



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - K141502

RANCANG BANGUN MODUL BI (*BUSINESS INTELLIGENCE*) BERDASARKAN *BALANCED SCORECARD* PADA APLIKASI *ENTERPRISE RESOURCE PLANNING RETAIL*

RENANDA AGUSTIANTORO
NRP 5113100028

Dosen Pembimbing
Prof. Drs. Ec. Ir. Riyanarto Sarno, M.Sc., Ph.D.
Dwi Sunaryono, S.Kom, M.Kom.

JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2018

[Halaman ini sengaja dikosongkan]



TUGAS AKHIR - K141502

RANCANG BANGUN MODUL BI (*BUSINESS INTELLIGENCE*) BERDASARKAN *BALANCED SCORECARD* PADA APLIKASI *ENTERPRISE RESOURCE PLANNING RETAIL*

RENANDA AGUSTIANTORO
NRP 5113100028

Dosen Pembimbing
Prof. Drs. Ec. Ir. Riyanarto Sarno, M.Sc., Ph.D.
Dwi Sunaryono, S.Kom, M.Kom.

JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2018

[Halaman ini sengaja dikosongkan]



FINAL PROJECT - K141502

**DESIGN AND IMPLEMENTATION OF BUSINESS
INTELLIGENCE MODULES BASED ON
BALANCED SCORECARD IN RETAIL
ENTERPRISE RESOURCE PLANNING**

**RENANDA AGUSTIANTORO
NRP 5113100028**

Supervisor

Prof. Drs. Ec. Ir. Riyanarto Sarno, M.Sc., Ph.D.

Dwi Sunaryono, S.Kom, M.Kom.

Department of Informatics
Faculty of Information Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2018

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN

RANCANG BANGUN MODUL BI (BUSINESS INTELLIGENCE) BERDASARKAN BALANCED SCORECARD PADA APLIKASI ENTERPRISE RESOURCE PLANNING RETAIL

TUGAS AKHIR

Diajukan Guna Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Bidang Studi Manajemen Informasi
Program Studi S-1 Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh:

Renanda Agustiantoro

NRP: 5113 100 028

Disetujui oleh Dosen Pembimbing Tugas Akhir:

Prof. Drs. Ec. Ir. Riyanarto Sarno, M.Sc., Ph.D.

NIP: 19590803 198601 1 001

(pembimbing 1)

Dwi Sunaryono, S.Kom., M.Kom.

NIP: 19720528 199702 1 001

(pembimbing 2)



**SURABAYA
MEI 2018**

[Halaman ini sengaja dikosongkan]

RANCANG BANGUN MODUL BI (*BUSINESS INTELLIGENCE*) BERDASARKAN *BALANCED SCORECARD* PADA APLIKASI *ENTERPRISE RESOURCE PLANNING RETAIL*

Nama Mahasiswa : Renanda Agustiantoro
NRP : 5113 100 028
Jurusan : Teknik Informatika FTIf-ITS
Dosen Pembimbing 1 : Prof. Drs. Ec. Ir. Rivanarto Sarno,
M.Sc., Ph.D.
Dosen Pembimbing 2 : Dwi Sunaryono, S.Kom., M.Kom.

ABSTRAKSI

Perkembangan dunia bisnis yang sangat pesat mengharuskan penggunaan teknologi untuk mempermudah dan mempercepat proses. *Enterprise Resource Planning* atau *ERP* merupakan contoh dari aplikasi yang menjadi pilihan pada pebisnis besar khususnya yang bergerak di bidang manufaktur. Namun sekarang *ERP* juga mulai merambah ke usaha kecil dan menengah yang bergerak dibidang *retail*. Meskipun begitu, kurangnya pengetahuan dalam ilmu-ilmu bisnis membuat *ERP* menjadi kurang optimal dikalangan usaha menengah. Untuk itu, Tugas Akhir ini adalah sebagai pembantu para pemilik usaha kecil dan menengah dalam mengambil langkah-langkah bisnis dengan cara memberikan rekomendasi-rekomendasi kepada mereka, disinilah *Modul Business Intelligence* berperan penting.

Rekomendasi yang ditawarkan pada modul *BI* ini diantaranya adalah rekomenasi pemilihan *supplier*. Pemilihan ini menggunakan bobot pada indikator sebagai metodenya. Selain pemilihan *supplier*, modul ini juga memberikan rekomendasi kapan dan berapa jumlah *re-stock* yang harus dilakukan. Rekomendasi yang diberikan oleh modul *BI* dihitung menggunakan rumus-rumus terpercaya seperti *EOQ* dan *ROP* sehingga menghasilkan rekomendasi paling ekonomis dan optimal.

Di samping itu, modul ini juga memberikan informasi kepada pengguna tentang keadaan keuangan perusahaannya serta dilengkapi dengan peramalan penjualan untuk periode selanjutnya.

Kata Kunci: *Business Intelligence, Enterprise Resource Planning, Retail, Supply Chain Management.*

DESIGN AND IMPLEMENTATION OF BUSINESS INTELLIGENCE MODULES BASED ON BALANCED SCORECARD IN RETAIL ENTERPRISE RESOURCE PLANNING

Student Name : Renanda Agustiantoro
Student ID : 5113 100 028
Major : Informatics Department FTIf-ITS
Advisor 1 : Prof. Drs. Ec. Ir. Rivanarto Sarno, M.Sc., Ph.D.
Advisor 2 : Dwi Sunaryono, S.Kom, M.Kom.

ABSTRACT

The rapid development of the business world requires the use of technology to facilitate and accelerate the process. Enterprise Resource Planning or ERP is an example of an application that becomes an option in large businesses especially engaged in manufacturing. But now ERP also began to penetrate into small and medium enterprises engaged in retail. However, lack of knowledge in the business sciences makes ERP less than optimal amongst medium-sized enterprises. To that end, this Final Project is to assist small and medium business owners in taking business steps by providing recommendations to them, here Business Module Intelligence plays an important role.

