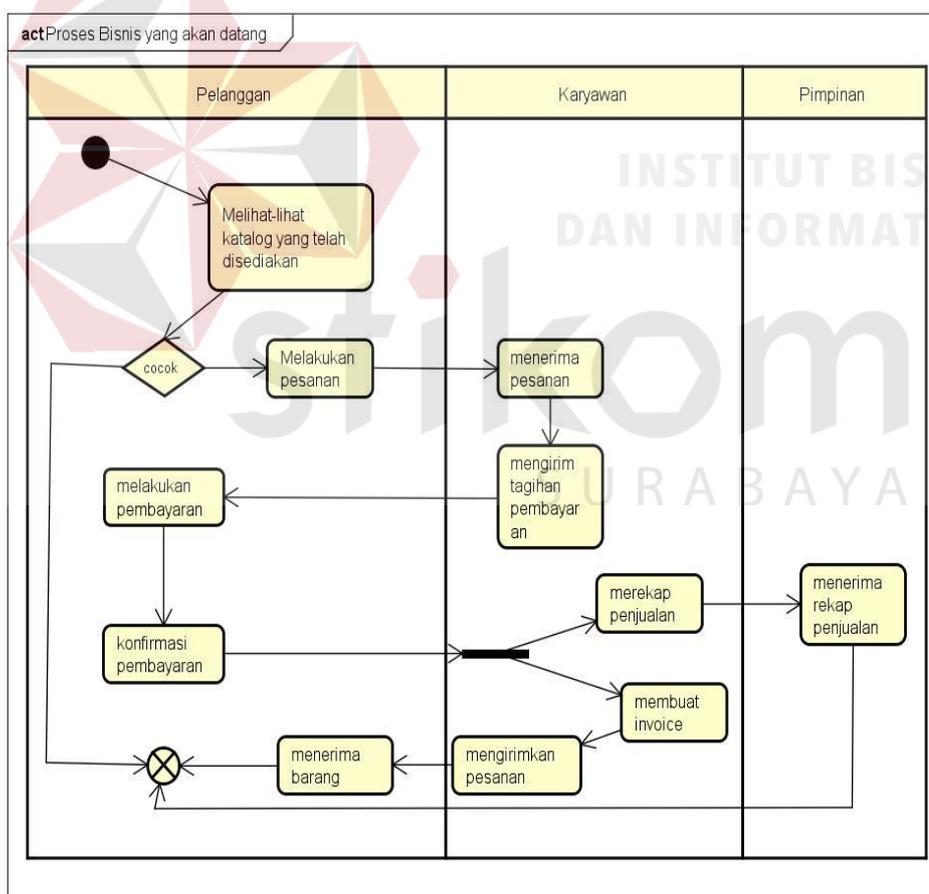
Berikut merupakan gambaran kebutuhan pengguna yang didapat dari menganalisa *Activity Diagram*.

Tabel 3.3 Kebutuhan pengguna

Jenis user	Kebutuhan/requirements
Pemilik	<ol style="list-style-type: none"> 1. Fungsi manajemen akun <ol style="list-style-type: none"> 1.1 Pembuatan/<i>create user admin</i> baru. 1.2 <i>Edit user admin</i>. 2. Fungsi penjualan produk <ol style="list-style-type: none"> 1.1 Unggah produk. 1.2 <i>Preview</i> produk. 3. Laporan penjualan produk.
Karyawan	<ol style="list-style-type: none"> 1. Fungsi manajemen produk <ol style="list-style-type: none"> 1.1 <i>Homepage</i> yang sederhana dengan mengedepankan gambar dari produk yang direkomendasi berdasarkan testimoni oleh <i>member</i>. 1.2 <i>Edit</i> produk. 1.3 Notifikasi pembelian produk. 1.4 Notifikasi produk telah diterima. 1.5 Notifikasi diskusi dan ulasan produk.
Member	<ol style="list-style-type: none"> 1. Fungsi akun <i>member</i> <ol style="list-style-type: none"> 1.1 Pendaftaran <i>member</i> baru.

Jenis user	Kebutuhan/requirements
	1.2 <i>Profile member.</i> 2. Fungsi pembelian produk 2.1 Pencarian produk. 2.2 <i>Preview</i> produk. 2.3 Rekomendasi produk 2.4 Keranjang belanja dan pembayaran. 2.5 Bagian konfirmasi produk diterima. 2.6 Bagian diskusi dan ulasan produk. 3. <i>Customer service.</i>

D. Proses Bisnis Yang Akan Datang

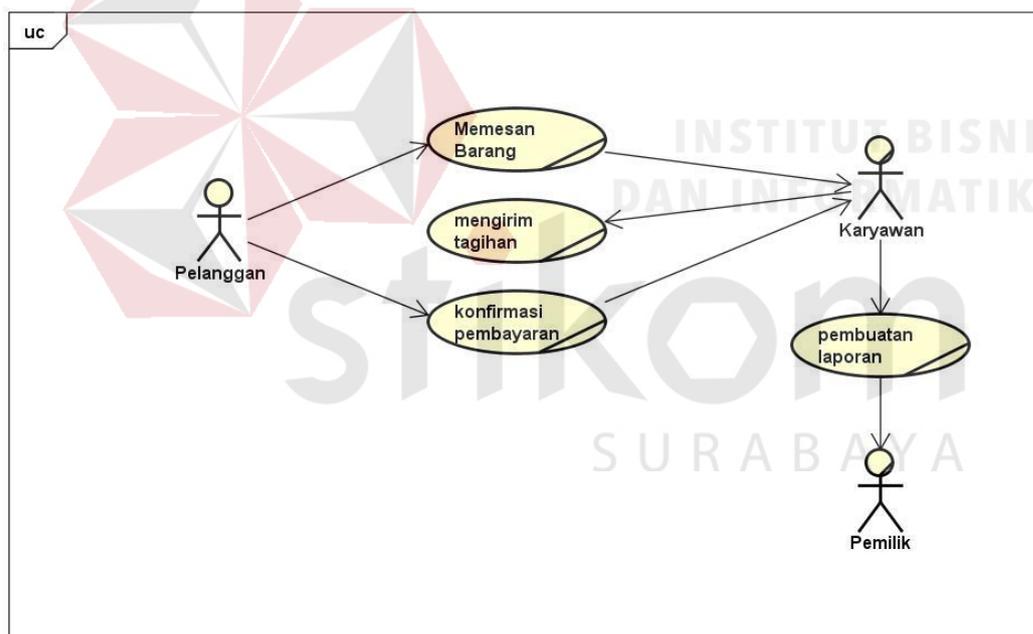


Gambar 3.3 Proses bisnis kedepannya

Untuk proses kedepannya pelanggan hanya perlu melihat katalog-katalog yang telah disediakan secara detail sehingga pelanggan tidak perlu menunggu

jawaban karyawan seputar produk-produk yang ada. Apabila pelanggan merasa ada yang cocok dengan barang yang tersedia pelanggan dapat langsung melakukan pemesanan. Setelah itu karyawan mengecek detail pemesanan dan mengkonfirmasi pemesanan. Setelah pemesanan dikonfirmasi pelanggan dapat langsung melakukan pembayaran untuk pemesanan yang dilakukan dan segera melakukan konfirmasi pembayaran. Setelah pembayaran dikonfirmasi karyawan segera mencetak *invoice* pesanan dan secara bersamaan mengirim rekap penjualan kepada pimpinan. Kemudian menyiapkan pesanan untuk segera dikirim.

E. Usecase Bisnis Diagram



Gambar 3.4 Usecase bisnis diagram

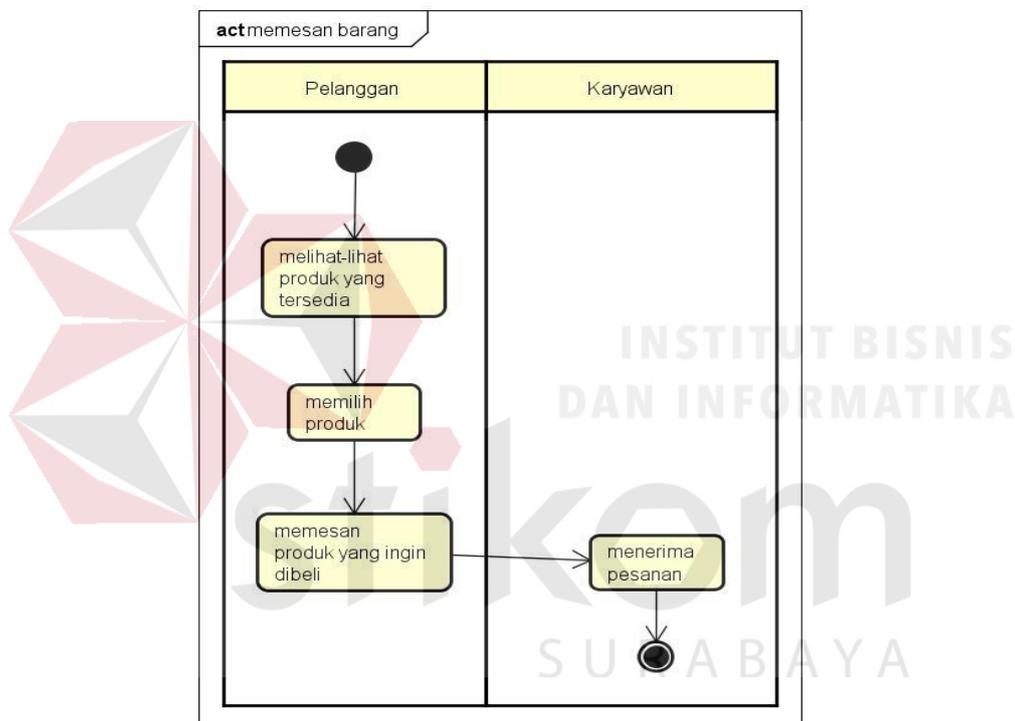
Usecase Bisnis merupakan gambaran singkat terkait proses bisnis yang akan dilakukan kedepannya dengan memecah proses bisnis kedalam poin-poin tertentu. Dalam Gambar 3.4 dijelaskan bahwa akan ada 4 proses bisnis utama yang menjadi usecase yaitu memesan barang, mengirim tagihan, konfirmasi pembayaran dan pembuatan laporan. Terdapat 3 aktor dalam *usecase* diatas yang

dibagi menjadi 2 tipe yaitu 2 aktor bisnis/internal yaitu karyawan dan pemilik serta 1 aktor eksternal yaitu pelanggan.

F. Activity Diagram

Activity Diagram atau diagram aktivitas merupakan penjabaran dari proses-proses yang dilakukan pada tiap usecase yang tertera pada usecase bisnis diagram.

F.1 Activity Diagram Memesan Barang

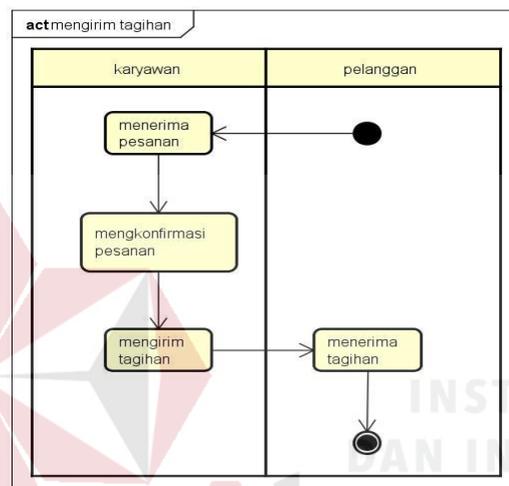


Gambar 3.5 *Activity Diagram* Memesan Barang

Pada Gambar 3.5 menjelaskan bahwa *usecase* ini proses dimulai pada saat pelanggan melihat-lihat produk yang tersedia lewat katalog yang diberikan. Proses selanjutnya pelanggan memilih produk yang akan dibeli. Apabila pelanggan selesai memilih pelanggan melakukan pemesanan. Proses selesai saat karyawan menerima pesanan produk dari pelanggan.

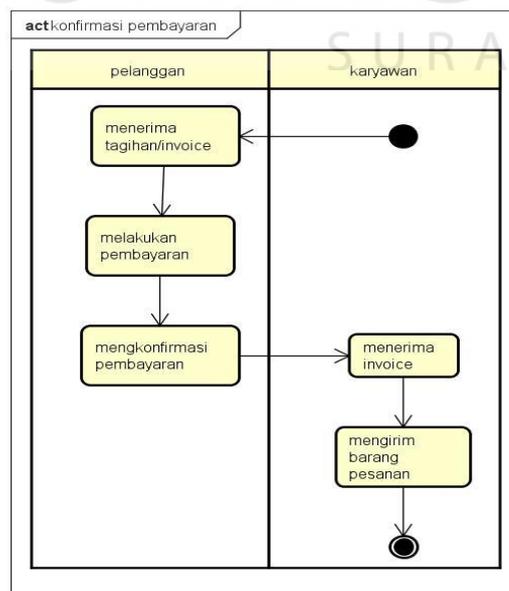
F.2 Activity Diagram Mengirim Tagihan

Pada Gambar 3.6 menjelaskan bahwa *usecase* ini proses dimulai pada saat karyawan telah menerima pesanan dari pelanggan. Kemudian karyawan melakukan konfirmasi pesanan yang dilanjutkan dengan pengiriman tagihan kepada pelanggan. Proses selesai saat pelanggan menerima tagihan.



Gambar 3.6 Activity Diagram Mengirim tagihan

F.3 Activity Diagram Konfirmasi Pembayaran

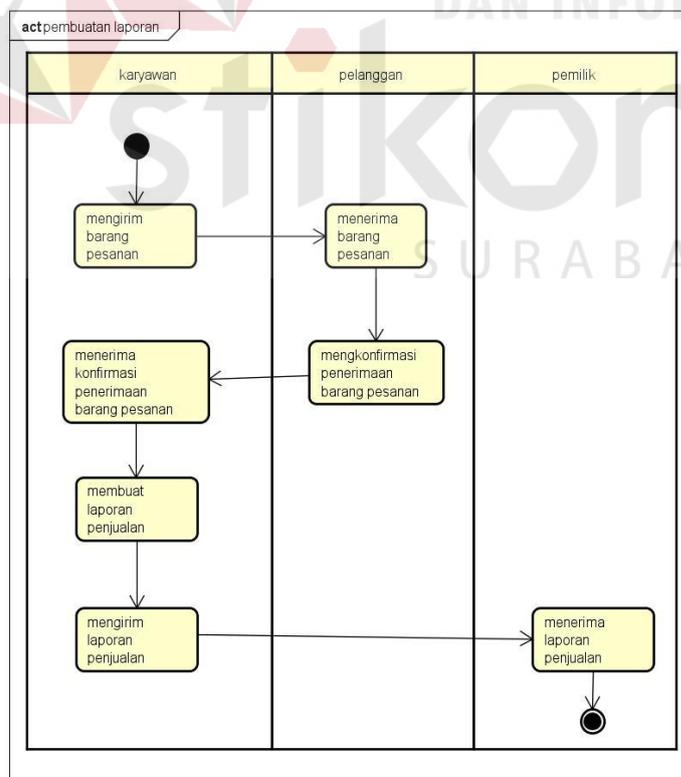


Gambar 3.7 Activity Diagram Konfirmasi Pembayaran

Pada Gambar 3.7 menjelaskan bahwa usecase ini proses dimulai pada saat pelanggan menerima tagihan yang diberikan oleh karyawan. Proses selanjutnya pelanggan melakukan pembayaran dan selanjutnya melakukan konfirmasi pembayaran. Proses selesai saat karyawan menerima konfirmasi pembayaran dari pelanggan.

F.4 Activity Diagram Pembuatan Laporan

Pada Gambar 3.8 menjelaskan bahwa usecase ini proses dimulai pada saat karyawan mengirim barang pesanan pelanggan. Apabila pelanggan telah menerima barang pesannya, Pelanggan harus melakukan konfirmasi penerimaan barang. Setelah karyawan menerima konfirmasi penerimaan barang, Karyawan membuat laporan penjualan agar segera untuk dikirim ke pemilik. Proses berakhir saat pemilik menerima laporan penjualan.



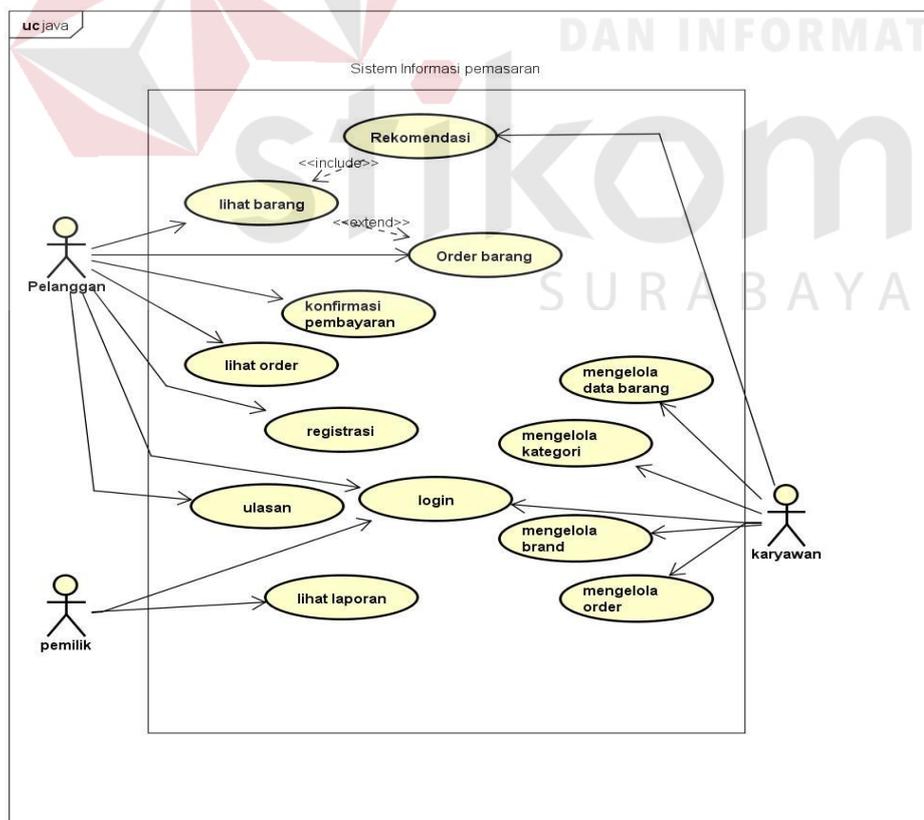
Gambar 3.8 Activity Diagram Pembuatan Laporan

3.3.2 Desain

Pada tahap ini mengembangkan spesifikasi untuk perangkat keras, perangkat lunak, orang, jaringan dan sumber data dan produk informasi yang akan memenuhi persyaratan fungsional dari sistem informasi bisnis yang diusulkan serta mengembangkan model logis dari sistem yang baru.

A. UseCase System Diagram

Gambaran deskripsi dari interaksi yang terjadi antara pengguna dan sistem dapat digambarkan menggunakan diagram *usecase*. Pada tahap ini dilakukan sebuah pemodelan untuk aktivitas disertai interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat. Berikut merupakan gambaran use case dari aplikasi rekomendasi produk:



Gambar 3.9 UseCase System Diagram

Pada Gambar 3.9 menjelaskan *usecase sistem diagram* dimana terdapat 3 aktor yaitu pelanggan, karyawan dan pemilik. Serta terdapat 11 usecase yang dilakukan oleh tiap-tiap actor yang terhubung yaitu lihat barang, order barang, konfirmasi pembayaran, registrasi, login, lihat laporan, mengelola data barang mengelola kategori, mengelola *brand*, dan mengelola order. Adapun pengelompokan usecase berdasarkan aktor dapat dilihat pada Tabel 3.4 berikut.

Tabel 3.4 Pengelompokan *usecase*

Kode Use Case	Use Case	Aktor
UC-01	Registrasi	Pelanggan
UC-02	Lihat Barang	Pelanggan
UC-03	<i>Order</i> Barang	Pelanggan
UC-04	Konfirmasi Pembayaran	Pelanggan
UC-05	Lihat <i>Order</i>	Pelanggan
UC-06	Mengelola Data Barang	Karyawan
UC-07	Mengelola Kategori	Karyawan
UC-08	Mengelola <i>Brand</i>	Karyawan
UC-09	Mengelola <i>Order</i>	Karyawan
UC-10	Lihat Laporan	Pemilik
UC-11	Rekomendasi	Karyawan

B. Flow of Event

Flow of event merupakan suatu elemen yang mendokumentasikan bisnis dari setiap *usecase* yang ada. Pada elemen ini juga akan dijelaskan bagaimana interaksi antara aktor dengan sistem. Berikut penjelasan dari masing-masing *flow of event* :

B.1 Registrasi

Tabel 3.5 *Flow of event Registrasi*

<i>UseCase Name</i>	Registrasi
<i>Brief Description</i>	Menjelaskan bagaimana pelanggan akan melakukan registrasi
<i>Actor</i>	Pelanggan
<i>Related Use Case</i>	-
<i>Stakeholder</i>	Karyawan
<i>Precondition</i>	-
<i>Postcondition</i>	1. Pelanggan berhasil terdaftar dalam sistem
<i>Basic Flow</i>	<ol style="list-style-type: none"> 1. Pelanggan membuka halaman utama sistem 2. Pelanggan menekan menu Registrasi 3. Pelanggan mengisi data yang diperlukan. 4. Pelanggan menekan tombol daftar.
<i>Exception</i>	-
<i>Subflow</i>	-

Pada Tabel 3.5 menjelaskan alur registrasi pelanggan. Pelanggan memilih menu login pada halaman utama sistem lalu memilih menu Registrasi Akun Baru. Pada halaman registrasi akun baru pelanggan memasukkan data sesuai yang dibutuhkan lalu menekan tombol daftar.