The recommendations offered in this BI module include recommendation of supplier selection. This selection uses the weight of the indicator as the method. In addition to supplier selection, this module also provides recommendations when and how many re-stock to do. Recommendations provided by the BI module are calculated using reliable formulas such as EOQ and ROP to produce the most economical and optimal recommendations. In addition, this module also provides information to users about the financial state of the company and is equipped with sales forecasting for the next period.

Keywords: Business Intelligence, Enterprise Resource Planning, Retail, Supply Chain Management.

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Puji syukur kepada Tuhan Yang Maha Esa atas segala karunia dan rahmat-Nya penulis dapat menyelesaikan Tugas Akhir yang berjudul:

RANCANG BANGUN MODUL BI (*BUSINESS INTELLIGENCE*) BERDASARKAN *BALANCED SCORECARD* PADA APLIKASI *ENTERPRISE RESOURCE PLANNING RETAIL*

Melalui lembar ini, penulis ingin menyampaikan ucapan terimakasih dan penghormatan yang sebesar-besarnya kepada:

1. Bapak, Ibu, kakak dan keluarga besar yang selalu memberikan dukungan penuh untuk menyelesaikan Tugas Akhir ini.
2. Prof. Drs. Ec. Ir. Riyanarto Sarno, M.sc., Ph.D., beserta keluarga selaku dosen pembimbing 1 yang telah banyak membantu, membimbing, bahkan memotivasi penulis untuk menyelesaikan Tugas Akhir ini.
3. Bapak Dwi Sunaryo selaku dosen pembimbing 2 yang telah banyak memberikan semangat, motivasi, serta arahan kepada penulis dalam menyelesaikan Tugas Akhir ini.
4. Tim ERP, selaku tim yang membantu dalam pengerjaan Tugas Akhir ini.
5. Ulululu Group, selaku tim yang membantu dalam pengerjaan Tugas Akhir ini.
6. Bapak dan Ibu dosen Jurusan Teknik Informatika ITS lainnya yang telah banyak menyampaikan ilmu dan bimbingan yang tak ternilai harganya bagi penulis.

7. Teman-teman Laboratorium Manajemen Informasi sekaligus teman-teman seperjuangan anak didik Tugas Akhir Prof. Riyanarto Sarno yaitu Ical, Andi, Zaenal, dan Faizal.
8. Teman-teman angkatan 2013 Departemen Teknik Informatika ITS yang telah menemani dan mendukung penulis selama kuliah.
9. Bapak Mujib sebagai penjaga parkir Departemen Teknik Informatika yang telah menjaga sepeda motor penulis selama pengerjaan Tugas Akhir ini.
10. Serta pihak-pihak lain yang namanya tidak dapat penulis sebutkan satu per satu.

Bagaimanapun juga penulis telah berusaha sebaik-baiknya dalam menyusun Tugas Akhir ini, namun penulis mohon maaf apabila terdapat kekurangan yang penulis lakukan. Kritik dan saran yang membangun dapat disampaikan sebagai bahan perbaikan selanjutnya.

Surabaya, Mei 2018

Renanda Agustiantoro

DAFTAR ISI

LEMBAR PENGESAHAN.....	vii
ABSTRAKSI.....	ix
ABSTRACT.....	xi
KATA PENGANTAR.....	xiii
DAFTAR ISI.....	xv
DAFTAR GAMBAR.....	xix
DAFTAR TABEL.....	xxi
DAFTAR KODE SUMBER.....	xxiii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Permasalahan.....	2
1.3 Batasan Permasalahan.....	2
1.4 Tujuan.....	3
1.5 Manfaat.....	3
1.6 Metodologi.....	3
1.7 Sistematika Penulisan.....	5
BAB II DASAR TEORI.....	7
2.1. ERP (Enterprise Resource Planning).....	7
2.2. <i>RETAIL</i>	8
2.2.1. <i>Inventory</i>	8
2.2.2. <i>Sales</i>	9
2.2.3. <i>Purchase</i>	9
2.2.4. <i>General Ledger dan Accounting</i>	10
2.2.5. <i>Business Intelligence</i>	10
2.3. <i>SUPPLY CHAIN MANAGEMENT</i>	10
2.4. <i>BALANCED SCORECARD</i>	11
2.5. <i>DECISION SUPPORT SYSTEM</i>	11

BAB III ANALISIS DAN PERANCANGAN SISTEM.....	13
3.1. Analisis.....	13
3.1.1. Analisis Permasalahan.....	13
3.1.2. Analisis Kebutuhan	14
3.2. Deskripsi Umum Sistem.....	15
3.2.1. <i>Input</i>	16
3.2.2. Proses.....	16
3.2.3. Kasus Penggunaan.....	28
3.3. Perancangan Sistem.....	41
3.3.1. Perancangan Basis Data	42
3.3.2. Perancangan Antar Muka	43
BAB IV IMPLEMENTASI.....	47
4.1. Lingkungan Pengembangan Sistem	47
4.2. Implementasi Modul <i>Business Intelligence</i>	47
4.2.1. Membuat Tabel-Tabel yang Dibutuhkan.....	47
4.2.2. Implementasi Perhitungan ROP	50
4.2.3. Implementasi Perhitungan EOQ.....	54
4.2.1. Implementasi <i>Supplier Selection</i>	55
4.2.2. Implementasi Rangkuman Laporan Keuangan....	57
4.2.3. Implementasi <i>Forecasting</i> Penjualan	60
4.2.4. Implementasi Perhitungan Harga Optimal	62
4.3. Implementasi Antarmuka	68
4.3.1. Antarmuka Halaman Utama	68
4.3.2. Antarmuka tampilan halaman Config Autostock	68
BAB V PENGUJIAN DAN EVALUASI	69
5.1. Lingkungan Pengujian.....	69
5.2. Skenario Pengujian.....	69
5.2.1. Pengujian Fungsionalitas.....	69