B.2 Lihat Barang

Pada Tabel 3.6 berikut menjelaskan alur melihat barang. Pelanggan harus membuka halaman utama sistem lalu memilih barang yang ingin dilihat detailnya.

Tabel 3.6 *Flow of event* Lihat Barang

<i>UseCase Name</i>	Lihat Barang
<i>Brief Description</i>	Menjelaskan bagaimana pelanggan melihat barang
<i>Actor</i>	Pelanggan
<i>Related Use Case</i>	-
<i>Stakeholder</i>	Karyawan
<i>Precondition</i>	1. Pelanggan membuka peramban dan memasukkan alamat website sistem.
<i>Postcondition</i>	1. Detail barang yang diinginkan tampil
<i>Basic Flow</i>	1. Pelanggan membuka halaman utama sistem 2. Pelanggan memilih/klik barang yang ingin dilihat
<i>Exception</i>	-
<i>Subflow</i>	-

B.3 Order Barang

Tabel 3.7 ini menjelaskan alur *order* barang. Saat *order* barang pelanggan bisa membeli satu barang saja atau dapat memilih lebih dari satu barang.

Tabel 3.7 *Flow of event* Order barang

<i>UseCase Name</i>	Order Barang
<i>Brief Description</i>	Menjelaskan bagaimana pelanggan akan melakukan <i>order</i> barang
<i>Actor</i>	Pelanggan
<i>Related Use Case</i>	Lihat Barang
<i>Stakeholder</i>	Karyawan

<i>Precondition</i>	1. Pelanggan berhasil masuk dalam sistem
<i>Postcondition</i>	1. <i>Order</i> pelanggan dicatat oleh sistem
<i>Basic Flow</i>	<ol style="list-style-type: none"> 1. Use case dimulai ketika pelanggan memilih barang yang akan dibeli. 2. Pelanggan memilih ukuran dan memasukkan jumlah barang yang ingin dibeli 3. Pelanggan menekan tombol Beli Sekarang 4. Pelanggan memilih kurir yang ingin digunakan, memilih provinsi tujuan, memilih kabupaten atau kota tujuan, dan mengisi alamat lengkap tujuan. 5. Pelanggan menekan tombol Proses 6. Pelanggan memilih paket dari ekspedisi yang digunakan 7. Pelanggan menekan tombol Konfirmasi Pesanan
<i>Exception</i>	-
<i>Subflow</i>	<p>a. Pelanggan membeli lebih dari 1 barang</p> <ol style="list-style-type: none"> 1. Pelanggan menekan tombol Tambah ke Keranjang Belanja 2. Pelanggan memilih barang dan memilih ukuran serta memasukkan jumlah barang lagi hingga semua barang yang dibutuhkan pelanggan dimasukkan kedalam keranjang 3. Pelanggan menekan menu <i>Cart</i> 4. Pelanggan memilih kurir yang ingin digunakan, memilih provinsi tujuan, memilih kabupaten atau kota tujuan, dan mengisi alamat lengkap tujuan. 5. Pelanggan menekan tombol Proses

	6. Pelanggan memilih paket dari ekspedisi yang digunakan
	7. Pelanggan menekan tombol Konfirmasi Pesanan

B.4 Konfirmasi Pembayaran

Tabel 3.8 berikut menjelaskan alur konfirmasi pembayaran. Setelah pelanggan berhasil masuk kedalam sistem, pelanggan memilih atau klik pada menu *order*, lalu klik tombol Konfirmasi Pembayaran pada *order* yang dimaksud.

Tabel 3.8 *Flow of event* Konfirmasi Pembayaran

<i>Use Case Name</i>	Konfirmasi Pembayaran
<i>Brief Description</i>	Menjelaskan bagaimana pelanggan akan melakukan konfirmasi pembayaran
<i>Actor</i>	Pelanggan
<i>Related Use Case</i>	-
<i>Stakeholder</i>	Karyawan
<i>Precondition</i>	1. Pelanggan berhasil masuk kedalam sistem 2. Pelanggan sudah melakukan <i>order</i>
<i>Postcondition</i>	2. Status <i>order</i> pelanggan menjadi Konfirmasi Pembayaran
<i>Basic Flow</i>	1. Pelanggan memilih atau klik tombol <i>order</i> 2. Pelanggan menekan tombol Konfirmasi Pembayaran pada <i>order</i> yang telah dilakukan
<i>Exception</i>	-
<i>Subflow</i>	-

B.5 Lihat *Order*

Tabel 3.9 ini menjelaskan alur lihat *order*. Pada halaman utama sistem pelanggan dapat melakukan klik pada menu *order* untuk melihat histori singkat *order* yang pernah dilakukan.

Tabel 3.9 *Flow of event* Lihat Order

Use Case Name	Lihat <i>Order</i>
Brief Description	Menjelaskan bagaimana pelanggan melihat riwayat <i>order</i> secara singkat yang pernah dilakukan
Actor	Pelanggan
Related Use Case	-
Stakeholder	Karyawan
Precondition	1. Pelanggan berhasil masuk dalam sistem
Postcondition	1. Sistem menampilkan histori singkat <i>order</i> pelanggan
Basic Flow	1. Pelanggan memilih atau klik pada menu <i>order</i>
Exception	-
Subflow	-

B.6 Mengelola Data Barang

Tabel 3.10 ini menjelaskan alur mengelola barang. Dalam *usecase* ini terdapat 4 sub-flow yaitu menambah, melihat, *edit* dan hapus barang.

Tabel 3.10 *Flow of event* Mengelola Data Barang

Use Case Name	Mengelola Data Barang
Brief Description	Menjelaskan bagaimana karyawan mengelola data barang

Actor	Karyawan
Related Use Case	-
Stakeholder	Karyawan
Precondition	1. Karyawan berhasil masuk dalam sistem
Postcondition	1. Produk baru tersimpan dalam sistem dan atau 2. Sistem menampilkan detail produk dan atau 3. Detail produk berubah sesuai masukan pengguna dan atau 4. Produk terhapus dari sistem
Basic Flow	1. Karyawan memilih atau klik menu Produk 2. Karyawan menekan atau klik tombol tambah barang 3. Karyawan memasukkan data sesuai dengan barang yang ingin dimasukkan 4. Karyawan menekan atau klik tombol Submit
Exception	-
Subflow	a. Melihat barang 1. Karyawan menekan atau klik tombol Lihat pada barang yang ingin dilihat b. <i>Edit</i> Barang 1. Karyawan menekan atau klik tombol <i>edit</i> pada barang yang ingin di <i>edit</i> 2. Karyawan mengganti data barang sesuai yang diinginkan 3. Karyawan menekan atau klik tombol submit c. Hapus Barang 1. Karyawan menekan atau klik tombol delete pada barang yang ingin dihapus

B.7 Mengelola Kategori

Tabel 3.11 ini menjelaskan alur mengelola kategori. Dalam *usecase* ini terdapat 4 *sub-flow* yaitu menambah, *edit* dan hapus kategori.

Tabel 3.11 *Flow of event* Mengelola Kategori

<i>UseCase Name</i>	Mengelola Kategori
<i>Brief Description</i>	Menjelaskan bagaimana karyawan mengelola kategori barang
<i>Actor</i>	Karyawan
<i>Related Use Case</i>	-
<i>Stakeholder</i>	Karyawan
<i>Precondition</i>	1. Karyawan berhasil masuk dalam sistem
<i>Postcondition</i>	1. Kategori baru tersimpan dalam sistem dan atau 2. Sistem menampilkan kategori dan atau 3. Kategori berubah sesuai masukan pengguna dan atau 4. Kategori terhapus dari sistem
<i>Basic Flow</i>	1. Karyawan memilih atau klik menu Kategori 2. Karyawan menekan atau klik tombol tambah kategori 3. Karyawan memasukkan data sesuai dengan kategori yang ingin dimasukkan 4. Karyawan menekan atau klik tombol Submit
<i>Exception</i>	-
<i>Subflow</i>	a. <i>Edit</i> Kategori 1. Karyawan menekan atau klik tombol <i>edit</i> pada kategori yang ingin di <i>edit</i> 2. Karyawan mengganti kategori sesuai yang diinginkan

	<ol style="list-style-type: none"> 3. Karyawan menekan atau klik tombol submit <p>b. Hapus Kategori</p> <ol style="list-style-type: none"> 1. Karyawan menekan atau klik tombol delete pada kategori yang ingin dihapus
--	---

B.8 Mengelola *Brand*

Tabel 3.12 ini menjelaskan alur mengelola *brand*. Dalam *usecase* ini terdapat 3 *sub-flow* yaitu menambah, *edit* dan hapus kategori.

Tabel 3.12 *Flow of event* Mengelola Brand

<i>UseCase Name</i>	Mengelola <i>Brand</i>
<i>Brief Description</i>	Menjelaskan bagaimana karyawan mengelola <i>brand</i> barang
<i>Actor</i>	Karyawan
<i>Related Use Case</i>	-
<i>Stakeholder</i>	Karyawan
<i>Precondition</i>	1. Karyawan berhasil masuk dalam sistem
<i>Postcondition</i>	<ol style="list-style-type: none"> 1. <i>Brand</i> baru tersimpan dalam sistem dan atau 2. Sistem menampilkan <i>brand</i> dan atau 3. <i>Brand</i> berubah sesuai masukan pengguna dan atau 4. <i>Brand</i> terhapus dari sistem
<i>Basic Flow</i>	<ol style="list-style-type: none"> 1. Karyawan memilih atau klik menu <i>Brand</i> 2. Karyawan menekan atau klik tombol tambah <i>brand</i> 3. Karyawan memasukkan data sesuai dengan <i>brand</i> yang ingin dimasukkan 4. Karyawan menekan atau klik tombol Submit

<i>Exception</i>	-
<i>Subflow</i>	<p>a. <i>Edit Brand</i></p> <ol style="list-style-type: none"> 1. Karyawan menekan atau klik tombol <i>edit</i> pada <i>brand</i> yang ingin di <i>edit</i> 2. Karyawan mengganti <i>brand</i> sesuai yang diinginkan 3. Karyawan menekan atau klik tombol submit <p>b. <i>Hapus Brand</i></p> <ol style="list-style-type: none"> 1. Karyawan menekan atau klik tombol delete pada <i>brand</i> yang ingin dihapus

B.9 Mengelola *Order*

Tabel 3.13 ini menjelaskan alur mengelola *order* Dalam *usecase* ini terdapat 2 *sub-flow* yaitu menambah, *edit* dan hapus kategori.

Tabel 3.13 *Flow of event* Mengelola *Order*

<i>UseCase Name</i>	Mengelola <i>Order</i>
<i>Brief Description</i>	Menjelaskan bagaimana karyawan mengelola <i>order</i> barang
<i>Actor</i>	Karyawan
<i>Related Use Case</i>	-
<i>Stakeholder</i>	Karyawan
<i>Precondition</i>	1. Karyawan berhasil masuk dalam sistem
<i>Postcondition</i>	<ol style="list-style-type: none"> 1. Sistem menampilkan detail <i>order</i> dan atau 2. Data barang dalam <i>Order</i> berubah sesuai dengan masukan pengguna dan atau 3. Status <i>order</i> berubah sesuai dengan masukan pengguna.

<i>Basic Flow</i>	<ol style="list-style-type: none"> 1. Karyawan memilih atau klik menu <i>Order</i> 2. Karyawan menekan atau klik tombol lihat pada <i>order</i> yang ingin dilihat
<i>Exception</i>	-
<i>Subflow</i>	<ol style="list-style-type: none"> a. <i>Edit Barang Order</i> <ol style="list-style-type: none"> 1. Karyawan menekan atau klik tombol <i>edit</i> pada barang pada <i>order</i> yang ingin di <i>edit</i> 2. Karyawan mengganti data barang yang <i>diorder</i> sesuai yang diinginkan 3. Karyawan menekan atau klik tombol submit b. <i>Mengganti Status Order</i> <ol style="list-style-type: none"> 1. Karyawan menekan atau klik tombol status yang diinginkan

B.10 Rekomendasi

Tabel 3.14 ini menjelaskan alur rekomendasi. Dalam *usecase* ini terdapat 1 *sub-flow* yaitu edit rekomendasi.

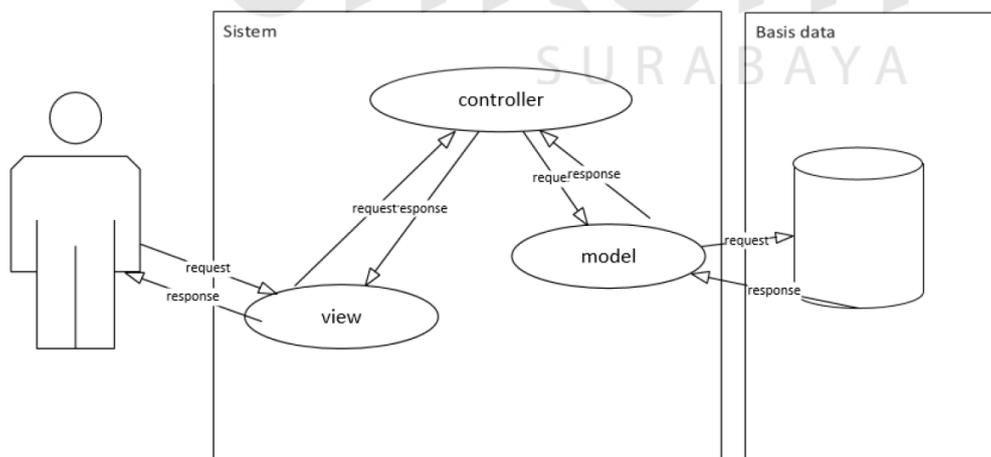
Tabel 3.14 *Flow of event* Rekomendasi

<i>UseCase Name</i>	Rekomendasi
<i>Brief Description</i>	Menjelaskan bagaimana alur algoritma apriori dalam sistem
<i>Actor</i>	Karyawan
<i>Related Use Case</i>	Lihat Barang
<i>Stakeholder</i>	Karyawan
<i>Precondition</i>	1. Karyawan berhasil masuk dalam sistem
<i>Postcondition</i>	1. Nilai Confidence pasangan barang muncul

<i>Basic Flow</i>	1. Karyawan memilih menu rekomendasi
<i>Exception</i>	-
<i>Subflow</i>	<p>a. <i>Edit Rekomendasi</i></p> <ol style="list-style-type: none"> 1. Karyawan menekan atau klik tombol ubah batasan 2. Karyawan mengisi nilai batasan yang baru 3. Karyawan mengklik tombol <i>submit</i>

C. Desain Arsitektur

Desain Arsitektur yang akan digunakan dalam pembuatan sistem ini menggunakan konsep *ModelViewController* (MVC). Pada konsep MVC ketika pengguna menginginkan suatu data, maka sebuah *controller* akan bertanggung jawab untuk menangani permintaan data. Fungsi yang dijalankan akan memanggil model yang bertanggung jawab untuk mengambil data langsung dari *database*. Setelah data didapatkan oleh *controller* maka data akan diteruskan menuju *view* untuk ditampilkan kepada pengguna. Apabila digambarkan adalah sebagai berikut



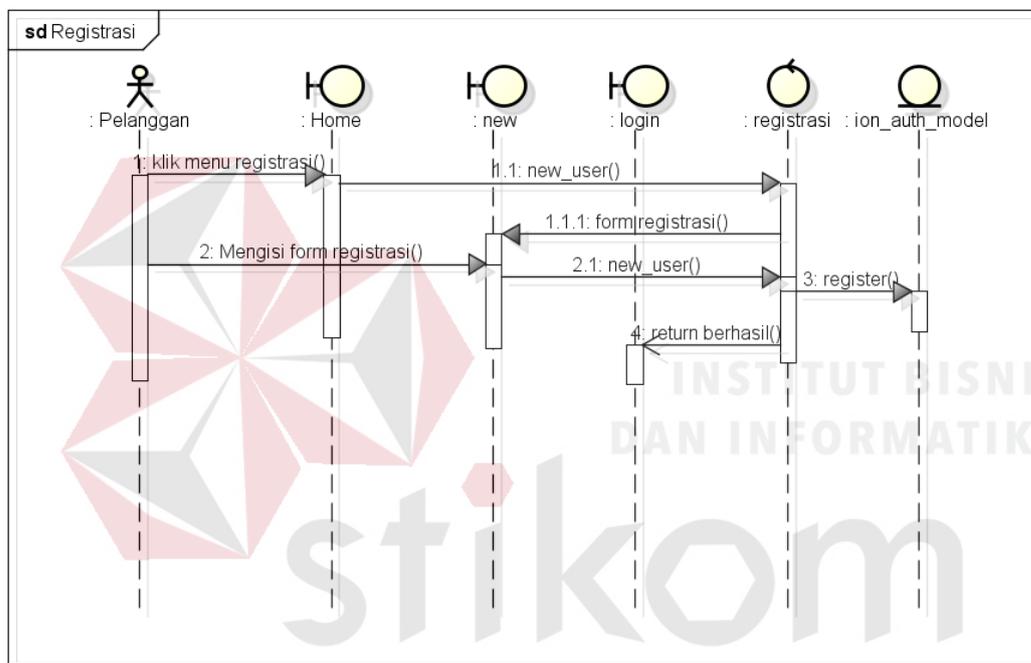
Gambar 3.10 Arsitektur MVC

D. Sequence Diagram

Sequence Diagram merupakan suatu diagram yang menggambarkan interaksi antar objek yang diatur dalam urutan waktu serta mengindikasikan komunikasi diantara objek-objek tersebut berdasar *flow of event* dari *usecase*.

Berikut penjelasan dari masing-masing *sequence diagram*:

D.1 Registrasi

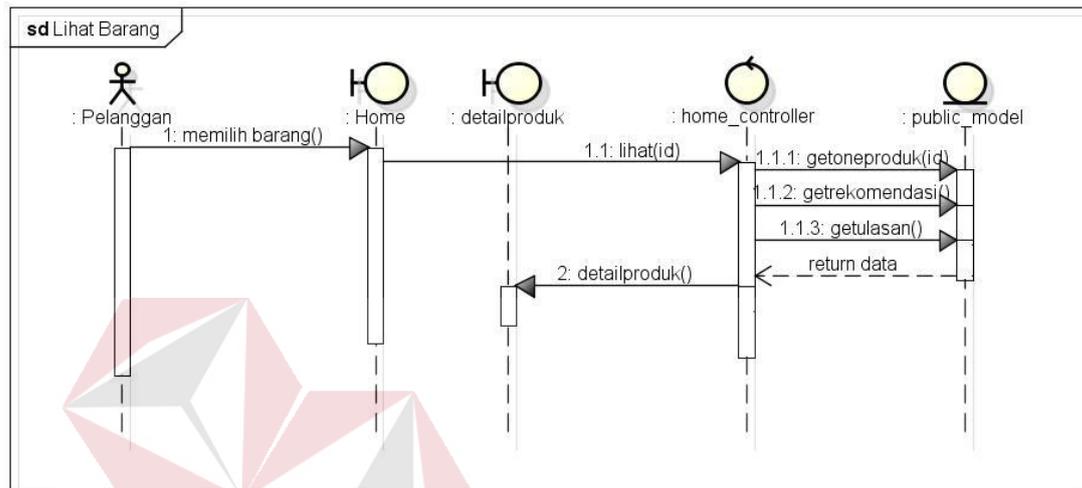


Gambar 3.11 Diagram *Sequence* Registrasi

Pada Gambar 3.11 dijelaskan bahwa diagram *sequence* Registrasi diawali dari pelanggan memilih atau klik menu registrasi pada halaman home yang akan memanggil fungsi dari *controller* registrasi yang bernama *new_user*. Selanjutnya *controller* registrasi akan menampilkan halaman baru yaitu sebuah form yang bernama form registrasi. Setelah halaman form registrasi muncul maka pelanggan atau pengguna mengisi form registrasi tersebut. Data pada form tersebut akan digunakan untuk membuat pengguna baru dengan menggunakan fungsi *new_user* pada *controller*.

Ketika data diterima oleh *controller*, *controller* akan memasukkan data tersebut kedalam basis data menggunakan fungsi *register* dan selanjutnya akan memunculkan halaman login serta pesan bahwa akun berberhasil dibuat.

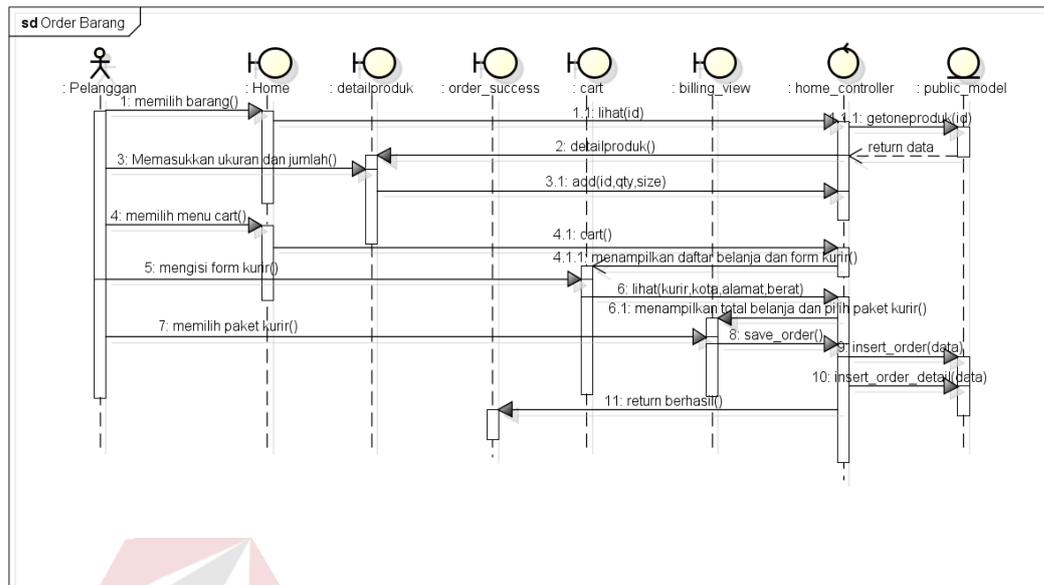
D.2 Lihat Barang



Gambar 3.12 Diagram *Sequence* Lihat Barang

Pada Gambar 3.12 dijelaskan bahwa diagram *sequence* lihat barang memiliki beberapa tahap. Diawali dari pelanggan memilih barang pada halaman *home*, dimana dari halaman *home* memanggil fungsi *lihat* pada *controller*. Dari *controller* akan mengambil data pada basisdata dengan memanggil fungsi *getoneproduk*, *getrekomendasi* dan *getulasan* dimana fungsi tersebut mengembalikan data kepada *controller*. Data dari database tersebut ditampilkan kepada pengguna dengan membuka halaman baru yaitu detail produk, produk rekomendasi dan ulasan produk.

D.3 Order Barang



Gambar 3.13 Diagram *Sequence Order Barang*

Pada Gambar 3.13 dijelaskan bahwa diagram *sequence order* barang dimulai dari pelanggan memilih barang pada halaman *home*. Setelah memilih barang maka halaman tersebut akan memanggil fungsi lihat pada *controller* untuk mengambil data pada pada basis data. Pengambilan data ini menggunakan fungsi *getoneproduk* pada *controller*. Fungsi *getoneproduk* akan mengembalikan data kepada *controller*. Setelah data dari basis data dimiliki maka *controller* akan menampilkan data tersebut pada halaman detail produk.

Pada halaman detail produk pelanggan dapat memasukkan ukuran dan jumlah yang diinginkan. Setelah selesai maka halaman tersebut akan memanggil fungsi *add* untuk menyimpan data pesanan tersebut sementara pada *cart*. Proses memasukkan ukuran dan jumlah ini dapat dilakukan kembali oleh pelanggan. Pelanggan dapat memasukkan produk sesuai yang diinginkan.

Setelah selesai memasukkan barang ke dalam *cart*, pelanggan dapat memilih menu *cart* untuk melanjutkan proses pemesanan. Untuk menampilkan halaman *cart* halaman home akan memanggil fungsi *cart* pada controller untuk menampilkan data pesanan yang telah dimasukkan sebelumnya dan menampilkan form kurir. Data pesanan tersebut meliputi nama barang, ukuran, dan harga. Harga yang dicantumkan pada halaman *cart* ini akan didiskon secara otomatis oleh sistem. Pemotongan harga ini berdasarkan, apakah pelanggan yang bersangkutan memiliki diskon pada brand tersebut atau tidak.

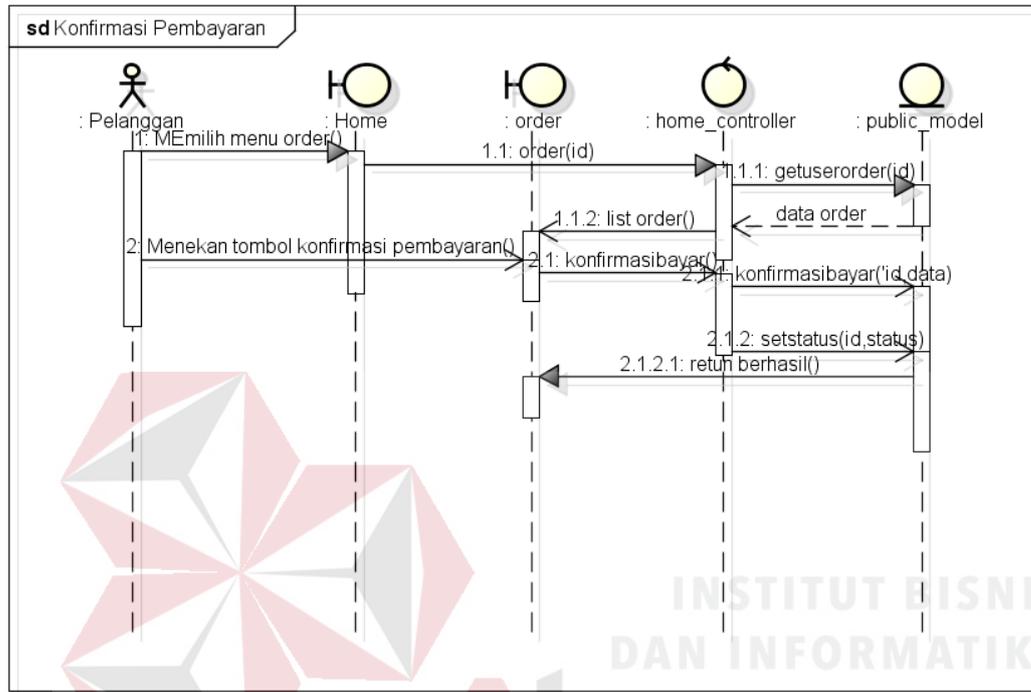
Selain data pesanan halaman tersebut juga menampilkan form kurir. Form kurir ini terdiri dari kolom kurir, kolom provinsi, kolom kabupaten, dan kolom alamat lengkap. Data atau pilihan pada kolom provinsi dan kolom kabupaten ini didapat dari integrasi dengan API RajaOngkir. Kolom ini berkaitan satu dengan yang lain, jika pengguna memilih salah satu provinsi maka pada kolom kabupaten hanya akan menampilkan kabupaten yang ada pada provinsi tersebut.

Pada halaman ini pelanggan diharuskan mengisi form kurir tersebut. Setelah diisi maka data dari form tersebut akan diproses oleh *controller*. Controller akan menampilkan data pesanan kembali, paket kurir dan total pembayaran. Paket kurir tersebut didapat dari integrasi dengan API RajaOngkir. Pada bagian total pembayaran dapat dilihat berapa total belanja ditambah dengan biaya kirim sesuai dengan paket kurir.

Setelah pelanggan memilih paket kurir halaman *billing_view* akan memanggil fungsi *save_order* pada *controller*. Controller akan menyimpan data pesanan tersebut pada basis data dengan memanggil fungsi *insert_order* dan

insert_order_detail pada model. Setelah memanggil kedua fungsi tersebut maka *controller* akan menampilkan halaman *order_success* pada pelanggan.

D.4 Konfirmasi Pembayaran



Gambar 3.14 Diagram *Sequence* Konfirmasi Pembayaran

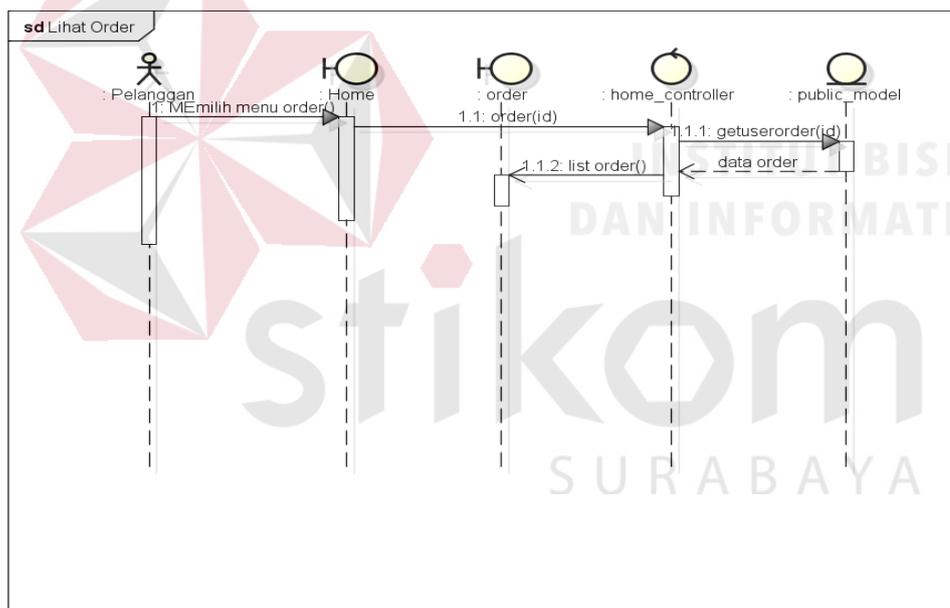
Pada Gambar 3.14 dijelaskan bahwa pada diagram *sequence* konfirmasi pembayaran dimulai pada pelanggan memilih menu *order* pada halaman *home*. Halaman tersebut akan memanggil fungsi *order* pada *controller* yang akan mengambil data pada basis data. Pengambilan data ini menggunakan fungsi *getuserorder* pada model. Fungsi tersebut akan mengembalikan data *order* kepada *controller* untuk ditampilkan pada halaman *order*.

Pada halaman *order* pelanggan dapat menekan tombol konfirmasi pembayaran. Tombol tersebut akan memanggil fungsi pada *controller* yaitu *konfirmasi_bayar* untuk merubah status *order* menjadi “verifikasi pembayaran”.

Controller akan memanggil fungsi `konfirmasi_bayar` dan `set_status` pada model untuk mengganti status pada basis data.

D.5 Lihat Order

Pada Gambar 3.15 dijelaskan bahwa diagram *sequence* order ini diawali dengan pelanggan memilih menu order pada halaman *home*. Halaman *home* akan memanggil fungsi `order` pada *controller*. *Controller* akan memanggil fungsi `getuserorder` pada model untuk mengambil data dari basis data. Fungsi `getuserorder` akan mengembalikan data kepada *controller* untuk ditampilkan pada halaman *order*.



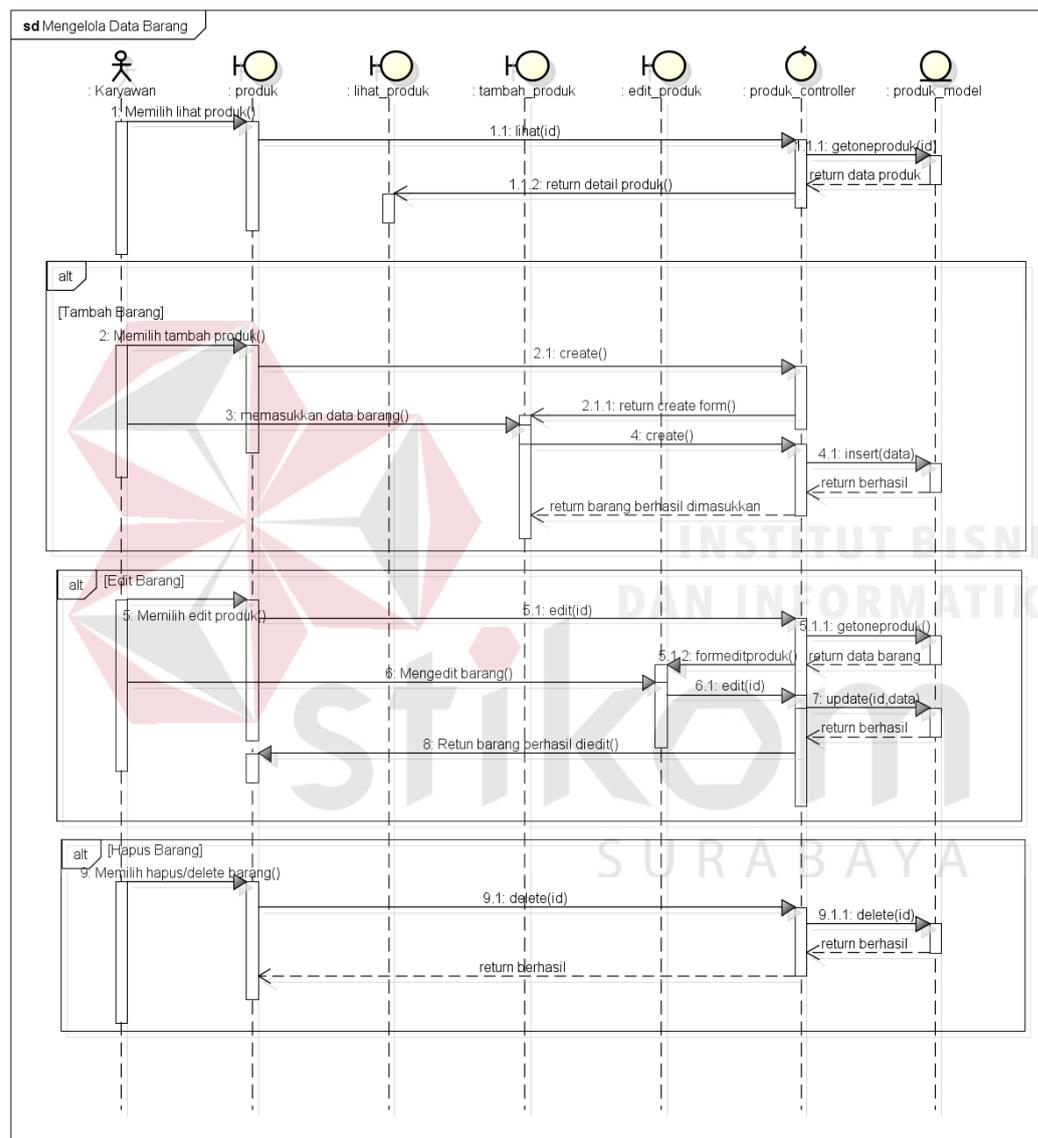
powered by Astah

Gambar 3.15 Diagram *Sequence* Lihat Order

D.6 Mengelola Data Barang

Dalam Gambar 3.16 dijelaskan bahwa diagram *sequence* mengelola barang terdapat satu buah langkah utama dan tiga buah langkah alternatif. *Sequence* yang utama adalah lihat barang yang diawali dari karyawan memilih lihat produk. Maka halaman produk akan memanggil fungsi lihat pada *controller*.

Controller akan mengambil data dari basis data menggunakan fungsi *getoneproduk* pada model. Fungsi *getoneproduk* akan mengembalikan data berupa data produk. Data produk tersebut akan ditampilkan oleh *controller* pada halaman lihat produk.



powered by Astah

Gambar 3.16 Diagram *Sequence* Mengelola Data Barang

Alternatif pertama pada *sequence* ini adalah tambah barang. Pada halaman produk. Karyawan menekan atau memilih tombol tambah produk. Tombol tersebut akan memanggil fungsi *create* pada *controller* controller akan

menampilkan halaman tambah produk. Setelah halaman tambah produk muncul karyawan dapat memasukkan data produk baru pada form tersebut. Setelah selesai memasukkan data, halaman tambah produk akan memanggil fungsi *create* pada *controller* untuk dimasukkan kedalam basis data. Untuk memasukkan data produk baru pada basis data *controller* akan memanggil fungsi *insert* pada model. Fungsi tersebut akan mengembalikan status pemasukan data berhasil pada *controller*. Dan diakhiri dengan *controller* menampilkan pesan bahwa barang berhasil dimasukkan.

Alternatif kedua adalah *edit* produk. Karyawan dapat memilih menu *edit* pada halaman produk. Tombol tersebut akan memanggil fungsi *edit* yang ada pada *controller*. *Controller* akan mengambil data produk yang dimaksud menggunakan fungsi *getoneproduk* pada model untuk kemudian ditampilkan pada halaman *edit* produk. Pada halaman *edit* produk, karyawan dapat melakukan penyuntingan sesuai dengan form yang tersedia pada halaman tersebut. Jika sudah, maka halaman akan memanggil fungsi *edit* pada *controller* untuk dimasukkan kedalam basis data dengan menggunakan fungsi *update* pada model. Fungsi *update* akan mengembalikan data status pada *controller* jika proses *edit* berhasil dan diakhiri dengan *controller* menampilkan pesan bahwa proses *edit* berhasil.

Alternatif ketiga adalah hapus produk. Karyawan dapat memilih atau menekan tombol *delete* untuk menghapus produk. Setelah tombol dipilih, tombol akan memanggil fungsi *delete* pada *controller*. *Controller* akan memanggil fungsi *delete* pada model untuk melakukan penghapusan produk pada basis data.

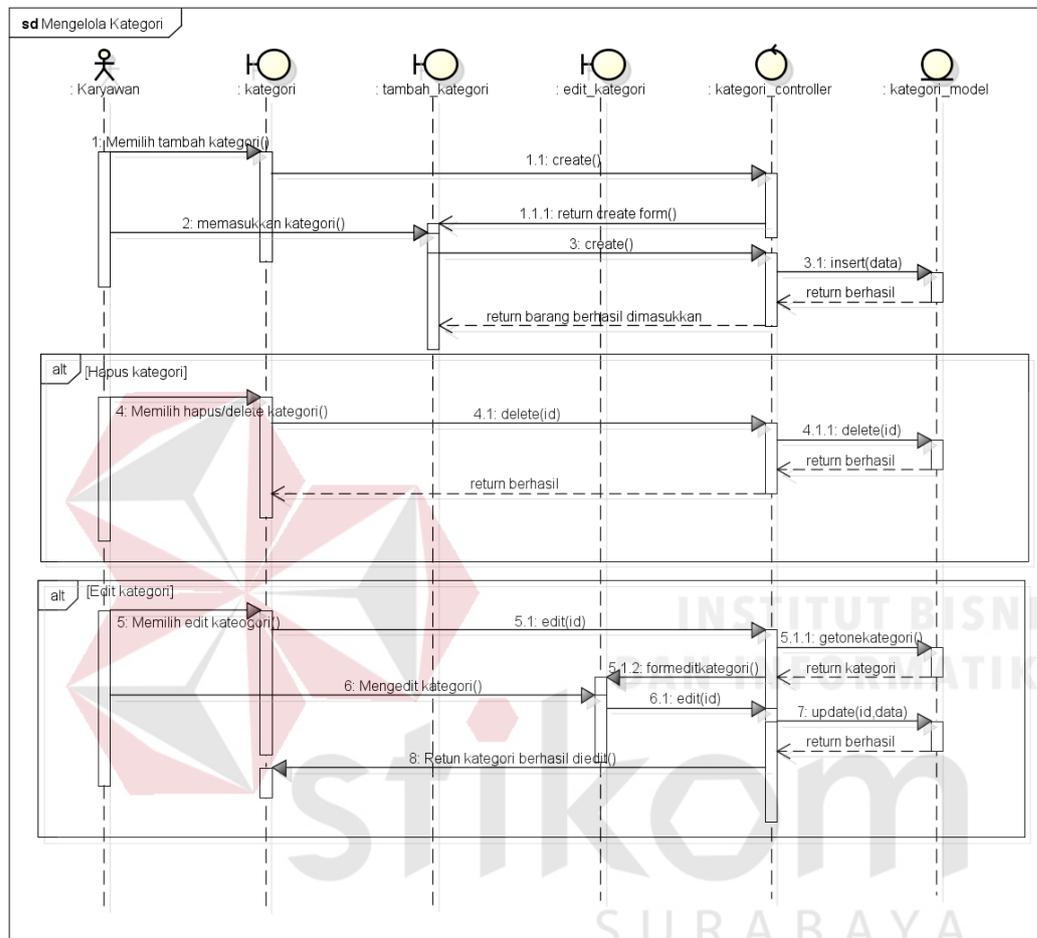
D.7 Mengelola Kategori

Dalam Gambar 3.17 dijelaskan bahwa diagram *sequence* mengelola kategori terdapat satu buah langkah utama dan dua buah langkah alternatif. *Sequence* yang utama adalah tambah kategori yang diawali dari karyawan memilih tombol tambah kategori. Maka halaman kategori akan memanggil fungsi *create* pada *controller*. Selanjutnya *controller* akan menampilkan form. Setelah pengguna memasukkan kategori baru kedalam form maka kategori baru akan dimasukkan kedalam basis data menggunakan fungsi *insert* pada model.

Alternatif pertama adalah hapus kategori. Karyawan dapat memilih atau menekan tombol *delete* untuk menghapus kategori. Setelah tombol dipilih, tombol akan memanggil fungsi *delete* pada *controller*. *Controller* akan memanggil fungsi *delete* pada model untuk melakukan penghapusan kategori pada basis data. Kategori yang bisa dihapus adalah kategori yang tidak digunakan satupun oleh suatu produk. Jika kategori ketika dihapus memiliki produk yang merefrensi kepada kategori tersebut, maka akan muncul pesan bahwa kategori masih memiliki produk.