5.3.	Evaluasi Pengujian	83
5.3.1.	Evaluasi Pengujian Fungsionalitas	83
BAB VI KESIMPULAN DAN SARAN.....		85
6.1.	Kesimpulan.....	85
6.2.	Saran.....	86
DAFTAR PUSTAKA.....		87
LAMPIRAN		89
BIODATA PENULIS.....		91

[Halaman ini sengaja dikosongkan]

DAFTAR GAMBAR

Gambar 3.1 Deskripsi Umum Sistem.....	15
Gambar 3.2 Struktur model <i>AHP</i> evaluasi kinerja <i>supplier</i>	18
Gambar 3.3 <i>Dupont Analysis Framework</i>	23
Gambar 3.4 Diagram Kasus Penggunaan	29
Gambar 3.5 Diagram aktivitas use case UC-1.....	31
Gambar 3.6 Diagram aktivitas use case UC-2.....	33
Gambar 3.7 Diagram aktivitas use case UC-3.....	35
Gambar 3.8 Diagram aktivitas use case UC-4.....	37
Gambar 3.9 Diagram aktivitas use case UC-5.....	39
Gambar 3.10 Diagram aktivitas use case UC-6.....	41
Gambar 3.11 Perancangan tampilan halaman utama	44
Gambar 3.12 Perancangan tampilan halaman pembelian.....	45
Gambar 5.1 Hasil Perhitungan oleh Sistem.....	78
Gambar 5.2 Hasil Perhitungan EOQ oleh Sistem	79
Gambar 5.3 Hasil Perhitungan ROP oleh Sistem	80
Gambar 5.4 Hasil Perhitungan PED dan OP oleh Sistem	82
Gambar A.1 Antarmuka Halaman Utama	89
Gambar A.2 Antarmuka Halaman Config Autostock	90

[Halaman ini sengaja dikosongkan]

DAFTAR TABEL

Tabel 3.1 Daftar Kebutuhan Fungsional Sistem.....	14
Tabel 3.2 Skala Fundamental Saaty	19
Tabel 3.3 Perbandingan Pasangan Kriteria	19
Tabel 3.4 Tabel Matriks Perbandingan Berpasangan Memilih <i>Supplier</i>	20
Tabel 3.5 Matriks Perbandingan yang Dinormalisasi	20
Tabel 3.6 Matriks Perbandingan yang Dibobot	21
Tabel 3.7 Nilai Hasil Vektor Konsistensi	21
Tabel 3.8 Nilai Indeks Acak (<i>random index</i>)	22
Tabel 3.9 Nilai Bobot Indikator	22
Tabel 3.10 Daftar Kode Kasus Penggunaan.....	28
Tabel 3.11 Spesifikasi Kasus Penggunaan Memberikan Pilihan Supplier Optimal	30
Tabel 3.12 Spesifikasi Kasus Penggunaan Memberikan Jumlah Re-Stock yang Ekonomis	32
Tabel 3.13 Spesifikasi Kasus Penggunaan Memberikan Status Ketersediaan Barang.....	34
Tabel 3.14 Spesifikasi Kasus Penggunaan Memberikan Informasi Dari Laporan Keuangan	36
Tabel 3.15 Spesifikasi Kasus Penggunaan Memberikan Informasi Tentang Prediksi Penjualan	38
Tabel 5.1 Pengujian Menampilkan Rekomendasi Supplier.....	70
Tabel 5.2 Pengujian Menampilkan Jumlah <i>Re-stock</i>	71
Tabel 5.3 Pengujian Menampilkan Batas <i>Re-stock</i>	72
Tabel 5.4 Pengujian Menampilkan Rangkuman Laporan Keuangan.....	73
Tabel 5. 5 Pengujian Menampilkan Prediksi Penjualan	74

[Halaman ini sengaja dikosongkan]

DAFTAR KODE SUMBER

Kode Sumber 4.1 Membuat Tabel BI.....	49
Kode Sumber 4.2 Membuat Data ROP Baru.....	50
Kode Sumber 4.3 Update Data ROP.....	53
Kode Sumber 4.4 Mengambil hasil ROP.....	53
Kode Sumber 4.5 Implementasi Perhitungan EOQ.....	55
Kode Sumber 4.6 Implementasi <i>Supplier Selection</i>	57
Kode Sumber 4.7 Implementasi Perhitungan <i>Return of Assets</i> ...	58
Kode Sumber 4.8 Implementasi Perhitungan <i>ROI</i>	59
Kode Sumber 4.9 Implementasi Perhitungan <i>Return of Equity</i> ...	60
Kode Sumber 4.10 Implementasi <i>Forecast</i> Penjualan.....	62

[Halaman ini sengaja dikosongkan]

BAB I

PENDAHULUAN

Pada bab ini akan dipaparkan mengenai garis besar Tugas Akhir yang meliputi latar belakang, tujuan, rumusan dan batasan permasalahan, metodologi pembuatan Tugas Akhir, dan sistematika penulisan.

1.1 Latar Belakang

Kegiatan bisnis sangat membantu usaha-usaha pemenuhan kebutuhan masyarakat oleh perusahaan. Bisnis ini meliputi semua aspek kegiatan untuk menyalurkan barang-barang melalui saluran produktif, dari membeli barang mentah sampai menjual barang jadi. Pedagang yang berperan melakukan pembelian serta penjualan merupakan jalur penghubung antara produsen dengan konsumen.