Alternatif kedua adalah *edit* kategori. Karyawan dapat memilih menu *edit* pada halaman kategori. Tombol tersebut akan memanggil fungsi *edit* yang ada pada *controller*. *Controller* akan mengambil data kategori yang dimaksud menggunakan fungsi *getonekategori* pada model untuk kemudian ditampilkan pada halaman *edit* kategori. Pada halaman *edit* kategori, karyawan dapat melakukan penyuntingan sesuai dengan form yang tersedia pada halaman tersebut. Jika sudah, maka halaman akan memanggil fungsi *edit* pada *controller* untuk dimasukkan kedalam basis data dengan menggunakan fungsi *update* pada

model. Fungsi *update* akan mengembalikan data status pada *controller* jika proses sunting berhasil dan diakhiri dengan *controller* menampilkan pesan bahwa proses *edit* berhasil.

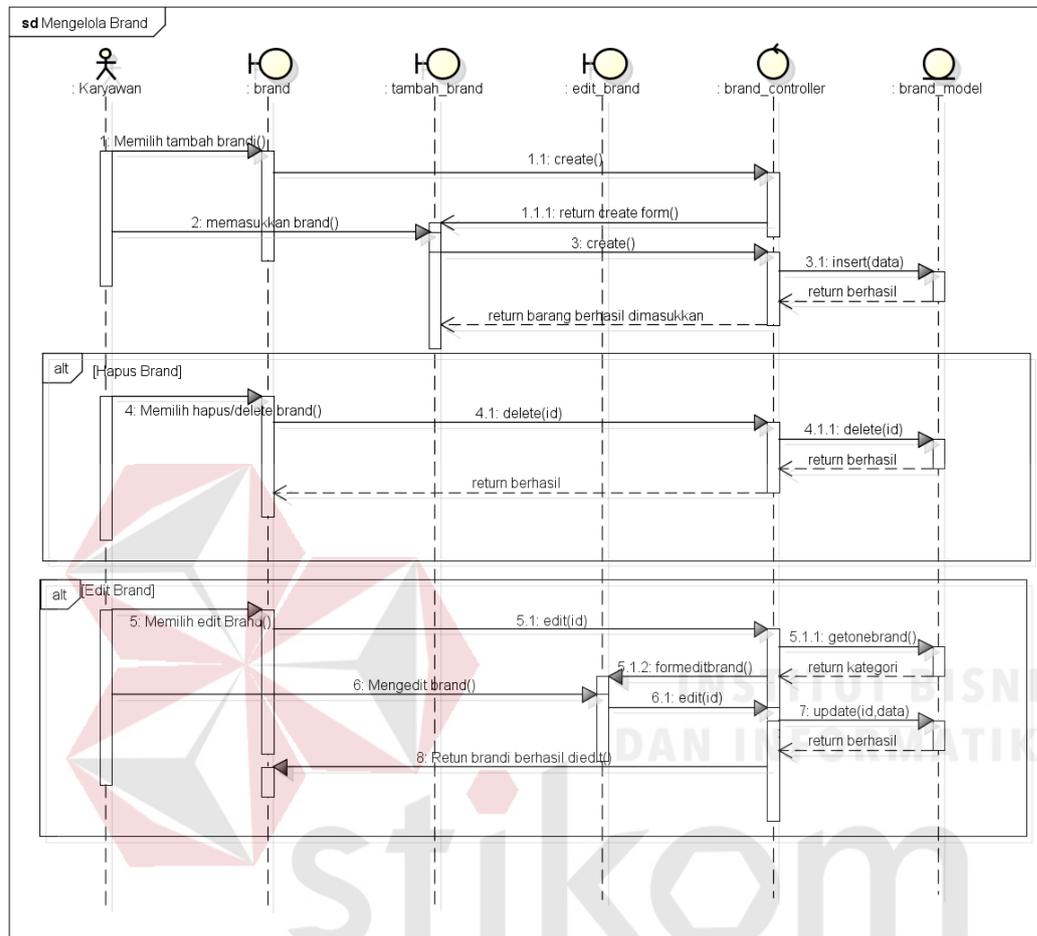


Gambar 3.17 Diagram *Sequence* Mengelola Kategori

D.8 Mengelola *Brand*

Dalam Gambar 3.18 dijelaskan bahwa diagram *sequence* mengelola *brand* terdapat satu buah langkah utama dan dua buah langkah alternatif. *Sequence* yang utama adalah tambah *brand* yang diawali dari karyawan memilih tombol tambah *brand*. Maka halaman *brand* akan memanggil fungsi *create* pada *controller*. Selanjutnya *controller* akan menampilkan form. Setelah pengguna

memasukkan *brand* baru kedalam form maka *brand* baru akan dimasukkan kedalam basis data menggunakan fungsi *insert* pada model.



Gambar 3.18 Diagram *Sequence* Mengelola *Brand*

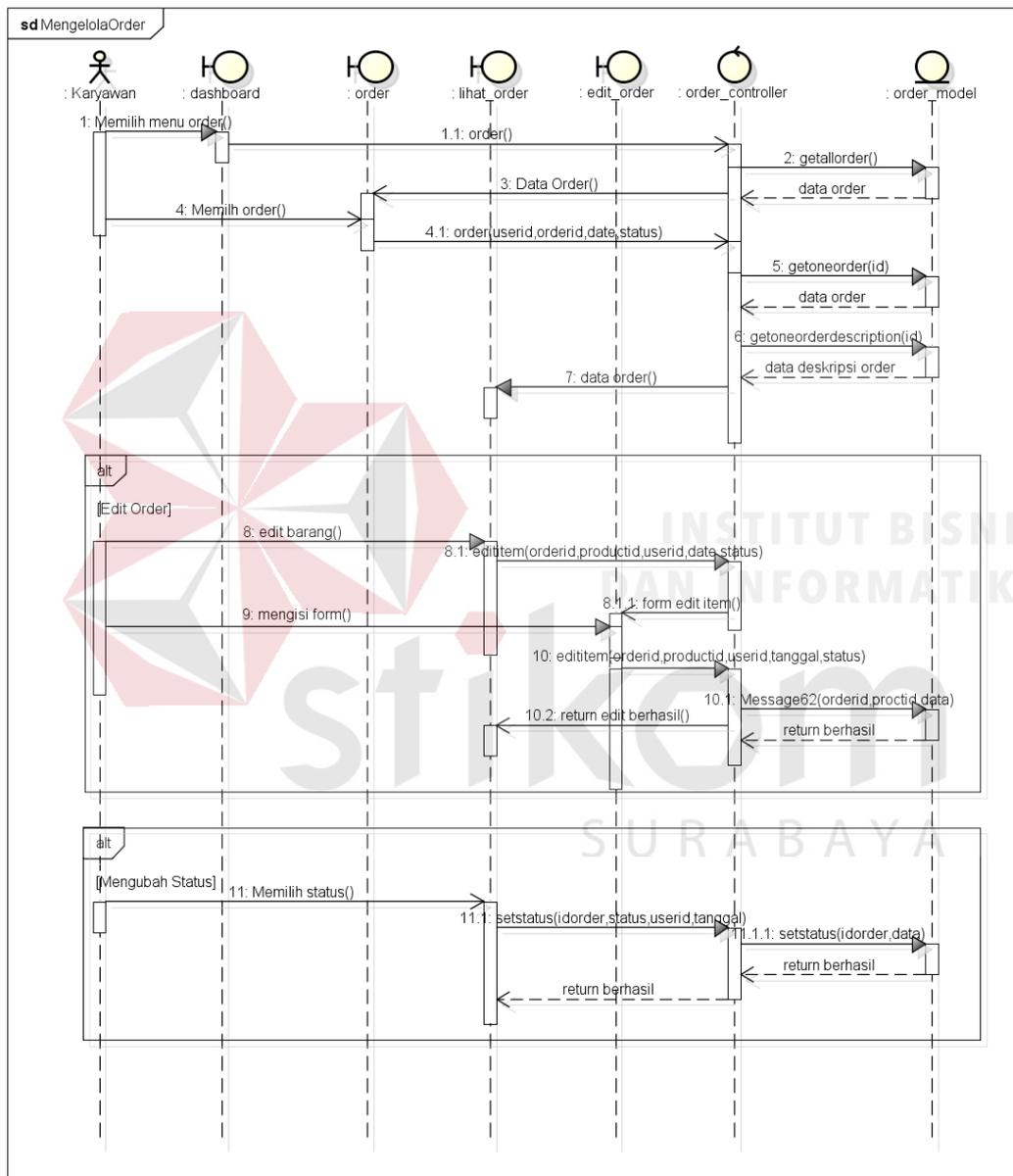
Alternatif pertama adalah hapus *brand*. Karyawan dapat memilih atau menekan tombol *delete* untuk menghapus *brand*. Setelah tombol dipilih, tombol akan memanggil fungsi *delete* pada *controller*. *Controller* akan memanggil fungsi *delete* pada model untuk melakukan penghapusan *brand* pada basis data. *Brand* yang bisa dihapus adalah *brand* yang tidak digunakan satupun oleh suatu produk. Jika *brand* ketika dihapus memiliki produk yang merefrensi kepada *brand* tersebut, maka akan muncul pesan bahwa *brand* masih memiliki produk.

Alternatif kedua adalah *edit brand*. Karyawan dapat memilih menu *edit* pada halaman *brand*. Tombol tersebut akan memanggil fungsi *edit* yang ada pada *controller*. *Controller* akan mengambil data *brand* yang dimaksud menggunakan fungsi *getonebrand* pada model untuk kemudian ditampilkan pada halaman *edit brand*. Pada halaman *edit brand*, karyawan dapat melakukan pengeditan sesuai dengan form yang tersedia pada halaman tersebut. Jika sudah, maka halaman akan memanggil fungsi *edit* pada *controller* untuk dimasukkan kedalam basis data dengan menggunakan fungsi *update* pada model. Fungsi *update* akan mengembalikan data status pada *controller* jika proses sunting berhasil dan diakhiri dengan *controller* menampilkan pesan bahwa proses *edit* berhasil.

D.9 Mengelola Order

Dalam Gambar 3.19 dijelaskan bahwa diagram *sequence* mengelola *order* terdapat satu buah langkah utama dan dua buah langkah alternatif. *Sequence* yang utama adalah lihat *order* yang diawali dari karyawan memilih tombol menu *order*. Maka halaman *dashboard* akan memanggil fungsi *order* pada *controller*. Selanjutnya *controller* akan mengambil data order dari basis data menggunakan fungsi *getallorder* dari model. Lalu model akan mengembalikan data order yang didapat. Setelah data order didapatkan maka *controller* akan menampilkan data order tersebut dalam halaman baru yaitu halaman *order*. Setelah halaman *order* ditampilkan karyawan dapat memilih order yang ingin dilihat. Setelah order dipilih maka halaman akan memanggil fungsi pada *controller* untuk mengambil data sesuai dengan order pilihan karyawan. Pengambilan data order pada basis data menggunakan fungsi yang ada pada model yaitu *getoneorder* dan *getoneorderdescription*. Fungsi *getoneorder*

digunakan untuk mengambil data ringkasan *order* secara umum sedangkan *getoneorderdescription* digunakan untuk mengambil data barang yang ada pada *order* tersebut. Setelah kedua data tersebut didapatkan maka data akan ditampilkan pada halaman baru yaitu halaman lihat *order*.



powered by Astah

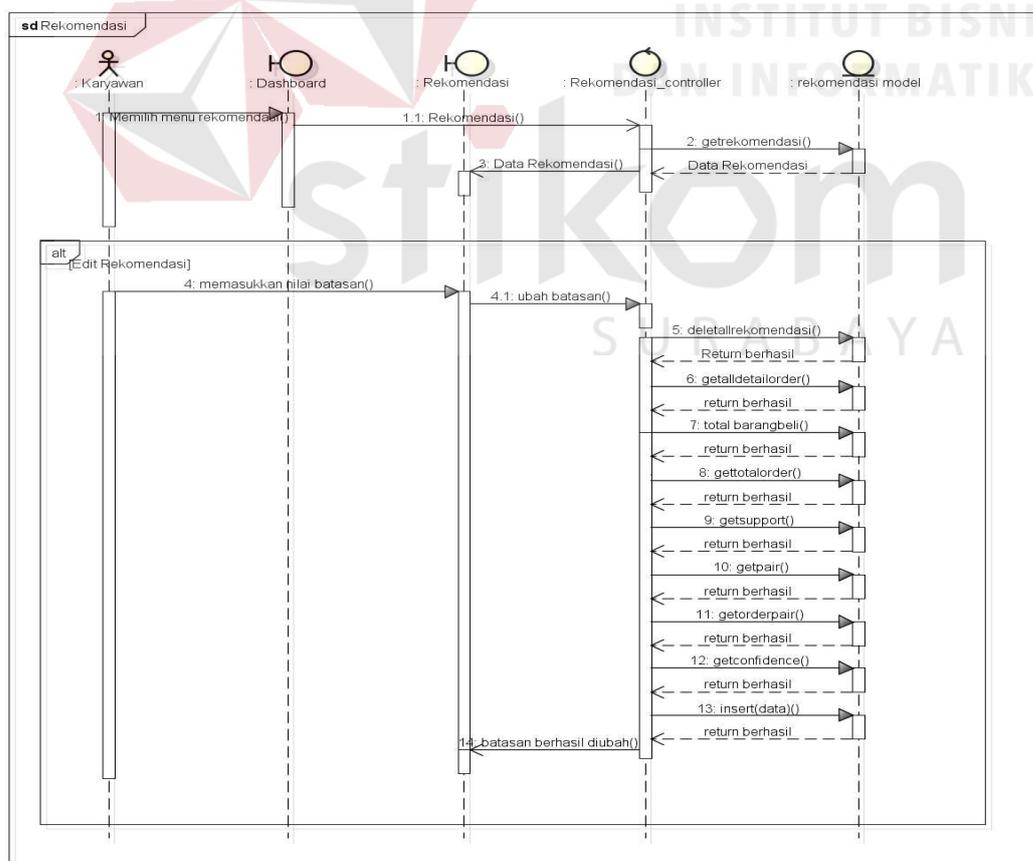
Gambar 3.19 Diagram *Sequence* Mengelola *Order*

Alternatif pertama adalah *edit order*. Karyawan dapat memilih atau menekan tombol *edit* pada daftar barang untuk di sunting. Setelah tombol dipilih,

tombol akan memanggil fungsi *edit* barang pada *controller*. *Controller* akan memanggil fungsi *edit item* pada model untuk mengambil data item tersebut pada basis data. Setelah data didapatkan maka *controller* akan menampilkan halaman form *edit item*. Setelah karyawan mengisi form tersebut maka *controller* akan menyimpan perubahan kedalam basis data.

Alternatif kedua adalah mengubah status. Pada halaman lihat *order*, karyawan dapat mengubah status sesuai dengan tombol status yang dipilih. Ketika status dipilih maka akan memanggil fungsi pada *controller* yaitu *setstatus*. Fungsi tersebut akan memanggil fungsi pada model untuk mengubah status *order* tersebut pada basis data.

D.10 Rekomendasi



Gambar 3.20 Diagram *Sequence* Rekomendasi

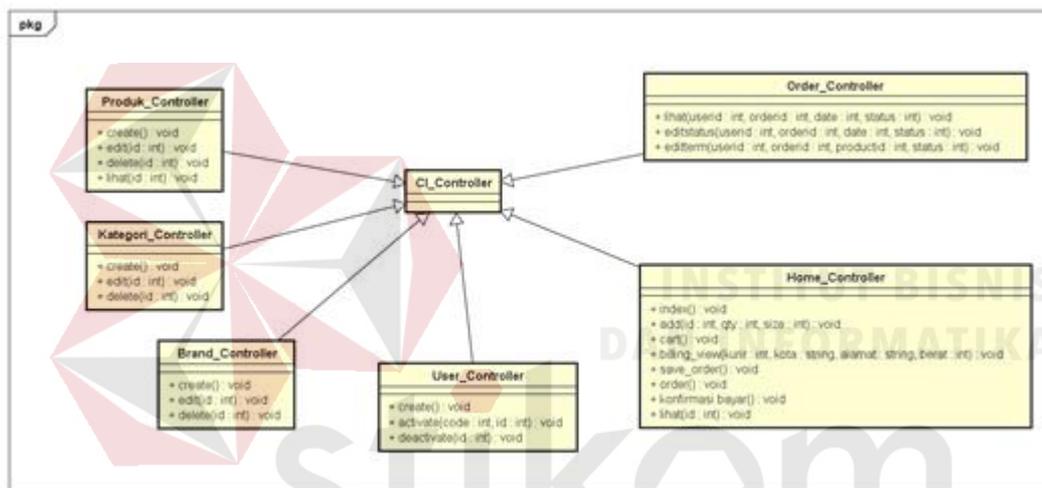
Dalam Gambar 3.20 dijelaskan bahwa diagram *sequence* rekomendasi terdapat satu buah langkah utama dan satu buah langkah alternatif. Sequence yang utama adalah lihat rekomendasi yang diawali dari karyawan memilih tombol menu rekomendasi. Maka halaman dashboard akan memanggil fungsi rekomendasi pada *controller*. Selanjutnya *controller* akan mengambil data rekomendasi dari basis data menggunakan fungsi *getrekomendasi* dari model. Lalu model akan mengembalikan data order yang didapat. Setelah data rekomendasi didapatkan maka *controller* akan menampilkan data rekomendasi tersebut dalam halaman baru yaitu halaman rekomendasi.

Alternatifnya adalah *edit* rekomendasi. Karyawan dapat memasukkan angka batasan baru pada halaman rekomendasi dan klik *submit*. Maka *controller* akan langsung memanggil fungsi *deleteallrekomendasi* untuk menghapus data rekomendasi yang lama. Kemudian dilanjutkan dengan melakukan perhitungan Algoritma Apriori lagi dengan langkah pertama yaitu *controller* memanggil fungsi *getalldetailorder* untuk mendapatkan seluruh riwayat *order*. Kedua *controller* menjalankan fungsi *totalbarangbeli* untuk mendapatkan produk-produk yang ada dalam riwayat *order*. Ketiga *controller* menjalankan fungsi *gettotalorder* untuk menghitung jumlah order yang didapatkan pada masing-masing produk. Keempat *controller* menjalankan fungsi *getsupport* untuk mendapatkan nilai *support* masing-masing produk. Kelima *controller* menjalankan fungsi *getpair* untuk melakukan kombinasi produk *2-itemset*. Keenam *controller* menjalankan fungsi *getorderpair* untuk menghitung jumlah *order* yang didapat pada masing-masing kombinasi. Ketujuh *controller* menjalankan fungsi *getconfidence* untuk mendapatkan nilai *confidence* tiap kombinasi. Terakhir *controller* memanggil

fungsi *insert* untuk menyimpan hasil perhitungan Algoritma Aprori pada *database*.

E. Class Diagram

Class Diagram merupakan diagram yang menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Karena arsitektur yang dipakai berkonsep MVC maka *class diagram* dibagi menjadi 2 yaitu *class diagram controller* dan *class diagram model*.



Gambar 3.21 *Class Diagram Controller*

E.1 Class Diagram Controller

Untuk menggambarkan hubungan *controller* yang digunakan pada aplikasi ini dapat digambarkan dengan menggunakan diagram yang dijelaskan pada Gambar 3.20. Pada diagram tersebut dapat digambarkan bahwa aplikasi ini memiliki enam buah *controller*. Masing-masing *controller* meng-*extend* *CI_Controller* karena dalam pembuatan aplikasi ini menggunakan kerangka kerja CodeIgniter.

Pada diagram berikut masing - masing *controller* memiliki fungsi sesuai dengan nama dari kelas *controller* tersebut. Pada *controller* produk memiliki 4 fungsi utama yaitu, *create*, *edit*, *delete*, dan lihat. *Controller* ini bertugas untuk mengolah data produk dari model sesuai dengan fungsi yang dipanggil.

Pada *controller* kategori memiliki 3 fungsi utama, yaitu *create*, *edit*, dan *delete*. *Controller* ini bertugas untuk mengolah data kategori dari model kategori sesuai dengan fungsi yang dipanggil.

Pada *controller brand* memiliki 3 fungsi utama, yaitu *create*, *edit*, dan *delete*. *Controller* ini bertugas untuk mengolah data *brand* dari model *brand* sesuai dengan fungsi yang dipanggil.

Pada *controller user* terdapat 3 fungsi utama yaitu, *create*, *activate*, dan *deactivate*. *Controller* ini bertugas untuk mengolah data *user* dari model *ion_auth_model* sesuai dengan fungsi yang dipanggil.

Lalu pada *controller home* memiliki 8 fungsi utama yaitu *index*, *add*, *cart*, *billing_view*, *save_order*, *order*, *konfirmasi_bayar* dan lihat. *Controller* ini memiliki fungsi paling banyak karena seluruh fungsi untuk pelanggan berada pada *controller* ini.

Sedangkan pada *controller order* memiliki 3 fungsi utama yaitu lihat, *setstatus*, dan *edititem*. *Controller* ini bertugas untuk mengolah data order dari model order sesuai dengan fungsi yang dipanggil.

E.2 Class Diagram Model

Untuk menggambarkan hubungan *model* yang digunakan pada aplikasi ini dapat digambarkan dengan menggunakan diagram yang dijelaskan pada Gambar

3.21. Pada diagram tersebut dapat digambarkan bahwa aplikasi ini memiliki tujuh buah *model*. Masing-masing *model* meng-*extend* *CI_Model* karena dalam pembuatan aplikasi ini menggunakan kerangka kerja CodeIgniter.

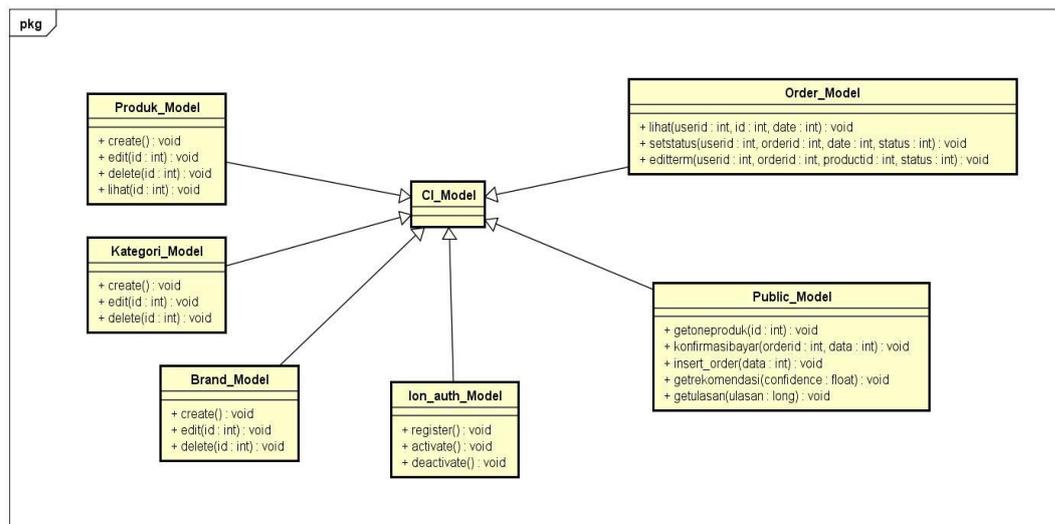
Pada diagram berikut masing - masing *model* memiliki fungsi sesuai dengan nama dari kelas *model* tersebut. Pada *model* produk memiliki 4 fungsi utama yaitu, *create*, *edit*, *delete*, dan lihat. *Model* ini bertugas untuk mengolah data produk dari basisdata sesuai dengan fungsi yang dipanggil.

Pada *model* kategori memiliki 3 fungsi utama, yaitu *create*, *edit*, dan *delete*. *Model* ini bertugas untuk mengolah data produk dari basisdata sesuai dengan fungsi yang dipanggil.

Pada *model* brand memiliki 3 fungsi utama, yaitu *create*, *edit*, dan *delete*. *Model* ini bertugas untuk mengolah data brand dari basisdata sesuai dengan fungsi yang dipanggil.

Pada *model* *ion_auth_user* terdapat 3 fungsi utama yaitu, *create*, *activate*, dan *deactivate*. *Model* ini bertugas untuk mengolah data *user* dari basisdata sesuai dengan fungsi yang dipanggil.

Lalu pada *model* *home* memiliki 8 fungsi utama yaitu *index*, *add*, *cart*, *billing_view*, *save_order*, *order*, *konfirmasi_bayar* dan *lihat*. *Model* ini memiliki fungsi paling banyak karena seluruh fungsi untuk pelanggan berada pada *model* ini. Sedangkan pada *controller* *order* memiliki 3 fungsi utama yaitu *lihat*, *setstatus*, dan *edititem*. *Controller* ini bertugas untuk mengolah data order dari model order sesuai dengan fungsi yang dipanggil.



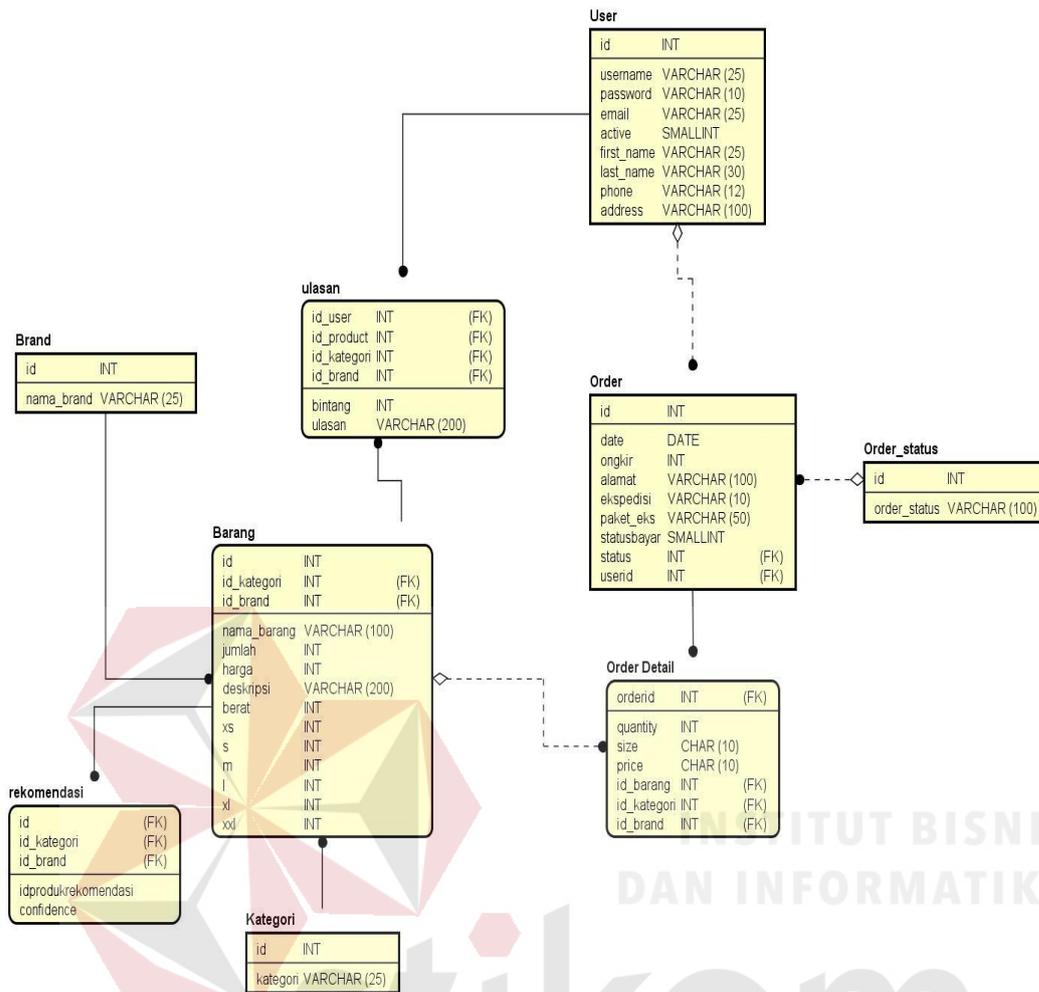
Gambar 3.22 *Class Diagram Model*

F. Desain Struktur Database

Desain struktur Database berisi tentang *Physical Data Model (PDM)* dan tabel-tabel database berdasarkan PDM tersebut.

F.1 Physical Data Model

Gambar 3.22 merupakan diagram *Physical Data Model (PDM)*. Diagram PDM berikut menunjukkan struktur data pada sebuah entitas atau tabel. Pada PDM ini dapat digambarkan hubungan antara tabel yang satu dengan yang lain. Sistem yang dibuat ini membutuhkan 7 tabel yang memiliki relasi satu sama lain. Pada PDM juga dapat diperlihatkan bagaimana struktur dari tabel tersebut mulai dari kolom hingga tipe data yang digunakan.



Gambar 3.23 Physical Data Model

F.2 Struktur Tabel

F.2.1 Tabel Kategori

Nama Tabel : Kategori

Jumlah field : 2

Fungsi : Untuk menyimpan data kategori

Tabel 3.15 Tabel Kategori

No	Nama field	Tipe	Lebar	Keterangan
1	id	int	11	Id kategori
2	kategori	varchar	25	Nama kategori

F.2.2 Tabel Brand

Nama Tabel : *Brand*

Jumlah *field* : 2

Fungsi : Untuk menyimpan data *brand*

Tabel 3.16 Tabel *Brand*

No	Nama <i>field</i>	Tipe	Lebar	Keterangan
1	Id	int	11	Id <i>brand</i>
2	Nama_ <i>brand</i>	varchar	25	Nama <i>brand</i>

F.2.3 Tabel Barang

Nama Tabel : *Barang*

Jumlah *field* : 15

Fungsi : Untuk menyimpan data barang atau produk yang dijual.

Tabel 3.17 Tabel *Barang*

No	Nama <i>field</i>	Tipe	Lebar	Keterangan
1	Nama_ <i>barang</i>	varchar	100	Nama <i>Barang</i>
2	id_kategori	int	11	Id kategori
3	id_ <i>brand</i>	Int	11	Id <i>brand</i>
5	harga	Int	11	Harga <i>barang</i>
6	deskripsi	text		Deskripsi <i>barang</i>
7	xs	int	11	Jumlah <i>barang</i> ukuran xs
8	s	int	11	Jumlah <i>barang</i> ukuran s
9	m	int	11	Jumlah <i>barang</i> ukuran m
10	l	int	11	Jumlah <i>barang</i> ukuran l
11	xL	int	11	Jumlah <i>barang</i>

No	Nama field	Tipe	Lebar	Keterangan
				ukuran xl
12	xxl	int	11	Jumlah barang ukuran xxl
13	berat	int	11	Berat barang
14	barangbaru	tinyint	1	Status barang baru 1 = barang baru 0 = barang lama
15	terlaris	tinyint	1	Status barang terlaris 1 = Barang muncul di bagian terlaris 0 + barang tidak muncul di bagian terlaris

F.2.4 Tabel Order Status

Nama Tabel : *Order Satus*

Jumlah field : 2

Fungsi : Untuk menyimpan data status *order*

Tabel 3.18 Tabel *Order Status*

No	Nama field	Tipe	Lebar	Keterangan
1	Id	int	11	Id <i>order status</i>
2	order_status	text		Keterangan order status

F.2.5 Tabel Order

Nama Tabel : *Order*

Jumlah field : 10

Fungsi : Untuk menyimpan data *order* secara umum

Tabel 3.19 Tabel Order

No	Nama field	Tipe	Lebar	Keterangan
1	Id	int	11	Id <i>order</i>
2	date	varchar	25	Tanggal <i>order</i>
3	userid	int	11	Id <i>user</i>
4	admin_viewed	Int	11	Status apakah order sudah ditinjau oleh admin 1 = Sudah 0 = Belum
5	status	Int	11	Status order sesuai dengan tabel order <i>status</i>
6	ongkir	Int	11	Jumlah ongkos kirim
7	Alamat	varchar	100	Deskripsi alamat
8	Ekspedisi	Varchar	10	Ekspedisi yang akan digunakan
9	Paketeks	Varchar	10	Paket ekspedisi yang akan digunakan
10	Statusbayar	tinyint	1	Status pembayaran order 1 = Sudah Bayar 0 = Belum Bayar

F.2.6 Tabel Order Detail

Nama Tabel : *Order Detail*

Jumlah *field* : 5

Fungsi : Untuk menyimpan data status *order*

Tabel 3.20 Tabel Status Order

No	Nama field	Tipe	Lebar	Keterangan
1	orderId	int	11	Id <i>order</i> yang berkaitan

No	Nama <i>field</i>	Tipe	Lebar	Keterangan
2	Id_barang	text	11	Id barang atau produk
3	Quantity	Int	11	Jumlah produk yang dipesan
4	Size	Int	11	Ukuran
5	Price	float		Harga barang

F.2.7 Tabel User

Nama Tabel : *User*

Jumlah *field* : 10

Fungsi : Untuk menyimpan data *user* atau pengguna dalam sistem

Tabel 3.21 Tabel *User*

No	Nama <i>field</i>	Tipe	Lebar	Keterangan
1	Id	int	11	Id <i>user</i>
2	username	varchar	25	<i>Username</i> untuk login
3	Password	Varchar	10	<i>Password</i> untuk login
4	Email	Varchar	25	Email <i>user</i>
5	<i>Active</i>	Smallint	1	Status akun aktif atau tidak 1 = aktif 0 = tidak aktif
6	Firstname	Varchar	25	Nama awal <i>user</i>
7	Lastname	Varchar	30	Nama akhir <i>user</i>
8	Phone	Varchar	12	Nomor telepon <i>user</i>
9	Email_status	smallint	1	Status verifikasi email <i>user</i> 1 = terverifikasi 0 = belum terverifikasi

No	Nama field	Tipe	Lebar	Keterangan
10	address	varchar	100	Alamat user

G. Desain Interface

Perancangan antarmuka merupakan perancangan yang menggambarkan tampilan sistem informasi atau layout dari sistem informasi. Antarmuka ini akan lebih memudahkan pada saat implementasi. Dalam bagian ini digambarkan tata letak menu, halaman form masukan, halaman laporan dan lainnya.

G.1 Desain Interface Beranda

Perancangan antarmuka beranda dalam sistem informasi pemasaran ini memiliki 4 elemen penting, yaitu Menu, Kolom Pencarian, Menu *Sorting* dan Daftar Produk. Berikut merupakan penyusunan elemen tersebut dalam halaman home.



Gambar 3.24 Perancangan Antarmuka Beranda

G.2 Desain Interface Detail Barang

Perancangan antarmuka detail barang dalam sistem informasi pemasaran ini memiliki 5 elemen penting, yaitu Menu, Kolom Pencarian, Menu *Sorting*, Detail

Produk dan Rekomendasi Produk. Berikut merupakan penyusunan elemen tersebut dalam halaman detail barang.



Gambar 3.25 Perancangan Antarmuka Detail Barang

G.3 Desain *Interface* Keranjang Belanja



Gambar 3.26 Perancangan Antarmuka Keranjang Belanja

Perancangan antarmuka keranjang belanja dalam sistem informasi pemasaran ini memiliki 5 elemen penting, yaitu Menu, Kolom Pencarian, daftar Barang, Form Alamat dan Kurir serta Tombol Proses. Berikut merupakan penyusunan elemen tersebut dalam Keranjang Belanja.

G.4 Desain *Interface* Billing info

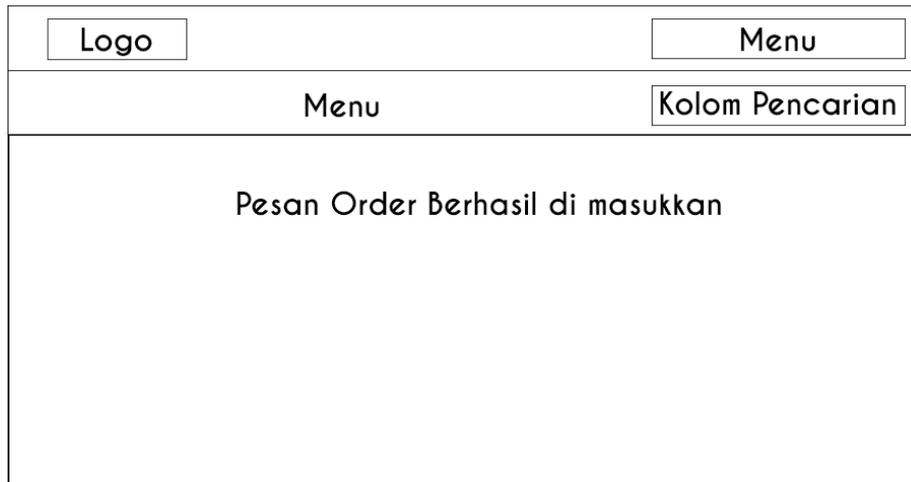
Perancangan antarmuka *billing info* dalam sistem informasi pemasaran ini memiliki 5 elemen penting, yaitu Menu, Kolom Pencarian, Daftar Barang Form Paket Kurir, dan tombol konfirmasi pesanan. Berikut merupakan penyusunan elemen tersebut dalam halaman *billing info*.



Gambar 3.27 Perancangan Antarmuka Billing Info

G.5 Desain *Interface* Order Sukses

Perancangan antarmuka *order sukses* dalam sistem informasi pemasaran ini memiliki 4 elemen penting, yaitu Menu, Kolom Pencarian, Menu *Sorting* dan Daftar Produk. Berikut merupakan penyusunan elemen tersebut dalam halaman *order sukses*.



Gambar 3.28 Perancangan Antarmuka *Order* Sukses

G.6 Desain *Interface* Daftar *Order*

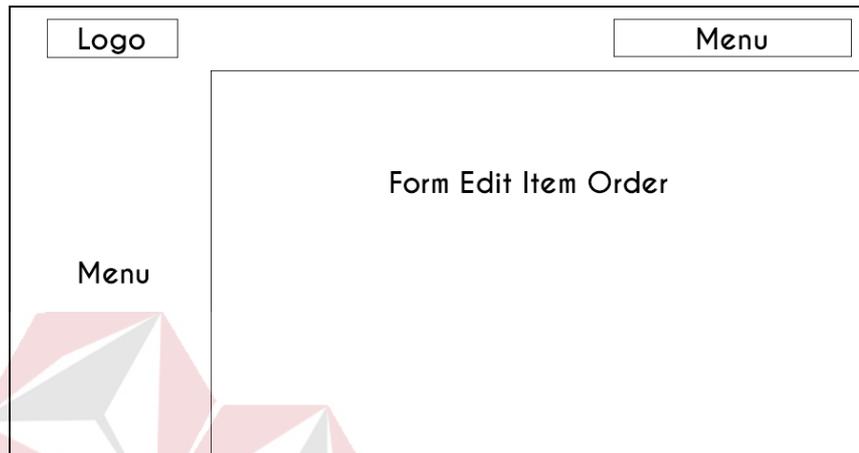
Perancangan antarmuka daftar order, kategori, produk dan brand dalam sistem informasi pemasaran ini memiliki 3 elemen penting, yaitu dua buah Menu atas dan kiri serta daftar isi dari setiap halaman yang disebutkan, Berikut merupakan penyusunan elemen tersebut.



Gambar 3.29 Perancangan Antarmuka Daftar *Order*

G.7 Desain Interface Edit Item Order

Perancangan antarmuka *edit* item order dalam sistem informasi pemasaran ini memiliki 3 elemen penting, yaitu dua buah menu di sisi kanan atas dan di sisi kiri serta *form edit* item order. Berikut merupakan penyusunan elemen tersebut.



Gambar 3.30 Perancangan Antarmuka *Edit item order*

G.8 Desain Interface Edit Kategori



Gambar 3.31 Perancangan Antarmuka *Edit Kategori*

Perancangan antarmuka *edit* kategori, produk dan *brand* dalam sistem informasi pemasaran ini memiliki 4 elemen penting, yaitu dua buah menu di kanan

atas dan disisi kiri, form *edit* dan tombol *submit*. Berikut merupakan penyusunan elemen tersebut.

G.9 Desain *Interface* Tambah Kategori

Perancangan antarmuka tambah kategori, produk dan *brand* dalam sistem informasi pemasaran ini memiliki 4 elemen penting, yaitu dua buah menu dikanan atas dan disisi kiri, form *edit* dan tombol *submit*. Berikut merupakan penyusunan elemen tersebut.



Gambar 3.32 Perancangan Antarmuka Tambah Kategori

G.10 Desain *Interface* Daftar User

Perancangan antarmuka daftar *user* dalam sistem informasi distribusi produk ini memiliki 3 elemen penting, yaitu dua buah menu dan daftar user. Berikut merupakan penyusunan elemen tersebut.



Gambar 3.33 Perancangan Antarmuka Daftar *User*

G.11 Desain Interface Tambah User

Perancangan antarmuka tambah *user* dalam sistem informasi pemasaran ini memiliki 4 elemen penting, yaitu dua buah menu di kanan atas dan disisi kiri, form tambah *user* dan tombol *submit*. Berikut merupakan penyusunan elemen tersebut.



Gambar 3.34 Perancangan Antarmuka Tambah *User*

BAB IV

IMPLEMENTASI DAN EVALUASI SISTEM

Pada bagian ini menjelaskan tentang hasil dari implementasi yang diperoleh dari analisis data dan perancangan sistem informasi. Hasil dari analisis data digunakan untuk menentukan kebutuhan dari sistem informasi agar lebih mudah dalam implementasi sistem informasi yang sudah dijelaskan pada bab sebelumnya. Adapun isi dari bab 4 sendiri meliputi Spesifikasi sistem, implementasi apriori, implementasi antarmuka dan pengujian.

4.1 Spesifikasi Sistem

Pada bagian ini dijelaskan spesifikasi perangkat keras dan perangkat lunak yang digunakan ketika implementasi sistem informasi. Perangkat keras yang digunakan adalah suatu unit laptop yang spesifikasinya dijelaskan pada Tabel 4.1.

Tabel 4.1 Spesifikasi Perangkat Keras

Unit Komputasi	Laptop ASUS A450LC
CPU	Intel Core i5-4200U 2.3 GHz
Kapasitas RAM	8 GB
Kapasian Penyimpanan	500 GB
Kartu Grafis	NVIDIA GT-720M
Resolusi Layar	1366 X 768 pixels

Sedangkan perangkat lunak yang digunakan untuk implementasi sistem informasi akan dijelaskan pada Tabel 4.2 berikut.