Perkembangan dunia bisnis yang pesat mengharuskan penggunaan teknologi untuk mempermudah dan mempercepat proses. Maka dari itu dikembangkan aplikasi-aplikasi yang bertujuan membantu dalam dunia bisnis. *Enterprise Resource Planning* atau yang sering disingkat ERP merupakan contoh dari aplikasi yang dikembangkan dan sangat populer di kalangan pebisnis-pebisnis besar saat ini. *Enterprise Resource Planning* merupakan sistem informasi yang ditujukan bagi perusahaan manufaktur maupun jasa yang berperan mengintegrasikan dan mengotomatiskan proses bisnis yang berhubungan dengan aspek operasi, produksi maupun distribusi di perusahaan yang bersangkutan. Dengan demikian, diharapkan setelah menerapkan ERP pada perusahaan dapat mengoptimalkan biaya dan waktu.

Pada tahun 2016 lalu telah dibuat sebuah aplikasi ERP yang bernama EZERP [1]. Aplikasi ini mempunyai modul yang cukup lengkap dan fokus untuk perusahaan yang bergerak di bidang manufakturing. Hal tersebut mengakibatkan penggunaan yang rumit bagi perusahaan yang bergerak di bagian pengecer dan

distributor. Selain itu, pada aplikasi ERP ini masih belum ada modul yang membantu pemilik perusahaan dalam menentukan langkah bisnis selanjutnya. Maka dari itu, perlu adanya penambahan Modul BI (*Business Intelligence*) yang berdasarkan *Balanced Scorecard* sehingga dapat membantu perusahaan dalam menilai kesehatan perusahaannya dan menentukan strategi selanjutnya untuk memenuhi tujuan jangka pendek dan jangka panjang.

1.2 Rumusan Permasalahan

Rumusan masalah yang diangkat dalam Tugas Akhir ini antara lain:

1. Bagaimana metode untuk memilih *supplier* yang optimal?
2. Bagaimana cara menentukan jumlah dan waktu pembelian stok barang yang ekonomis?
3. Bagaimana cara memberikan informasi kepada pengguna dari data laporan keuangan?
4. Bagaimana mengaplikasikan peramalan penjualan dalam aplikasi ERP Retail?
5. Bagaimana cara menentukan harga yang optimal?

1.3 Batasan Permasalahan

Permasalahan yang dibahas dalam Tugas Akhir ini memiliki beberapa batasan, antara lain:

1. Pemilihan *supplier* hanya menggunakan 5 indikator, yaitu *quality*, *cost*, *delivery*, *responsiveness*, dan *flexibility*.
2. Periode waktu yang digunakan pada laporan keuangan merupakan periode waktu harian.
3. Penjualan yang diramal merupakan total penjualan dari semua barang.

1.4 Tujuan

Tujuan dari pembuatan Tugas Akhir ini adalah:

1. Membuat modul *Business Intelligence* yang dapat memberikan rekomendasi bagi pengguna dalam pengambilan keputusan untuk rencana bisnis selanjutnya.
2. Membuat *dashboard* yang dapat memberikan informasi pengguna tentang keadaan perusahaan sekarang dan kemungkinan yang akan datang.

1.5 Manfaat

Manfaat dari hasil tugas akhir ini adalah membangun sebuah modul *Business Intelligence* pada aplikasi ERP (*Enterprise Resource Planning*) sebagai salah satu kontribusi dalam pengembangan masyarakat, terutama UKM-UKM yang berfokus pada penjualan secara *retail*. Dengan dibangunnya aplikasi ini, diharapkan dapat mempermudah perusahaan dalam mengambil keputusan untuk bisnis selanjutnya.

1.6 Metodologi

Langkah-langkah yang ditempuh dalam pengerjaan Tugas Akhir ini yaitu:

a. Penyusunan proposal tugas akhir

Proposal tugas akhir ini berisi latar belakang pembuatan tugas akhir, rumusan masalah, batasan masalah, tujuan pembuatan, manfaat, ringkasan tugas akhir, metodologi hingga jadwal kegiatan pembuatan tugas akhir. Proposal tugas akhir juga berisi tinjauan pustaka yang digunakan sebagai referensi pembuatan tugas akhir ini.

b. Studi literatur

Studi literatur yang dilakukan berfokus pada hal-hal yang bersangkutan dengan Modul BI, yaitu mengenai ERP, *Business Intelligence*, *Supply Chain Management*, *Balanced Scorecard* dan *Decision Support System*.

c. Analisis dan desain perangkat lunak

Tahap ini meliputi perancangan sistem berdasarkan studi literatur dan pembelajaran konsep teknologi dari perangkat lunak yang ada. Tahap ini mendefinisikan alur dari implementasi. Langkah-langkah yang dikerjakan juga didefinisikan pada tahap ini. Pada tahapan ini dibuat prototipe sistem, yang merupakan rancangan dasar dari sistem yang akan dibuat. Serta dilakukan desain suatu sistem dan desain proses-proses yang ada.

d. Implementasi perangkat lunak

Implementasi perangkat lunak ini dibangun dengan bahasa pemrograman PHP dan database MySQL. Selain itu untuk memudahkan pengembangan dan perbaikan di tahap selanjutnya, maka dibangun dengan kerangka kerja *CodeIgniter*.

e. Pengujian dan evaluasi

Tahap ini membahas bagaimana hasil perancangan perangkat lunak yang telah dibuat, kemudian diimplementasikan dalam bentuk tampilan *dashboard* dalam sistem ERP Retail untuk dilakukan uji coba. Pengujian akan dilakukan dengan melihat seberapa besar pengaruh informasi yang diberikan oleh *dashboard* BI ini terhadap kondisi keuangan perusahaan tersebut. Tahap ini juga dimaksudkan untuk mengevaluasi jalannya sistem, mencari masalah yang mungkin timbul dan mengadakan perbaikan jika terdapat kesalahan.