Tabel 4.2 Spesifikasi Perangkat Lunak

Sistem Informasi	Microsoft Windows 8/10
Web Server	Apache 2.4.25
DBMS	Mysql
Bahasa Pemrograman	PHP 5.6.30
Editor Kode Program	Sublime Text 3
Peramban	Mozilla Firefox Quantum 63.0.1
Bahasa Pemrograman Pendukung	Javascript, HTML

4.2 Implementasi Apriori

Implementasi Algoritma Apriori dalam aplikasi dibagi dalam 5 tahap utama yang sebelum masuk kedalam tahap pertama dimulai sistem mencari histori transaksi dalam *database* dengan menggunakan source code sebagai berikut

```

$rawlistorder=$this->public_model->getalldetailorder()->result_array();
$listorder = array();
$i = 0;
foreach ($rawlistorder as $row) {
    if (array_key_exists($row['orderid'], $listorder) == false) {
        $i = 0;
        $listorder[$row['orderid']][$i] = $row['productid'];
    }elseif (array_key_exists($row['orderid'], $listorder) == true)
    {
        $i++;
        $listorder[$row['orderid']][$i] = $row['productid'];
    }
}

```

Dalam *source code* tersebut dilakukan proses pencarian histori transaksi dalam *database* yang dibentuk dalam *array* 2D dan disajikan dalam bentuk tabel dengan 2 kolom, kolom pertama adalah kolom *orderid* dan kolom kedua berisi id produk yang dibeli. Hasil dari *source code* diatas dapat dilihat pada Gambar 4.1 berikut.

ID Order	ID Produk
76	58, 57, 23, 30
77	57, 23
81	23, 53, 2
80	2, 1
79	31, 54
78	31, 58, 55
75	53, 1, 30
74	31, 54, 56
53	57, 1
73	32, 58, 56
57	30, 58
58	31, 58
67	56, 2, 58
68	55, 54
70	2, 57
71	54, 31
72	54, 31, 32, 58
82	30, 32, 56
83	54, 31, 1
84	56, 32

Gambar 4.1 Hasil pencarian histori transaksi

Setelah dilakukan pencarian histori transaksi dilakukanlah proses perhitungan yang dilakukan dalam 5 tahap.

4.2.1 Tahap 1

Pada tahap ini dilakukan perhitungan jumlah transaksi yang dilakukan tiap produk pada seluruh transaksi yang terjadi. Implementasi tahap ini dilakukan pada aplikasi dengan menggunakan *source code* sebagai berikut.

```
$rawlist2=$this->public_model->gettotalbarangterbeli()->result_array();
$list2 = array();
```

```

foreach ($rawlist2 as $row) {
    if (array_key_exists($row['idbarang'], $list2) == false) {
        $list2[$row['idbarang']] = $row['jumlah'];
    }elseif (array_key_exists($row['idbarang'], $list2) == true) {
        $list2[$row['idbarang']] = $row['jumlah'];
    }
}

```

Dalam *source code* diatas dilakukan perhitungan jumlah transaksi yang dilakukan tiap produk pada seluruh transaksi yang terjadi yang dibentuk dalam *array* 2D dan disajikan dalam bentuk tabel 2 kolom, dimana dalam kolom-kolomnya berisi id produk dan jumlah transaksi. Hasil dari *source code* diatas dapat dilihat pada Gambar 4.2 berikut.

ID Produk	Jumlah
31	7
58	7
54	6
56	5
2	4
1	4
30	4
32	4
57	4
23	3
55	2
53	2

Gambar 4.2 Hasil tahap 1

4.2.2 Tahap 2

Tahap kedua dalam Algoritma Apriori adalah menghitung nilai *support* tiap produk terhadap transaksi dengan rumus :

$$\text{Support} = \frac{\text{Jumlah transaksi produk}}{\text{total transaksi}}$$

Implementasi tahap ini dilakukan pada aplikasi dengan menggunakan *source code* sebagai berikut.

```

$jumlahorder = $this->diskon_model->gettotalorder()->row();
$support1 = array();
foreach ($list2 as $key => $value) {
    $x = $value / $jumlahorder->jumlahorder;
    if ($x >= $batas->batas) {
        $support1[$key] = $x;
    }
}

```

Dalam *source code* diatas dilakukan perhitungan nilai *support* tiap produk yang dibentuk dalam *array* 2D dan disajikan dalam bentuk tabel 2 kolom, dimana dalam kolom-kolomnya berisi id produk dan nilai *support*. Setelah diketahui nilai *support* kemudian disaring dengan batas yang telah ditentukan. Hasil dari *source code* diatas dapat dilihat pada Gambar 4.3 berikut.

ID Produk	Support
31	0.35
58	0.35
54	0.3
56	0.25
2	0.2
1	0.2
30	0.2
32	0.2
57	0.2
23	0.15
55	0.1
53	0.1

Gambar 4.3 Hasil tahap 2

Hasil diatas merupakan keluaran tahap 2 dengan menggunakan batasan yang bernilai 0.

4.2.3 Tahap 3

Pada tahap ketiga ini dilakukan pengkombinasian produk yang lolos dalam penyaringan berdasarkan batasan yang ditentukan. Implementasi tahap ini dilakukan pada aplikasi dengan menggunakan *source code* sebagai berikut.

```

$maxpair = 0;
foreach ($listorder as $key => $value) {
    $x = count($listorder[$key]);
    if ($x > $maxpair) {
        $maxpair = $x; }
    $temp = array();
    $counter = 0;
    foreach ($support1 as $key => $value) {
        $temp[$counter] = $key;
        $counter++;}
    $pair = array();
    $counter = 0;
    for ($i=0; $i < count($temp); $i++) {
        for ($j=0; $j < count($temp); $j++) {
            $pair[$counter++] = $temp[$i].','.$temp[$j];}}

```

Dalam *source code* diatas dilakukan pengkombinasian produk yang dibentuk dalam *array* 2D dan disajikan dalam bentuk tabel 2 kolom, dimana dalam kolom-kolomnya berisi nomor dan *pair* barang. Hasil dari *source code* diatas dapat dilihat pada Gambar 4.4 berikut.

No	Pair
0	31,31
1	31,58
2	31,54
3	31,56
4	31,2
5	31,1
6	31,30
7	31,32
8	31,57
9	31,23
10	31,55
11	31,53
12	58,31
13	58,58

Gambar 4.4 Hasil tahap 3

Hasil diatas merupakan sebagian keluaran tahap 3 dengan menggunakan batasan yang bernilai 0. Jumlah *pair* atau pasangan produk yang dibentuk sebanyak 144 pasangan.

4.2.4 Tahap 4

Pada tahap ini dilakukan perhitungan jumlah transaksi tiap *pair* pada riwayat pembelian pelanggan. Implementasi tahap ini dilakukan pada aplikasi dengan menggunakan *source code* sebagai berikut.

```
$listorderpair = array();
$countner = 0;
foreach ($listorder as $key => $value) {
    if (count($listorder[$key]) == 2) {
        $x = $listorder[$key][0].','.$listorder[$key][1];
```

```

foreach ($pair as $key2 => $value2) {
    if ($x == $value2) {
        $listorderpair[$x] = ++$counter;
    }
}
else{
    $counter = 0;
}
}

```

Dalam *source code* diatas dilakukan perhitungan jumlah transaksi tiap *pair* yang dibentuk dalam *array* 2D dan disajikan dalam bentuk tabel 2 kolom, dimana dalam kolom-kolomnya berisi *pair* barang dan jumlah. Hasil dari *source code* diatas dapat dilihat pada Gambar 4.5 berikut.

Pair	Jumlah
57,23	1
2,1	1
57,1	1
30,58	2
31,58	3
55,54	1
2,57	2
54,31	3
56,32	1

Gambar 4.5 Hasil Tahap 4

4.2.5 Tahap 5

Pada tahap ini dilakukan perhitungan nilai *confidence* tiap pasangan dengan rumus:

$$\text{Confidence} = P(B | A) = \frac{\text{Jumlah Transaksi mengandung A dan B}}{\text{Jumlah Transaksi mengandung A}}$$

Implementasi tahap ini dilakukan pada aplikasi dengan menggunakan *source code* sebagai berikut.

```

$confidence = array();
foreach ($listorderpair as $key => $value) {
    foreach ($list2 as $key2 => $value2) {
        $str = explode(",", $key);
        if ($str[0] == $key2) {
            $confidence[$key."->".$key2] = $value / $value2;
            continue;
        }elseif ($str[1] == $key2) {
            $confidence[$key."->".$key2] = $value / $value2;
            continue;
        }
    }
}

```

Dalam *source code* diatas dilakukan perhitungan nilai *confidence* tiap *pair* atau pasangan produk yang dibentuk dalam *array* 2D. Kemudian dilakukan pemecahan *pair* menggunakan fungsi *explode* untuk membentuk tabel 3 kolom yang terbentuk dari nama produk, produk rekomendasi dan nilai *confidence*. Hasil dari *source code* diatas dapat dilihat pada Gambar 4.6 berikut.

Nama Produk	Rekomendasi Produk	Nilai Confidence
Baju Koko Polos Merah	motif kotak merah	0.25
motif kotak merah	Baju Koko Polos Merah	0.333333
GE8 Coklat Kopi	GE14 Merah	0.25
GE14 Merah	GE8 Coklat Kopi	0.25
GE14 Merah	Baju Koko Polos Merah	0.25
Baju Koko Polos Merah	GE14 Merah	0.25
Peci Rajut	KD 7	0.285714
KD 7	Peci Rajut	0.5
Laila Instant Ungu	Peci Rajut	0.428571
Peci Rajut	Laila Instant Ungu	0.428571
GM 09	Couple 06 Maroon	0.166667
Couple 06 Maroon	GM 09	0.5

Gambar 4.6 Hasil Tahap 5

Setelah dilakukan perhitungan nilai *confidence* maka dilakukan maka Algoritma Apriori telah selesai dilakukan. Keluaran dari hasil impementasi ini dapat dilihat saat pelanggan mengklik barang untuk melihat barang yang akan dibeli, maka sistem akan memunculkan produk rekomendasi yang terdapat pada kolom rekomendasi produk. Keluaran ini dapat dilihat pada Gambar 4.16.

4.3 Implementasi Antarmuka

Dalam bagian ini terdapat desain antarmuka yang dibagi menjadi 2 desain yaitu desain untuk Pemilik/admin dan pelanggan.

4.3.1 Admin

Pada bagian admin terdapat 6 fungsi utama yaitu Dashboard, Order, Produk, Brand/Kategori, Rekomendasi, dan User.

A. Implementasi Antarmuka Dashboard

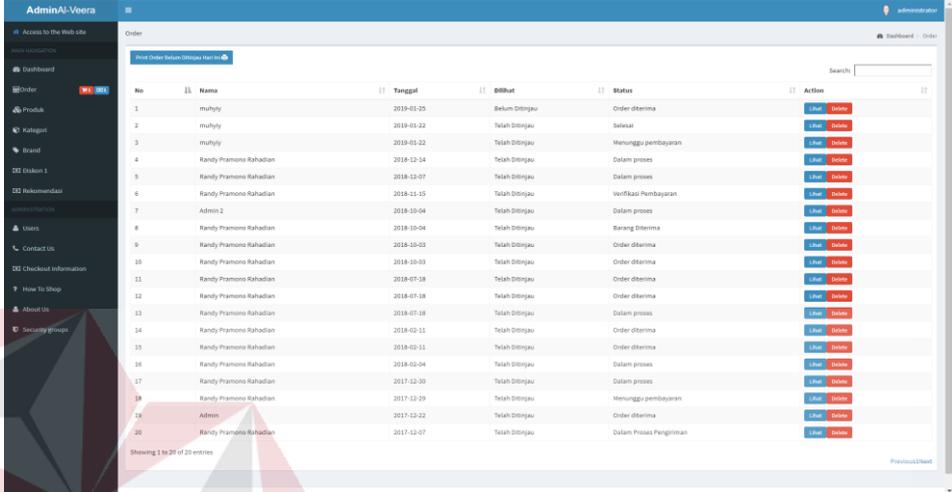


Gambar 4.7 Antarmuka Dashboard

Sesaat setelah pemilik atau admin melakukan login maka admin akan menuju pada halaman awal yang berupa dashboard. Pada halaman dashboard ini terdapat grafik-grafik terkait penjualan barang pada toko, seperti jumlah order per

hari, order per bulan, order tiap brand bulan ini, dan 5 barang terlaris. Selain itu juga terdapat informasi jumlah order hari ini, order per bulan, rata-rata order tiap bulan dan total order.

B. Implementasi Antarmuka Daftar Order



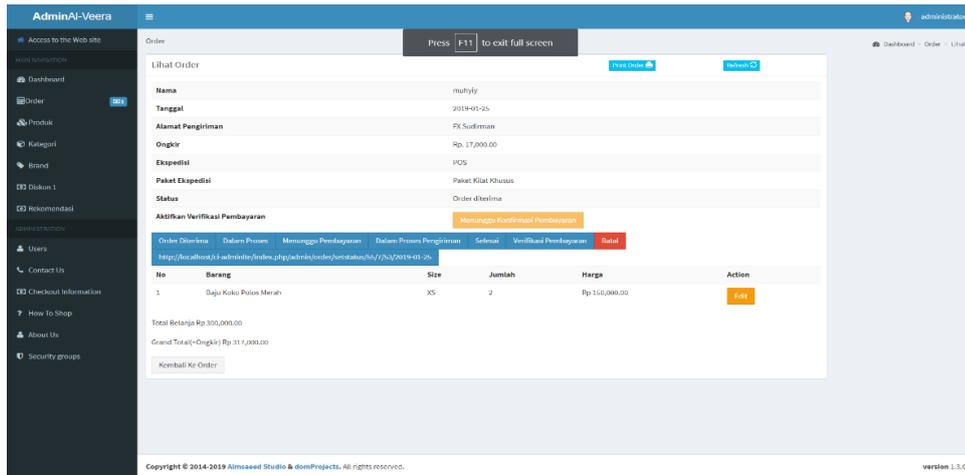
No	Nama	Tanggal	Dibuat	Status	Action
1	Mulyly	2019-01-25	Belum Ditinggou	Order diterima	Lihat Delete
2	Mulyly	2019-01-22	Telah Ditinggou	Salaku	Lihat Delete
3	Mulyly	2019-01-22	Telah Ditinggou	Menunggu pembayaran	Lihat Delete
4	Randy Pramono Rahadian	2018-12-14	Telah Ditinggou	Dalam proses	Lihat Delete
5	Randy Pramono Rahadian	2018-12-07	Telah Ditinggou	Dalam proses	Lihat Delete
6	Randy Pramono Rahadian	2018-11-15	Telah Ditinggou	Verifikasi Pembayaran	Lihat Delete
7	Admin 2	2018-10-04	Telah Ditinggou	Dalam proses	Lihat Delete
8	Randy Pramono Rahadian	2018-10-04	Telah Ditinggou	Barang Diterima	Lihat Delete
9	Randy Pramono Rahadian	2018-10-03	Telah Ditinggou	Order diterima	Lihat Delete
10	Randy Pramono Rahadian	2018-10-03	Telah Ditinggou	Order diterima	Lihat Delete
11	Randy Pramono Rahadian	2018-07-18	Telah Ditinggou	Order diterima	Lihat Delete
12	Randy Pramono Rahadian	2018-07-18	Telah Ditinggou	Order diterima	Lihat Delete
13	Randy Pramono Rahadian	2018-07-18	Telah Ditinggou	Dalam proses	Lihat Delete
14	Randy Pramono Rahadian	2018-02-11	Telah Ditinggou	Order diterima	Lihat Delete
15	Randy Pramono Rahadian	2018-02-11	Telah Ditinggou	Order diterima	Lihat Delete
16	Randy Pramono Rahadian	2018-02-04	Telah Ditinggou	Dalam proses	Lihat Delete
17	Randy Pramono Rahadian	2017-12-30	Telah Ditinggou	Dalam proses	Lihat Delete
18	Randy Pramono Rahadian	2017-12-29	Telah Ditinggou	Menunggu pembayaran	Lihat Delete
19	Admin	2017-12-22	Telah Ditinggou	Order diterima	Lihat Delete
20	Randy Pramono Rahadian	2017-12-07	Telah Ditinggou	Dalam Proses Pengiriman	Lihat Delete

Gambar 4.8 Antarmuka Daftar Order

Dalam antarmuka daftar *order*, data *order* disajikan dalam bentuk tabel. Tabel ini berisi tentang garis besar order yang masuk yaitu, nama pelanggan, tanggal order, status dilihat, status order, dan dua buah pilihan aksi yaitu lihat dan delete. Pada tabel ini juga dilengkapi dengan kolom pencarian agar mudah mencari *order*.

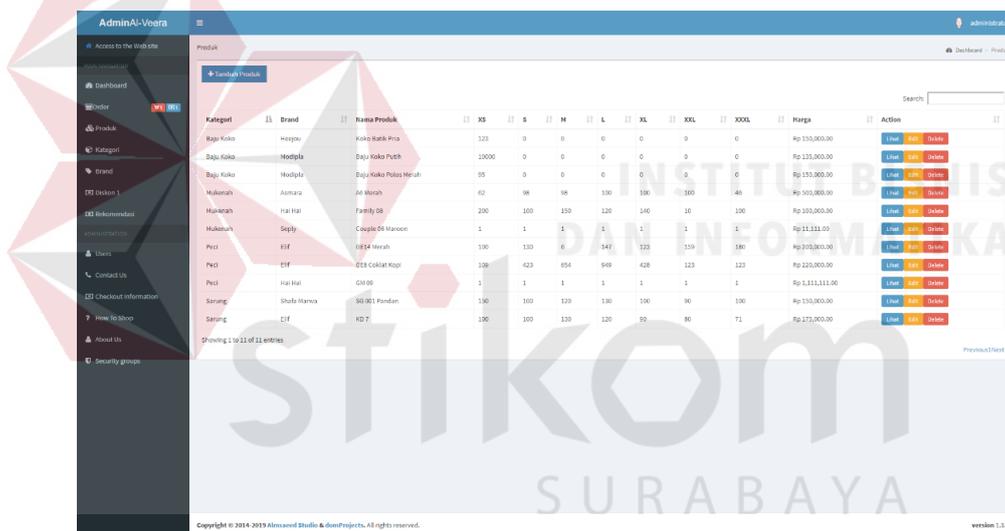
C. Implementasi Antarmuka Lihat Order

Dalam implementasi antarmuka lihat *order* terdapat dua bagian yaitu deskripsi order dan detail order. Bagian deskripsi order memuat nama pelanggan, tanggal, alamat pengiriman, ongkor kirim, ekspedisi, paket ekspedisi, nomor resi dan status. Pada bagian detail order terdapat tabel barang yang dibeli.



Gambar 4.9 Antarmuka lihat order

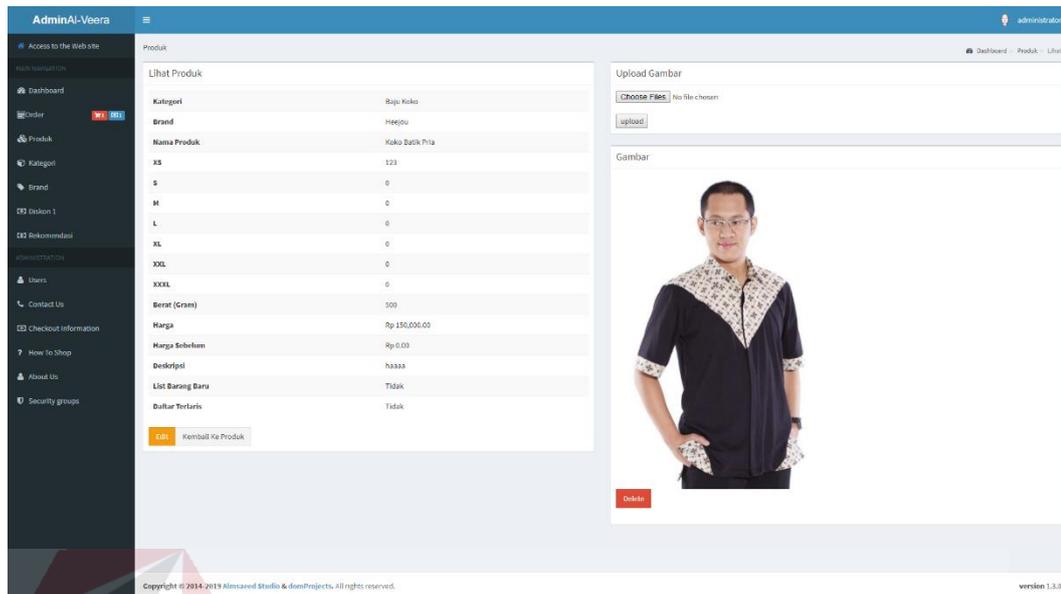
D. Implementasi Antarmuka Daftar Produk



Gambar 4.10 Antarmuka Daftar Produk

Dalam implementasi antarmuka daftar produk, daftar produk ditampilkan dalam bentuk tabel. Data tabel ini berisi kategori, nama *brand* nama produk, stok setiap ukuran, harga dan tiga tombol aksi pada masing-masing produk. Pada tabel ini juga disertakan kolom pencarian untuk mempermudah pencarian produk secara spesifik. Serta tabel juga bisa melakukan sorting data berdasarkan nama kolom.