f. Penyusunan Buku Tugas Akhir

Pada tahap ini dilakukan penyusunan laporan yang menjelaskan dasar teori dan metode yang digunakan dalam tugas akhir ini serta hasil dari implementasi aplikasi perangkat lunak yang telah dibuat. Sistematika penulisan buku tugas akhir secara garis besar antara lain:

1. Pendahuluan

- a. Latar Belakang
- b. Rumusan Masalah

- c. Batasan Tugas Akhir
 - d. Tujuan
 - e. Metodologi
 - f. Sistematika Penulisan
2. Tinjauan Pustaka
 3. Desain dan Implementasi
 4. Pengujian dan Evaluasi
 5. Kesimpulan dan Saran
 6. Daftar Pustaka

1.7 Sistematika Penulisan

Buku Tugas Akhir ini bertujuan untuk mendapatkan gambaran dari pengerjaan Tugas Akhir ini. Selain itu, diharapkan dapat berguna untuk pembaca yang tertarik untuk melakukan pengembangan lebih lanjut. Secara garis besar, buku Tugas Akhir terdiri atas beberapa bagian seperti berikut ini.

Bab I Pendahuluan

Bab ini berisi latar belakang masalah, tujuan dan manfaat pembuatan Tugas Akhir, permasalahan, batasan masalah, metodologi yang digunakan, dan sistematika penyusunan Tugas Akhir.

Bab II Dasar Teori

Bab ini membahas beberapa teori penunjang yang berhubungan dengan pokok pembahasan modul BI dengan berbagai fitur pendukung untuk memudahkan pengguna dalam proses mempelajari materi pada modul *Enterprise Resource Planning Retail* yang mendasari pembuatan Tugas Akhir ini. Teori-teori yang berhubungan dengan metode *Balanced Scorecard* yang dapat digunakan sebagai acuan dalam menghitung nilai kinerja perusahaan berdasarkan sasaran strategisnya.

Bab III Analisis dan Perancangan Sistem

Bab ini membahas mengenai perancangan perangkat lunak modul BI dengan fitur-fitur penunjang proses analisis keadaan perusahaan dan penunjang proses pemilihan keputusan. Perancangan perangkat lunak meliputi perancangan data, arsitektur, proses dan perancangan antarmuka aplikasi.

Bab IV Implementasi

Bab ini berisi implementasi dari perancangan modul *business intelligence* dengan penambahan grafik peramalan penjualan menggunakan metode *Fb-Prophet Forecasting*.

Bab V Pengujian dan Evaluasi

Bab ini membahas pengujian dengan metode uji penerimaan pengguna (*User Acceptance Test*) untuk membuktikan bahwa hasil pengujian sudah memenuhi kebutuhan dari pengguna.

Bab VI Kesimpulan dan Saran

Bab ini berisi kesimpulan dari hasil pengujian pada modul *business intelligence*. Bab ini juga membahas saran-saran untuk pengembangan sistem lebih lanjut.

Daftar Pustaka

Merupakan daftar referensi yang digunakan untuk mengembangkan Tugas Akhir.

Lampiran

Merupakan bab tambahan yang berisi daftar istilah yang penting pada aplikasi ini.

BAB II

DASAR TEORI

Pada bab ini akan dibahas mengenai teori-teori yang menjadi dasar dari pembuatan Tugas Akhir.

2.1. ERP (Enterprise Resource Planning)

ERP atau *Enterprise Resources Planning* merupakan sebuah sistem yang mampu mengintegrasikan semua proses bisnis yang ada di dalam suatu perusahaan, sehingga dapat memaksimalkan kinerja perusahaan tersebut. ERP bekerja sebagai kekuatan lintas fungsional perusahaan yang mengintegrasikan dan mengotomatisasi berbagai proses bisnis internal dan sistem informasi termasuk *manufacturing*, logistik, distribusi, akuntansi, keuangan, dan sumber daya manusia dari sebuah perusahaan [2].

Keuntungan penggunaan ERP di antaranya adalah integrasi data keuangan, standarisasi proses operasi, standarisasi data dan informasi, penurunan inventori dan tenaga kerja, peningkatan servis dan kontrol keuangan dan penurunan waktu yang dibutuhkan untuk mendapatkan informasi. Ada pun departemen-departemen atau bagian-bagian yang pada umumnya terintegrasi meliputi:

- a. Pengelolaan keuangan (*financial management*) diwakili oleh modul *Account Payable, Account Receivable, Fixed Asset, Cost Management, General Ledger*.
- b. Pengelolaan rantai pasokan (*supply chain management*) diwakili oleh modul *Purchasing*.
- c. Perencanaan produksi dan manufaktur (*manufacturing resource planning*) diwakili oleh modul *Production dan Inventory*.
- d. Pengelolaan sumber daya manusia (*Human Resource Management*) diwakili oleh modul *Human Resource*.
- e. Pengelolaan relasi dengan pelanggan (*customer relationship management*) diwakili oleh modul *Sales and Distribution*.

2.2. RETAIL

Retail merupakan suatu usaha bisnis yang berusaha memasarkan barang dan jasa kepada konsumen akhir yang menggunakannya untuk keperluan pribadi dan rumah tangga. Produk yang dijual dalam usaha *retail* adalah barang, jasa maupun gabungan dari keduanya [3]. Pada umumnya mendistribusikan barang/jasa secara eceran dengan cara membeli barang dari produsen atau *retailer* lain dalam jumlah banyak untuk kemudian dijual kembali dengan menambah nilai jual kepada konsumen ataupun perusahaan lain. Modul ERP *retail* diantaranya adalah *Business Process Management, Inventory, Sales, Purchase, Business Intelligence, General Ledger dan Accounting*.