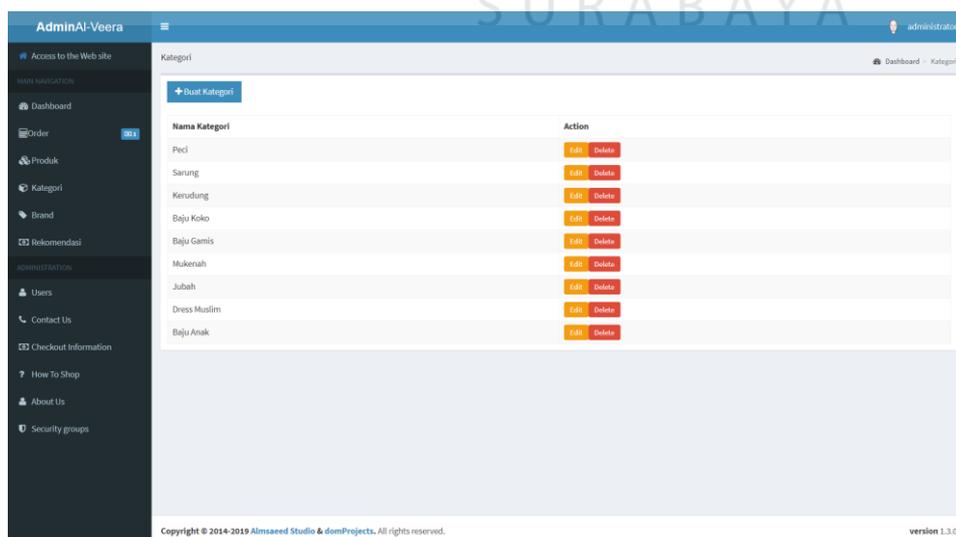
E. Implementasi Antarmuka Lihat Produk



Gambar 4.11 Antarmuka Lihat Produk

Implementasi antarmuka lihat produk terdapat dua bagian, kiri dan kanan. Pada bagian kiri terdapat detail keterangan produk, mulai dari kategori, *brand*, nama produk, jumlah *stock* per ukuran, deskripsi, harga, dan opsi. Dibagian kanan terdapat tombol untuk menambah gambar dan daftar gambar produk.

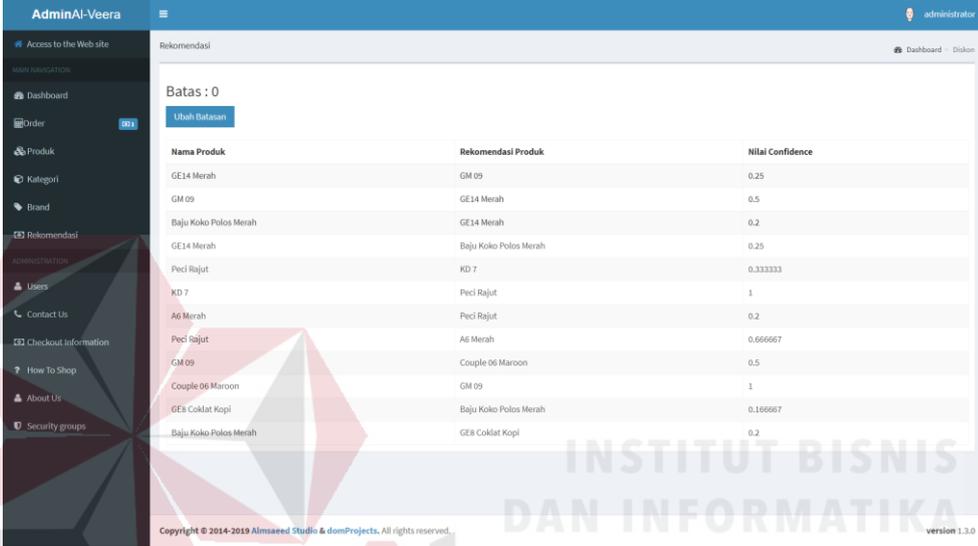
F. Implementasi Antarmuka Daftar Brand



Gambar 4.12 Antarmuka Daftar *Brand*

Implementasi Antarmuka daftar *brand* dan kategori ditampilkan dalam bentuk tabel 2 kolom, dimana dalam kolom pertama berisi nama brand atau kategori, sedangkan pada kolom kedua berisi dua tombol aksi yaitu *edit* dan *delete*.

G. Implementasi Antarmuka Rekomendasi



Nama Produk	Rekomendasi Produk	Nilai Confidence
GE14 Merah	GM 09	0.25
GM 09	GE14 Merah	0.5
Baju Koko Polos Merah	GE14 Merah	0.2
GE14 Merah	Baju Koko Polos Merah	0.25
Peci Rajut	KD 7	0.333333
KD 7	Peci Rajut	1
A6 Merah	Peci Rajut	0.2
Peci Rajut	A6 Merah	0.666667
GM 09	Couple 06 Maroon	0.5
Couple 06 Maroon	GM 09	1
GE8 Coklat Kopi	Baju Koko Polos Merah	0.166667
Baju Koko Polos Merah	GE8 Coklat Kopi	0.2

Gambar 4.13 Antarmuka Rekomendasi

Implementasi antarmuka rekomendasi menampilkan informasi batasan yang digunakan dan hasil perhitungan Algoritma Apriori dalam bentuk tabel. Terdapat tombol ubah batasan untuk mengganti nilai batasan dalam perhitungan apriori. Hasil perhitungan apriori ditampilkan dalam bentuk tabel dengan 3 kolom dimana kolom pertama berisi nama produk, kolom kedua berisi nama produk rekomendasi dan kolom ketiga berisi nilai confidence pasangan produk tersebut.

H. Implementasi Antarmuka Daftar User

Implementasi antarmuka daftar *user* ditampilkan dalam bentuk tabel dengan 6 kolom, dimana kolom-kolomnya berisi nama, e-mail, status pengguna, status email, status akun dan kolom yang berisi 3 tombol aksi yaitu lihat, *edit* dan

delete. Selain itu terdapat 2 tombol dibagian kiri atas halaman yang digunakan untuk menambah user

Name	Email	Groups	Status Email	Status Akun	Action
Admin	admin@admin.com	admin	Verified	Active	Lihat Edit Delete
Admin 2	admin2@gmail.com	admin	Unverified	Active	Lihat Edit Delete
Ahmad Baihaqy	ahmadbaihaqy23@gmail.com	member	Unverified	Active	Lihat Edit Delete
Bagus Ramadhan	ramadhangrad@gmail.com	member	Unverified	Active	Lihat Edit Delete
Ferdi Budi	fb018@gmail.com	member	Unverified	Active	Lihat Edit Delete
Jhon	roti@jhon.com	member	Unverified	Active	Lihat Edit Delete
Larry	larrysikan@gmail.com	member reseller	Unverified	Active	Lihat Edit Delete
Muhyly	muhyly@m.com	member	Unverified	Active	Lihat Edit Delete
Ocin	ocin@ocin.com	member	Unverified	Active	Lihat Edit Delete
Randy Pramono Rahadian	rmdpramono@gmail.com	agen	Verified	Active	Lihat Edit Delete
Tam Tama	tamtama@gmail.com		Unverified	Active	Lihat Edit Delete
TEST SUBJECT	tesusubject@gmail.com	member	Unverified	Inactive	Lihat Edit Delete
TEST SUBJECT 2	tesusubject2@gmail.com	agen	Unverified	Inactive	Lihat Edit Delete
Testing 1	testing1@gmail.com	agen	Unverified	Inactive	Lihat Edit Delete

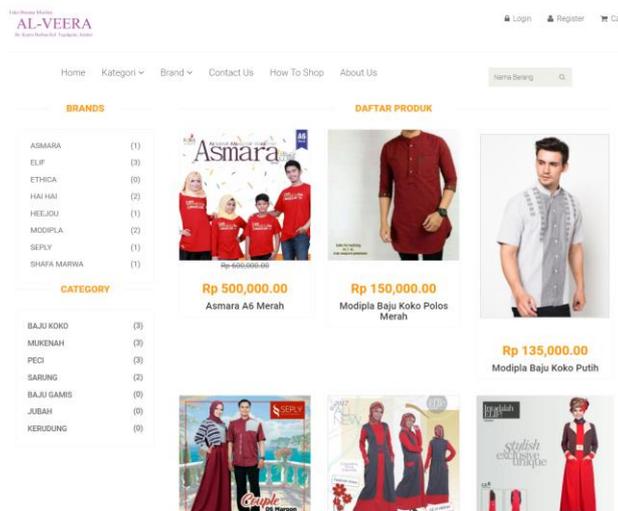
Gambar 4.14 Antarmuka Daftar *User*

4.3.2 Pelanggan

Pada Antarmuka pelanggan terdapat Beranda, Detail Barang, Cart, Billing Info, Billing berhasil dan Order Pelanggan.

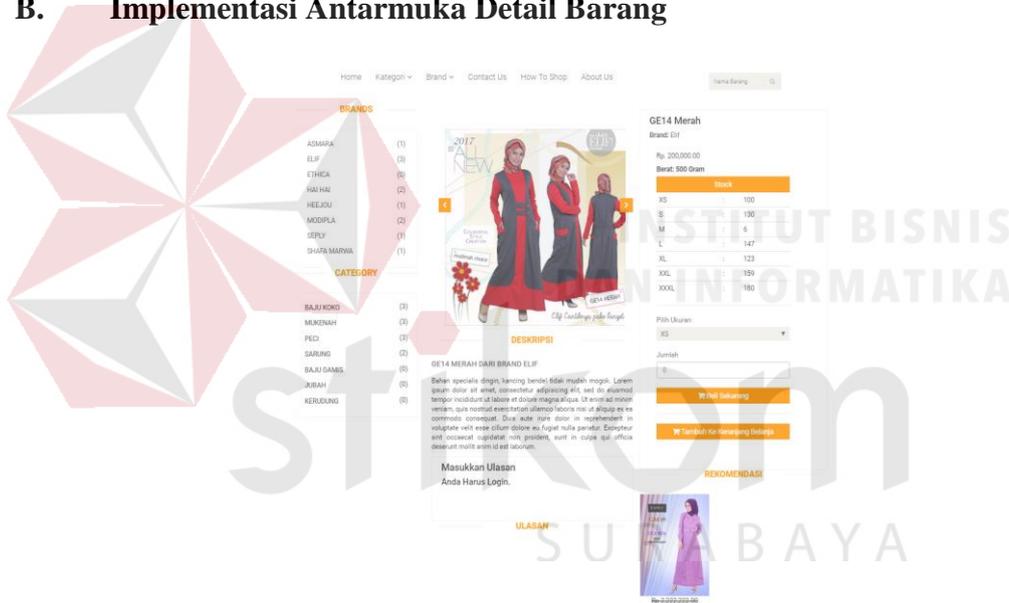
A. Implementasi Antarmuka Beranda

Berikut merupakan implementasi antarmuka sesuai dari perancangan pada bab sebelumnya. Implementasi antarmuka beranda atau *home* memiliki dua buah baris menu. Menu yang terletak di pojok kanan atas terdiri atas tiga buah pilhan yaitu *Login*, *Register*, dan *Cart*. Sedangkan menu yang berada dibawah logo memiliki enam pilihan menu yaitu *Home*, *Kategori*, *Brand*, *Contact Us*, *How To Shop* dan *About Us*. Menu *sorting* terdiri dari dua buah bagian yaitu *sorting* berdasarkan *brand* dan *sorting* berdasarkan kategori. Daftar barang menampilkan gambar, nama dan harga barang.



Gambar 4.15 Antarmuka Beranda

B. Implementasi Antarmuka Detail Barang



Gambar 4.16 Antarmuka Detail Barang

Antarmuka detail barang tidak jauh berbeda dari antarmuka beranda, hanya saja daftar barang berubah menjadi detail barang. Pada detail barang, terdapat delapan buah elemen yaitu foto barang, deskripsi barang, nama barang, data stok barang, *form* pemesanan, tombol pemesanan, ulasan produk dan rekomendasi produk lain.

C. Implementasi Antarmuka *Order* Pelanggan

Pada antarmuka *order* pelanggan terdapat daftar *order* yang pernah dilakukan oleh pelanggan. Dalam daftar *order* tersebut terdapat tanggal, harga, status order, nomor resi, tombol konfirmasi pembayaran dan tombol konfirmasi barang diterima.



Gambar 4.17 Antarmuka *Order* Pelanggan

D. Implementasi Antarmuka *Cart*

Pada antarmuka *order* pelanggan terdapat daftar barang yang ingin dibeli oleh pengguna dan form kurir. Form kurir terdiri dari pilihan kurir, pilihan provinsi, pilihan kabupaten dan deskripsi alamat lengkap pengguna dan pada tampilan *cart* terdapat dua buah tombol yaitu tombol *update* barang untuk mengganti jumlah barang jika salah dan tombol proses untuk memproses pembelian.

1. Harga barang sudah didiskon sesuai membership atau keagenan anda.

No	Name	Size	Price	Quantity	Total
1	Baju Koko Polos Merah	XS	Rp 150,000.00	2	Rp 300,000.00

Berat Total 900 Gram
Grand Total Rp. 300,000.00

Update Keranjang

Dari Kota - Mojokerto, Jawa Timur

Pilih Kurir

JNE

Provinsi Tujuan

Bali

Kabupaten/Kota Tujuan

Bedung - Kabupaten

Alamat Lengkap Tujuan

Masukkan alamat lengkap pengiriman Contoh: Desa Bakalan Wringin Pitu RT 01 RW 02, Kecamatan Balongbendo, Kabupaten Sidoarjo

Proses

Toko Online Terpopuler di Mojokerto

Copyright © 2013 E-SHOPPER Inc. All rights reserved. Designed by [Thamasa](#)

Gambar 4.18 Implementasi Antarmuka *Cart*

E. Implementasi Antarmuka *Billing Info*

Pada tampilan antarmuka *billing info* terdapat daftar barang yang akan dibeli oleh pengguna, pilihan paket kurir dan dua buah tombol. Tombol pilihan paket kurir ini berubah ubah sesuai dengan opsi yang dipilih oleh pengguna.

AL-VEERA
Rt. Kopu Hartan Kel. Tegayoko, Ambar

Home Kategori Brand Contact Us How To Shop About Us

Home Billing

No	Name	Size	Price	Quantity	Sub Total
1	Baju Koko Polos Merah	XS	Rp 150,000.00	2	300,000.00

Berat Total 900 Gram
Grand Total Rp. 300,000.00

Dari : Jawa Timur, Mojokerto
Tujuan : DKI Jakarta, Jakarta Selatan
Alamat Lengkap : FX Sudirman

Jenis Pengiriman :
POS Indonesia (POS)

Paket Kilat Khusus (Paket Kilat Khusus)
Rp 17,000, 1-2 HARI Hari.

Grand Total + Ongkir = Rp.
317,000.00

Kembali Konfirmasi Pesanan

Gambar 4.19 Implementasi Antarmuka *Billing Info*

F. Implementasi Antarmuka *Billing* Berhasil

AL-VEERA
Rt. Kopu Hartan Kel. Tegayoko, Ambar

Home Kategori Brand Contact Us How To Shop About Us

Terimakasih! Pesanan anda sudah dimasukkan kedalam sistem!
Silahkan lakukan pembayaran melalui rekening berikut :
BCD - 0956225554 a.n Si Ferdi SSSSS

[Go back](#)

Gambar 4.20 Implementasi Antarmuka *Billing* Berhasil

Pada tampilan ini terdapat pesan untuk pengguna. Pesan yang muncul bertuliskan terimakasih dan pesan yang kedua bertuliskan daftar rekening yang bisa digunakan sebagai sarana pembayaran.

4.4 Pengujian

Pada bagian ini akan menjelaskan tentang pengujian sistem informasi pemasaran. Pengujian sistem dibagi menjadi 2 bagian pokok yaitu (1) Pengujian Fungsional sistem yang meliputi *black box testing* dan Pengujian Algoritma Apriori; (2) Pengujian non-fungsional sistem yaitu *user acceptance testing*.

4.4.1 Pengujian *Black-box*

Pada sub bagian ini akan dilaksanakan pengujian dari kebutuhan fungsional sistem informasi Pemasaran yang akan menggunakan *black box testing*. Pengujian ini berfokus pada fungsionalitas sistem, pengujian penting dilakukan karena yang dilihat dari sistem pertama kali adalah fungsionalitas sistem seperti masukkan pengguna, aksi pengguna, dan lainnya. Sehingga pada sistem informasi pemasaran membutuhkan pengujian ini. Yang akan digunakan dalam pengujian *black box* ini adalah *validation testing*.

Validation testing ini dilakukan untuk mengetahui apakah fungsionalitas dari sistem informasi yang buat sudah sesuai dengan kebutuhan dari pengguna. Dari 28 fungsional sistem yang di implementasikan dan 28 kasus uji yang diuji semua memiliki status valid. Sehingga dapat disimpulkan bahwa sistem informasi distribusi barang layak untuk digunakan. Dengan uraian kasus uji pada tabel 4.3

Tabel 4.3 *Validation Testing*

Fungsi	Kasus Pengujian	Harapan	Hasil	Status
Sistem dapat menampilkan barang secara detail	Pengguna memilih salah satu barang pada halaman home	Sistem dapat menampilkan detail barang yang dipilih	Sistem menampilkan detail barang yang dipilih	valid

Fungsi	Kasus Pengujian	Harapan	Hasil	Status
Pelanggan dapat mendaftar kedalam sistem	Pelanggan melakukan pendaftaran atau registrasi kedalam sistem	Sistem dapat menampilkan form pendaftaran dan memasukkan data pelanggan kedalam sistem	Sistem menampilkan form pendaftaran dan memasukkan data pelanggan kedalam sistem	valid
Sistem dapat menerima pesanan dari pelanggan	Pelanggan melakukan pemesanan melalui sistem	Sistem dapat menyimpan <i>order</i> yang dilakukan pelanggan	Sistem menyimpan <i>order</i> yang dilakukan pelanggan	valid
Sistem dapat menampilkan jumlah yang harus dibayar pada pelanggan	Pelanggan melihat jumlah yang harus dibayar pada menu <i>order</i>	Sistem dapat menampilkan jumlah yang harus dibayar pada halaman <i>order</i> pelanggan	Sistem menampilkan jumlah yang harus dibayar pada halaman <i>order</i> pelanggan	valid
Sistem dapat digunakan untuk konfirmasi pembayaran	Pelanggan melakukan konfirmasi pembayaran dengan menekan tombol konfirmasi pembayaran	Sistem dapat mengubah status <i>order</i> menjadi verifikasi pembayaran setelah dikonfirmasi	Sistem mengubah status <i>order</i> menjadi verifikasi pembayaran setelah dikonfirmasi	valid
Sistem dapat digunakan untuk menambah barang	Karyawan memasukkan barang kedalam form tambah barang	Sistem dapat menyimpan data barang yang dimasukkan oleh karyawan	Sistem menyimpan data barang yang dimasukkan oleh karyawan	valid
Sistem dapat digunakan untuk menampilkan barang	Karyawan menekan tombol lihat pada daftar barang	Sistem dapat menampilkan detail barang	Sistem menampilkan detail barang	valid
Sistem dapat digunakan untuk edit	Karyawan melakukan edit dengan menekan	Sistem dapat mengubah data barang sesuai masukkan	Sistem mengubah data barang sesuai masukkan	valid

Fungsi	Kasus Pengujian	Harapan	Hasil	Status
barang	tombol edit pada daftar barang	karyawan	karyawan	
Sistem dapat digunakan untuk hapus barang	Karyawan menekan tombol hapus pada salah satu barang pada daftar barang	Sistem dapat menghapus barang sesuai masukkan dari karyawan	Sistem menghapus barang sesuai masukkan dari karyawan	valid
Sistem dapat digunakan untuk menambah kategori	Karyawan melakukan tambah kategori dengan menekan tombol tambah kategori	Sistem dapat membuat kategori baru sesuai masukkan dari karyawan	Sistem membuat kategori baru sesuai masukkan dari karyawan	valid
Sistem dapat digunakan untuk menampilkan kategori	Karyawan menekan menu kategori	Sistem dapat menampilkan seluruh kategori yang telah dibuat	Sistem menampilkan seluruh kategori yang telah dibuat	valid
Sistem dapat digunakan untuk edit kategori	Karyawan melakukan edit pada salah satu kategori yang telah dibuat	Sistem dapat merubah kategori sesuai masukkan dari karyawan	Sistem merubah kategori sesuai masukkan dari karyawan	valid
Sistem dapat digunakan untuk hapus kategori	Karyawan menghapus salah satu kategori yang telah dibuat	Sistem dapat menghapus kategori yang telah dibuat sesuai masukkan dari karyawan	Sistem menghapus kategori yang telah dibuat sesuai masukkan dari karyawan	valid
Sistem dapat digunakan untuk menambah <i>brand</i>	Karyawan membuat <i>brand</i> baru dengan menggunakan tombol tambah <i>brand</i>	Sistem dapat membuat <i>brand</i> baru sesuai dengan masukkan dari karyawan	Sistem membuat <i>brand</i> baru sesuai dengan masukkan dari karyawan	valid
Sistem dapat digunakan untuk	Karyawan memilih menu <i>brand</i> untuk	Sistem dapat menampilkan daftar <i>brand</i>	Sistem menampilkan daftar <i>brand</i>	valid