2.2.1. Inventory

Modul Persediaan (*Inventory Module*) adalah modul yang mengatur persediaan bahan baku dan dapat memberikan informasi keluar masuknya stok barang dalam perusahaan. Modul ini terdiri dari daftar barang dan jasa (*List of Item*), formulir penyesuaian persediaan (*Inventory Adjustment Form*), formulir pembiayaan pesanan (*Job Costing Form*), daftar gudang (*List of Warehouse*), formulir grup barang (*Item Grouping Form*), formulir penyesuaian harga jual barang (*Set Selling Price Adjustment Form*), dan formulir pindah barang (*Item Transfer Form*) [4]. Alasan perlunya manajemen persediaan adalah karena timbulnya ketidakpastian permintaan, ketidakpastian pasokan *supplier*, dan ketidakpastian waktu pemesanan.

Tujuannya adalah untuk memberikan pelayanan yang terbaik bagi konsumen, memperlancar proses produksi, mengantisipasi kekurangan persediaan (*stock out*), dan dalam rangka menghadapi fluktuasi harga. Beberapa pendekatan yang digunakan adalah *economic order quantity, periodic review, dan material requirement planning*.

2.2.2. Sales

Modul Penjualan (*Sales Module*) adalah modul yang mengoperasikan proses penjualan, mencatat penjualan dan mengolah transaksi penjualan yang ada di perusahaan. Modul ini terdiri dari Formulir Penawaran Penjualan (*Sales Quotation Form*), Formulir Pesanan Penjualan (*Sales Order Form*), Formulir Pengiriman Pesanan (*Delivery Order Form*), Formulir Faktur Penjualan (*Sales Invoice Form*), Formulir Retur Penjualan (*Sales Return Form*), Formulir Penerimaan Penjualan (*Sales Receipt Form*) [4].

Tujuan utama dari modul *sales* adalah menangani semua proses yang berhubungan dengan pembelian secara umum. Adapun fungsi-fungsi secara terperinci antara lain menangani pembelian, *pricelist product*, konversi otomatis untuk pembelian dengan mata uang tertentu (*multi currency*), biaya pengiriman, retur penjualan dan lain-lain.

2.2.3. Purchase

Modul Pembelian (*Purchase Module*) adalah modul yang mengoperasikan proses pembelian, mencatat pembelian dan mengolah transaksi pembelian yang ada di perusahaan. Modul ini terdiri dari formulir permintaan pembelian (*Purchase Requisition Form*), formulir pesanan pembelian (*Purchase Order Form*), formulir penerimaan barang (*Received Item Form*), formulir faktur pembelian (*Purchase Invoice Form*), formulir retur pembelian (*Purchase Return Form*), dan formulir pembayaran pembelian (*Purchase Payment Form*) [4].

Tujuan dari modul *purchase* adalah mencatat transaksi pembelian dengan berbagai kondisi, mulai dari pemesanan, pembelian tunai dan kredit, pembayaran hutang, pertukaran barang yang telah dijual.

2.2.4. General Ledger dan Accounting

Modul Buku Besar (*General Ledger Module*) adalah modul yang mencatat jurnal ke buku besar sebagai dasar pembuatan laporan neraca dan laporan laba rugi. Modul ini terdiri dari formulir daftar akun (*List of Account*), daftar mata uang (*List of Currency*), informasi perusahaan (*Company Info*), formulir bukti jurnal (*Journal Voucher Form*), proses akhir bulan (*Period End*), laporan keuangan (*Financial Statemen*) [4].

Tujuannya adalah untuk mencatat segala transaksi jurnal, buku besar, hingga menghasilkan laporan keuangan seperti neraca, laba-rugi, dan arus kas secara otomatis.

2.2.5. Business Intelligence

Business Intelligence atau biasa disebut BI merupakan sistem yang mengumpulkan, memproses dan mempresentasikan data yang terstruktur dari sumber yang banyak untuk mempersingkat waktu yang digunakan dalam mendapatkan informasi bisnis dan mengefisienkan proses pengambilan keputusan[5].

Tujuan dari *Business Intelligence* yaitu untuk memudahkan interpretasi dari jumlah data yang besar tersebut. Memungkinkan pencarian data perusahaan yang dinamis, pengambilan data, analisis data, dan penjelasan tentang kebutuhan keputusan manajerial [6].

2.3. SUPPLY CHAIN MANAGEMENT

Supply Chain Management merupakan pengelolaan jaringan bisnis yang saling berhubungan yang terlibat dalam penyediaan produk yang dibutuhkan oleh pelanggan. SCM mencakup semua perpindahan barang termasuk barang mentah, barang setengah jadi ataupun barang jadi dari titik asal sampai titik akhir konsumen [7].

2.4. *BALANCED SCORECARD*

Balanced Scorecard merupakan konsep yang digunakan untuk mengukur kemajuan perusahaan atau organisasi berdasarkan sasaran-sasaran strategisnya. *Balance Scorecard* terdiri dari empat perspektif yang ada, yaitu perspektif finansial, perspektif *costumer*, perspektif proses bisnis internal dan perspektif pertumbuhan dan pembelajaran [8]. *Balanced Scorecard* juga berfungsi untuk menjabarkan misi dan strategi dari sebuah perusahaan ke dalam tujuan-tujuan operasional dan tolok ukur kinerja perusahaan tersebut.

2.5. *DECISION SUPPORT SYSTEM*

Decision Support System merupakan sistem digunakan untuk mendukung dalam pengambil keputusan yang bersifat manajerial dalam situasi keputusan yang semi terstruktur. DSS dimaksudkan untuk menjadi tambahan pertimbangan untuk pengambil keputusan untuk memperluas kemampuan mereka tapi tidak untuk menggantikan penilaian mereka [9].

[Halaman ini sengaja dikosongkan]

BAB III

ANALISIS DAN PERANCANGAN SISTEM

Bab ini membahas tahap analisis dan perancangan sistem yang akan dibangun. Analisis membahas semua persiapan yang akan menjadi pokok pikiran pembuatan aplikasi ini. Mulai dari masalah yang melatarbelakangi, hingga analisis gambaran awal sistem yang akan dibuat. Perancangan sistem membahas hal-hal yang berkaitan dengan pondasi atau dasar pembuatan aplikasi, yang meliputi perancangan basis data, hingga tampilan antar muka halaman aplikasi.

3.1. Analisis

Tahap analisis dibagi menjadi beberapa bagian antara lain: analisis permasalahan, dan analisis kebutuhan. Pembahasan bagian-bagian tahap analisis akan dijabarkan sebagai berikut:

3.1.1. Analisis Permasalahan

Pengguna aplikasi ERP semakin hari semakin bertambah, hal tersebut dikarenakan penggunaan ERP dapat memudahkan pencatatan transaksi-transaksi bahkan sampai menampilkan laporan harian. Aplikasi ERP juga sudah mulai dikenal tidak hanya oleh kalangan pebisnis besar, namun juga sudah merambah ke kalangan usaha kecil dan menengah. Akan tetapi penggunaan ERP yang dilengkapi dengan banyak modul ini juga harus memiliki fungsi lain selain mempermudah pencatatan. Aplikasi ERP juga harus dapat membantu pemilik perusahaan untuk mengembangkan usahanya. Namun modul tersebut belum tersedia di aplikasi ERP retail yang telah dikembangkan.

Untuk melengkapi fungsi ERP retail, perlu dibuatnya modul baru yang dapat memberikan saran langkah bisnis dan memberikan informasi yang dibutuhkan oleh pemilik perusahaan untuk mengembangkan usahanya, yaitu modul *Business Intelligence*. Melalui modul ini, system memberikan

rekomendasi-rekomendasi dengan metode perhitungan tertentu sehingga menghasilkan keputusan paling optimal. Namun pengambilan keputusan tetap berada di tangan pemilik perusahaan.

3.1.2. Analisis Kebutuhan

Modul *Buisness Intelligence* ini difokuskan untuk memberikan informasi kepada pengguna tentang keadaan perusahaannya, khususnya dibidang *stock*, keuangan, dan *forecast* penjualan. Namun modul ini juga memberikan rekomendasi jumlah *re-stock* paling optimal dan waktu *re-stock* paling efektif. Tabel 3.1 merupakan daftar kebutuhan fungsional dari modul *Business Intelligence* yang akan dibangun.

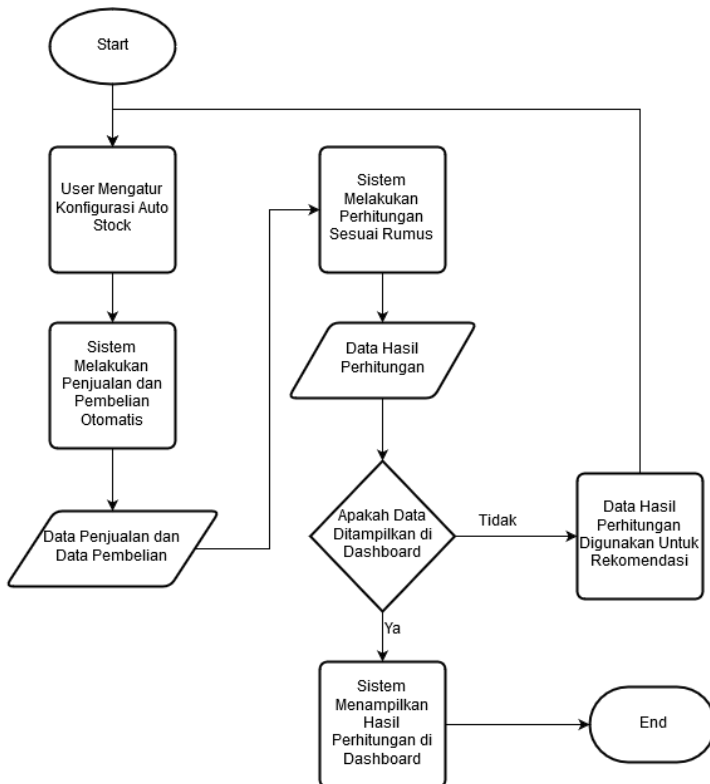
Tabel 3.1 Daftar Kebutuhan Fungsional Sistem

Kode Kebutuhan	Kebutuhan Fungsional	Deskripsi
F-1	Memberikan pilihan <i>supplier</i> yang optimal	Pengguna diberikan rekomendasi pilihan <i>supplier</i> yang paling optimal.
F-2	Memberikan jumlah <i>re-stock</i> yang optimal dan waktu <i>re-stock</i> yang efektif	Pengguna diberikan rekomendasi jumlah dan waktu <i>re-stock</i> yang paling optimal dan efektif
F-3	Memberikan rangkuman informasi dari laporan keuangan yang ada	Pengguna disajikan grafik <i>profit margin</i> , <i>total asset turnover</i> , dan <i>return of asset</i>

F-4	Memberikan informasi tentang prediksi penjualan	Pengguna disajikan grafik <i>forecasting</i> penjualan.
-----	---	---

3.2. Deskripsi Umum Sistem

Aplikasi yang akan dibuat pada Tugas Akhir ini adalah program aplikasi berbasis web. Gambar 3.1 adalah gambaran alur jalan sistem.



Gambar 3.1 Deskripsi Umum Sistem

Keterangan pada Gambar 3.1:

1. Pengguna memasukkan konfigurasi produk.
2. Data konfigurasi digunakan untuk membuat pembelian dan penjualan secara otomatis.
3. Setelah data penjualan dan pembelian tercatat, sistem menggunakannya untuk melakukan perhitungan sesuai dengan rumus.
4. Data hasil perhitungan akan ditampilkan ke halaman *dashboard* dan lainnya akan digunakan untuk rekomendasi kepada pengguna dalam mengisi konfigurasi produk.