Fungsi	Kasus Pengujian	Harapan	Hasil	Status
menampilkan <i>brand</i>	menampilkan daftar dranf	setelah menu <i>brand</i> dipilih	setelah menu <i>brand</i> dipilih	
Sistem dapat digunakan untuk edit <i>brand</i>	Karyawan melakukan edit pada salah satu brang yang telah dibuat	Sistem dapat merubah <i>brand</i> sesuai dengan masukkan dari karyawan	Sistem merubah <i>brand</i> sesuai dengan masukkan dari karyawan	valid
Sistem dapat digunakan untuk hapus <i>brand</i>	Karyawan menghapus salah satu <i>brand</i> yarn telah dibuat	Sistem dapat menghapus <i>brand</i> sesuai dengan masukna dari karyawan	Sistem menghapus <i>brand</i> sesuai dengan masukna dari karyawan	valid
Sistem dapat digunakan untuk menampilkan <i>order</i> pelanggan	Karyawan menekan menu <i>order</i> untuk memunculkan seluruh <i>order</i>	Sistem dapat menampilkan <i>order</i> yang telah dilakukan oleh pelanggan	Sistem menampilkan <i>order</i> yang telah dilakukan oleh pelanggan	valid
Sistem dapat digunakan untuk edit <i>order</i> pelanggan	Karyawan melakukan edit pada salah satu item pada suatu <i>order</i>	Sistem dapat merubah detail item pada suatu <i>order</i> sesuai dengan masukkan dari karyawan	Sistem merubah detail item pada suatu <i>order</i> sesuai dengan masukkan dari karyawan	valid
Sistem dapat digunakan untuk mengganti status <i>order</i> pelanggan	Karyawan mengganti status pada salah satu <i>order</i> denganb menekan salah satu tombol status yang diperlukan	Sistem dapat merubah <i>order</i> sesuai dengan tombol <i>order</i> yang dipilih oleh karywan	Sistem merubah <i>order</i> sesuai dengan tombol <i>order</i> yang dipilih oleh karywan	valid
Sistem dapat digunakan untuk mengaktifkan pelanggan baru	Karyawan menekan tombol inactive pada pelanggan yang statusnya inactive	Sistem dapat merubah status pelanggan inactive menjadi active	Sistem merubah status pelanggan inactive menjadi active	valid
Sistem dapat	Karyawan	Sistem dapat	Sistem merubah	valid

Fungsi	Kasus Pengujian	Harapan	Hasil	Status
digunakan untuk menonaktifkan pelanggan	menekan tombol active pada salah satu akun pelanggan yang memiliki status active	merubah status pelanggan dari active menjadi inactive	status pelanggan dari active menjadi inactive	
Sistem dapat digunakan untuk menambah pelanggan baru	Karyawan mendaftarkan pelanggan baru dengan menekan tombol create user	Sistem dapat mendaftarkan pelanggan baru sesuai dengan masukkan dari karyawan	Sistem mendaftarkan pelanggan baru sesuai dengan masukkan dari karyawan	valid

4.4.2 Pengujian Apriori

Pada bagian ini dilakukan pengujian terkait Algoritma Apriori dalam aplikasi apakah sudah berjalan baik atau tidak, dengan membandingkan perhitungan per tahap antara perhitungan manual menggunakan *excel* dengan aplikasi, serta melihat apakah keluaran yang dihasilkan sesuai. Pengujian dilakukan dengan menggunakan 12 data produk dan 20 data transaksi yang ada di *database*. Pada Tabel 4.4 dibawah ini merupakan tabel transaksi yang dipakai.

Tabel 4.4 Tabel Data Transaksi

No.	Daftar Transaksi
1	58, 57, 23, 30
2	57, 23
3	23, 53, 2
4	2, 1
5	31, 54
6	31, 58, 55
7	53, 1, 30
8	31, 54, 56
9	57, 1
10	32, 58, 56
11	30, 58
12	31, 58
13	56, 2, 58

No.	Daftar Transaksi
14	55, 54
15	2, 57
16	54, 31
17	54, 31, 32, 58
18	30, 32, 56
19	54, 31, 1
20	56, 32

Dan berikut adalah hasil pengujian dari tiap tahap yang dilakukan pada Algoritma Apriori.

A. Pengujian Tahap 1

Pada tahap ini dilakukan perhitungan jumlah transaksi yang dilakukan tiap produk pada seluruh transaksi yang terjadi. Tabel 4.5 merupakan hasil perbandingan perhitungan manual dan aplikasi yang telah dilakukan.

Tabel 4.5 Tabel Perbandingan tahap 1

No.	Manual		Aplikasi	
	ID Produk	Jumlah	ID Produk	Jumlah
1	58	7	58	7
2	31	7	31	7
3	54	6	54	6
4	56	5	56	5
5	57	4	57	4
6	2	4	2	4
7	1	4	1	4
8	30	4	30	4
9	32	4	32	4
10	23	3	23	3
11	55	2	55	2
12	53	2	53	2

Dari Tabel 4.5 diatas didapatkan bahwa hasil perhitungan yang dilakukan dengan cara manual dan perhitungan yang dilakukan oleh aplikasi telah sama. Sehingga dapat dikatakan pada tahap ini aplikasi berjalan dengan baik.

B. Pengujian Tahap 2

Tahap kedua dalam Algoritma Apriori adalah menghitung nilai *support* tiap produk terhadap transaksi dengan rumus :

$$\text{Support} = \frac{\text{Jumlah transaksi produk}}{\text{total transaksi}}$$

Dan berikut dihasilkan tabel perbandingan perhitungan manual menggunakan excel dan perhitungan yang dilakukan oleh aplikasi.

Tabel 4.6 Tabel Perbandingan tahap 2

No.	Manual		Aplikasi	
	ID Produk	Nilai Support	ID Produk	Jumlah
1	58	7/20 = 0.35	58	0.35
2	31	7/20 = 0.35	31	0.35
3	54	6/20 = 0.3	54	0.3
4	56	5/20 = 0.25	56	0.25
5	57	4/20 = 0.2	57	0.2
6	2	4/20 = 0.2	2	0.2
7	1	4/20 = 0.2	1	0.2
8	30	4/20 = 0.2	30	0.2
9	32	4/20 = 0.2	32	0.2
10	23	3/20 = 0.15	23	0.15
11	55	2/20 = 0.1	55	0.1
12	53	2/20 = 0.1	53	0.1

Dari Tabel 4.6 diatas didapatkan bahwa hasil perhitungan nilai *support* yang dilakukan dengan cara manual dan perhitungan yang dilakukan oleh aplikasi telah sama. Sehingga dapat dikatakan pada tahap ini aplikasi berjalan dengan baik.

C. Pengujian Tahap 3

Pada tahap ketiga ini dilakukan pengkombinasian produk yang lolos dalam penyaringan berdasarkan batasan yang ditentukan. Tabel 4.7 berikut akan menampilkan perbandingan hasil perhitungan manual dan aplikasi.

Tabel 4.7 Tabel Perbandingan tahap 3

No.	Subjek	Batasan	Jumlah produk yang lolos	Jumlah pasangan
1	Manual	0	12	73
	Aplikasi		12	144
2	Manual	0.1	12	73
	Aplikasi		12	144
3	Manual	0.2	9	40
	Aplikasi		9	81
4	Manual	0.3	3	3
	Aplikasi		3	9

Berdasarkan Tabel 4.7 didapat hasil jumlah pasangan produk yang muncul tidak sama antara perhitungan manual dan aplikasi. Hal ini dikarenakan dalam aplikasi masih terdapat pasangan dengan kombinasi produk yang sama, contohnya produk a dengan produk a, dan kombinasi yang satu dengan yang lain, contohnya kombinasi A-B dengan kombinasi B-A. Meskipun demikian hal ini tidak akan mempengaruhi hasil dari tahap-tahap selanjutnya. Dikarenakan kombinasi-kombinasi dengan produk yang sama tidak akan pernah ada dalam histori transaksi, dan kombinasi yang memiliki pasangan yang sama hanya akan masuk salah satunya saja, contohnya kombinasi A-B dan B-A maka yang dihitung hanya A-B saja.

D. Pengujian Tahap 4

Pada tahap ini dilakukan perhitungan jumlah transaksi tiap pair pada riwayat pembelian pelanggan. Tabel 4.8 berikut merupakan hasil perhitungan jumlah yang dilakukan berdasar produk yang lolos dengan batasan nilai *support* 0.3.

Tabel 4.8 Tabel Perbandingan Tahap 4

No.	Manual		Aplikasi	
	Kombinasi	Jumlah	Kombinasi	Jumlah
1	31, 58	1	31, 58	1
2	31, 54	1	31, 54	1

Pada Tabel 4.8 didapat bahwa hasil perhitungan yang dilakukan dengan cara manual dan perhitungan yang dilakukan oleh aplikasi telah sama. Sehingga dapat dikatakan pada tahap ini aplikasi berjalan dengan baik.

E. Pengujian Tahap 5

Pada tahap ini dilakukan perhitungan nilai *confidence* tiap pasangan dengan rumus:

$$\text{Confidence} = P(B | A) = \frac{\text{Jumlah Transaksi mengandung A dan B}}{\text{Jumlah Transaksi mengandung A}}$$

Tabel berikut merupakan hasil perbandingan perhitungan nilai *confidence* tiap pasangan.

Tabel 4.9 Tabel Perbandingan Tahap 5

No.	Manual		Aplikasi	
	kombinasi	confidence	kombinasi	confidence
1	31-58	1/7 = 0.14	31-58	0.143
2	58-31	1/7 = 0.14	58-31	0.143
3	54-31	1/6 = 0.17	54-31	0.167

No.	Manual		Aplikasi	
	kombinasi	confidence	kombinasi	confidence
4	31-54	$1/7 = 0.14$	31-54	0.143

Pada Tabel 4.9 didapat bahwa hasil perhitungan yang dilakukan dengan cara manual dan perhitungan yang dilakukan oleh aplikasi telah sama. Sehingga dapat dikatakan pada tahap ini aplikasi berjalan dengan baik.

Kesimpulan yang didapat pada pengujian Algoritma Apriori pada aplikasi ini didapat bahwa hampir semua fungsi bekerja dengan baik. Hanya fungsi pada tahap 3 saja yang perlu lebih diefisienkan lagi.

4.4.3 Pengujian UAT

Pada bagian ini akan dibahas mengenai UAT (*User Acceptance Testing*) yang memiliki tujuan memeriksa sistem informasi yang dibuat terhadap kebutuhan proses bisnis yang telah didefinisikan pada bab sebelumnya. Dengan pengujian ini dapat diketahui apakah sistem informasi yang di buat sudah bebas dari kesalahan, oleh karena itu digunakan pengujian UAT.

Pengujian UAT dilakukan dengan membuat daftar kuisisioner atau daftar pertanyaan yang dapat mewakili semua kebutuhan yang telah didefinisikan pada analisis kebutuhan. Daftar pertanyaan akan ditanyakan saat karyawan atau pengguna mencoba sistem informasi pemasaran secara langsung. Berikut merupakan beberapa pengguna yang akan mengisi daftar pertanyaan pada Tabel 4.10. Setelah responden melakukan simulasi dan menjawab pertanyaan yang dibuat, maka dapat diketahui apakah sistem dapat diterima atau tidak. Kriteria yang akan digunakan adalah *usability*, *functional correctness and completeness*, dan *performance*.

Tabel 4.10 Kriteria dan Pertanyaan UAT

No	Kriteria	Definisi	Daftar Pertanyaan
1.	<i>Usability</i>	Evaluasi produk dengan mengujinya pada pengguna yang mewakili. Bertujuan untuk menentukan kepuasan pengguna terhadap aplikasi yang dibuat.	1. Sistem informasi pemasaran mudah digunakan
			2. Tampilan sistem informasi pemasaran produk nyaman ketika digunakan
			3. Sistem informasi pemasaran mempermudah dalam mencari persediaan barang tertentu
			4. Sistem informasi pemasaran mempermudah dalam mencari <i>order</i> tertentu
2.	<i>Functional Correctness and Completeness</i> <i>and</i> <i>Completeness</i>	<i>Functional Correctness and Completeness</i> adalah pengujian untuk mengetahui apakah sistem yang dibuat sudah sesuai kebutuhan yang diinginkan pengguna dan telah didefinisikan pada spesifikasi kebutuhan	1. Pengelolaan Data Barang sesuai dengan kebutuhan.
			2. Pengelolaan Kategori sesuai dengan kebutuhan.
			3. Pengelolaan <i>Brand</i> sesuai dengan kebutuhan.
			4. Pengelolaan <i>order</i> sesuai dengan kebutuhan.
			5. Aktivasi akun pelanggan sesuai dengan kebutuhan.
			6. Nonaktifkan akun pelanggan sesuai dengan kebutuhan.
			7. Membuat akun pelanggan sesuai

No	Kriteria	Definisi	Daftar Pertanyaan
			dengan kebutuhan.
			8. Lihat Barang sesuai dengan kebutuhan.
			9. Registrasi sesuai dengan kebutuhan.
			10. Konfirmasi pembayaran sesuai dengan kebutuhan.
			11. Lihat order sesuai dengan kebutuhan.
			12. Order Barang sesuai dengan kebutuhan.
3.	<i>Performance</i>	Penentuan kecepatan atau efektivitas perangkat lunak.	1. Sistem informasi pemasaran mempercepat pencarian detail barang

Setelah kriteria dan pertanyaan ditentukan sesuai dengan kriteria yang diharapkan, maka tahap selanjutnya adalah menganalisis hasil wawancara yang ditunjukkan pada Tabel 4.11.

Tabel 4.11 Hasil Wawancara *User Acceptance Testing*

No	Pernyataan	Kode Task	Jawaban				
			Sangat Tidak Setuju	Tidak Setuju	Netral	Setuju	Sangat Setuju
1.	Sistem informasi pemasaran mudah digunakan	UAT-01-01	-	1	3	11	-
2.	Tampilan sistem informasi pemasaran nyaman ketika digunakan	UAT-01-02	-	-	15	-	-
3.	Sistem informasi pemasaran mempermudah	UAT-01-03	-	-	4	11	-

No	Pernyataan	Kode Task	Jawaban				
			Sangat Tidak Setuju	Tidak Setuju	Netral	Setuju	Sangat Setuju
	dalam mencari persediaan barang tertentu						
4.	Sistem informasi pemasaran mempermudah dalam mencari <i>order</i> tertentu	UAT-01-04	-	-	10	5	-
5.	Pengelolaan Data Barang sesuai dengan kebutuhan.	UAT-02-01	-	-	10	-	5
6.	Pengelolaan Kategori sesuai dengan kebutuhan.	UAT-02-02	-	-	10	5	-
7.	Pengelolaan <i>Brand</i> sesuai dengan kebutuhan.	UAT-02-03	-	-	10	5	-
8.	Pengelolaan <i>order</i> sesuai dengan kebutuhan.	UAT-02-04	-	-	10	5	-
9.	Aktivasi akun pelanggan sesuai dengan kebutuhan.	UAT-02-05	-	-	10	5	-
10.	Nonaktifkan akun pelanggan sesuai dengan kebutuhan.	UAT-02-06	-	-	10	5	-
11.	Membuat akun pelanggan sesuai dengan kebutuhan.	UAT-02-07	-	-	8	7	-
12.	Lihat Barang sesuai dengan kebutuhan.	UAT-02-08	-	-	2	8	5

No	Pernyataan	Kode Task	Jawaban				
			Sangat Tidak Setuju	Tidak Setuju	Netral	Setuju	Sangat Setuju
13.	Registrasi sesuai dengan kebutuhan.	UAT-02-09	-	-	-	15	-
14.	Konfirmasi pembayaran sesuai dengan kebutuhan.	UAT-02-10	-	-	4	8	3
15.	Lihat order sesuai dengan kebutuhan.	UAT-02-11	-	-	-	15	-
16.	Order Barang sesuai dengan kebutuhan.	UAT-02-12	-	-	5	6	4
17.	Sistem informasi pemasaran mempercepat pencarian detail barang	UAT-03-01	-	-	-	15	-
Total			0	1	111	126	17

Menghitung total nilai berdasarkan dari jawaban 15 orang responden

Tabel 4.12 Tabel Hasil Skala *Likert*

Jawaban	Bobot Nilai	Jumlah Jawaban (n)	Hasil
Sangat Setuju	5	17	75
Setuju	4	126	504
Netral	3	111	333
Tidak Setuju	2	1	2
Sangat Tidak Setuju	1	0	0
Total			914

Menghitung Nilai Y

$Y = \text{Jumlah Skala Likert} \times \text{Jumlah Responden} \times \text{Jumlah Pertanyaan}$

$$Y = 5 \times 15 \times 17$$

$$Y = 1275$$

Menghitung Nilai Persentase

$$\text{Rumus Index} = (\text{Total Nilai} / Y) \times 100\%$$

$$\text{Rumus Index} = (914/1275) \times 100\%$$

$$\text{Rumus Index} = 0.72$$

Dari perhitungan diatas didapat nilai 0.72 atau 72%, yang artinya nilai penerimaan responden terhadap sistem informasi yang telah dibuat sebesar 72%. Karena nilainya lebih dari 50% maka bisa dikatakan sistem informasi pemasaran ini diterima oleh responden.



BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil uji coba dan evaluasi yang telah dilakukan terhadap aplikasi rekomendasi produk menggunakan Algoritma Apriori studi kasus toko Al-Veera Jember, maka dapat ditarik kesimpulannya yaitu aplikasi rekomendasi produk dapat melakukan transaksi penjualan dengan baik. Sistem informasi pemasaran Al-Veera dapat mengelola order dengan baik. Sistem informasi pemasaran Al-Veera dapat menampilkan rekomendasi produk yang didapat dari perhitungan apriori dengan baik. aplikasi rekomendasi produk dapat menampilkan detail barang dengan baik. Dalam pengujian Apriori setiap tahap teruji dengan baik. Diperlukan data transaksi yang memiliki jumlah produk pembelian dengan minimal 2 produk dalam satu transaksi dalam jumlah besar untuk melakukan apriori.

5.2 Saran

Aplikasi Rekomendasi Produk ini masih terdapat kekurangan. Adapun beberapa saran yang dapat digunakan untuk pengembangan aplikasi ini. Adapun penambahan fitur yang diperlukan yaitu pemberitahuan pada sistem informasi pemasaran terhadap status pesannya dapat dilakukan via email agar pelanggan dapat mengetahui status pesannya. Penambahan fitur Tracking barang agar pelanggan dapat mengetahui keberadaan pesannya. Penambahan fitur refund agar pelanggan dapat melakukan refund dengan cepat.

DAFTAR PUSTAKA

- Aryanto, A. (2012). *Pembangunan Sistem Penjualan Online Pada Toko Indah Jaya Furniture Surakarta*. Journal Speed – Sentra Penelitian Engineering dan Edukasi – Volume 4 No 4 – 2012.
- Granito, H. (2008). *Panduan Mendirikan dan Mengelola Distro Clothing*. Yogyakarta: Media Pressindo.
- Hambling, B. and Goethem P.V. 2013. *User Acceptance Test A step-by-step guide*. BCS Learning and Development Ltd. Swindon, United Kingdom.
- Harahap, N. S. (2012). *Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android*. Bandung: Informatika.
- Irliana, N. (2013). *Market Basket Analysis pada perusahaan retail menggunakan algoritma apriori dan sales forecasting*. JURNAL TRANSFORMATIKA, Volume 11, No.1, Juli 2013 : 13 – 22.
- Janner, S. (2010). *Rekayasa Perangkat Lunak*. Yogyakarta: Andi.
- Uma Maheswari, K. (2018). *Finding Frequent Item Set using Apriori Algorithm for Online Shopping (Ekart)*. International Journal of Engineering Science and Computing.
- Nugroho, B. (2010). *Aplikasi Pemrograman Web Dinamis Dengan PHP & mySQL*. Yogyakarta: Gava Media.
- Pressman, R. (2015). *Rekayasa Perangkat Lunak: Pendekatan Praktisi Buku I*. Yogyakarta: Andi.

Shawn, Cohan. (2012). *Web Statis Dan Dinamis*. <http://www.informasi-baru.com/2007/12/web-statis-dan-dinamis.html>. [Diakses Pada tanggal 7 April 2018]

Sibero, A. F. (2011). *Kitab Suci Web Programming*. Yogyakarta: mediakom.

Sutabri, T. (2012). *Analisis Sistem Informasi*. Yogyakarta: Andi.

softwaretestinghelp, 2018. what-is-user-acceptance-testing-uat. [Online] Available at: <http://www.softwaretestinghelp.com/what-is-user-acceptance-testing-uat/> [Diakses 17 Februari 2018].

Triyanto, W.A. 2014. Association Rule Mining untuk Penentuan Rekomendasi Promosi Produk. Jurnal SIMETRIS, ISSN: 2252-4983. Vol. 5, No. 2, pp. 121-126.