Untuk penjelasan secara rinci akan dijelaskan di bawah mulai dari *input*, proses hingga *output*..

3.2.1. Input

Modul *Business Intelligence* menggunakan data-data yang lalu untuk memprediksi dan merekomendasikan langkah bisnis yang akan datang. Sehingga input untuk modul ini didapatkan dari data-data yang telah diinputkan oleh pengguna selama pengoperasian aplikasi ERP Retail. Namun data yang digunakan oleh modul BI ini hanya data transaksional saja, khususnya data keuangan, data pembelian, data penjualan, dan data penerimaan barang.

3.2.2. Proses

Pada tahap ini dijelaskan secara bertahap langkah-langkah yang dilakukan pada sistem diantaranya proses *supplier selection*, proses analisis keuangan, perhitungan EOQ dan ROP, proses *forecasting* penjualan, dan perhitungan harga optimal.

3.2.2.1. Supplier Selection

Proses pemilihan supplier dilakukan dengan cara pembobotan pada indikator yang digunakan. Pada proses ini digunakan 5 indikator yang akan digunakan. Indikator tersebut adalah:

- *Quality*

Indikator ini menilai *supplier* dari segi kualitas terhadap produk yang dihasilkan oleh *supplier*. Pada perusahaan *retail* yang tidak membuat sendiri produknya, maka kualitas produk dari *supplier* harus menjadi pengawasan yang utama. Indikator *quality* ini dapat dihitung melalui persentase barang rusak dari pembelian barang.
- *Cost*

Indikator ini menilai *supplier* dari biaya yang ditetapkan oleh *supplier*. Selain itu, diskon yang ditawarkan pada pembelian dengan jumlah tertentu juga ikut diperhitungkan sehingga perusahaan bisa mendapatkan barang dengan harga terbaik. Adanya kebebasan untuk membayar dengan sistem cicilan juga berpengaruh terhadap penilaian *supplier* pada indikator ini.
- *Delivery*

Indikator ini menilai *supplier* dari segi pelayanan pengiriman produk. Ketepatan waktu pengantaran barang merupakan faktor penting dalam memilih *supplier*.
- *Responsiveness*

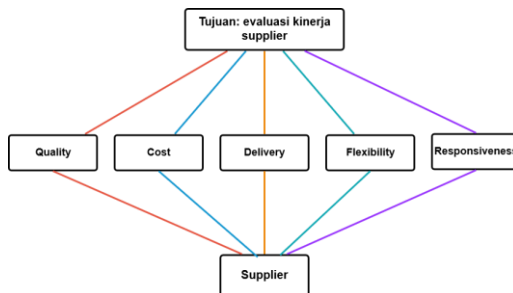
Indikator ini menilai *supplier* dari segi kemampuan dalam menanggapi masalah yang ada. Masalah ini diartikan sebagai keadaan saat terdapat barang yang rusak dalam pengirimannya. Kemampuan *supplier* dalam menanggapi pengembalian barang rusak inilah yang menjadi penilaian *supplier* tersebut.

- *Flexibility*

Indikator ini menilai *supplier* dari segi kecepatan pengiriman kembali barang pengganti dari barang yang rusak.

Bobot penilaian pada kelima indikator tersebut ditentukan berdasarkan keinginan pengguna. Indikator tersebut akan disusun berdasarkan prioritas indikator yang dibutuhkan. Maka dari itu, bobot penilaian yang digunakan berdasarkan prioritas indikatornya, sehingga indikator yang paling utama yang mendapatkan bobot paling tinggi. Nilai bobot tersebut dianalisis menggunakan metode AHP (*Analytic Hierarchy Process*) yang diimplementasikan dengan *software Expert Choice 11*. Tahapan menggunakan AHP dan EC disini dibagi menjadi 4 tahap, yaitu menyusun hirarki permasalahan, membuat matriks perbandingan berpasangan, melakukan sintesis, dan evaluasi konsistensi.

Pada tahap menyusun hirarki permasalahan, masalah yang digunakan dibagi sesuai dengan struktur model AHP sehingga menjadi kluster-kluster sederhana yang mewakili tingkat yang berbeda pada struktur hirarkikal. Pembagian dilakukan dari atas ke bawah, mulai dari tujuan, ke kriteria, ke alternatif akhir. Pada kasus ini, struktur model dibagi menjadi tujuan (evaluasi kinerja *supplier*), kriteria (*quality, cost, delivery, flexibility, responsiveness*) dan alternatif (*supplier*). Struktur model AHP kasus ini ditunjukkan pada Gambar 3.2.



Gambar 3.2 Struktur model AHP evaluasi kinerja *supplier*

Setelah model hirarki permasalahan dibuat, tahap selanjutnya merupakan pembuatan matriks perbandingan berpasangan (*matrix of pairwise comperisone*). Pada tahap perbandingan berpasangan, tiap kluster diberikan nilai untuk mengukur kepentingan tiap tingkat dalam hirarki. Perbandingan menggunakan nilai dari ‘*fundamental scale of Saaty*’ dengan 9 titik skala. Titik skala tersebut ditunjukkan pada Tabel 3.2 dan Tabel 3.3. Pada tiap tingkat hirarki, dilakukan pertimbangan numerik sehingga menghasilkan matrik berukuran $n \times n$ dimana n merupakan sejumlah kriteria dalam tingkat hirarki. Semua matrik berpasangan yang dihasilkan memiliki 2 sifat dasar, yaitu diagonal utama selalu bernilai 1 karena tiap kriteria dibandingkan dengan dirinya sendiri dan matrik kebalikan ($i_{ajb} = 1/(i_{bja})$). Sedangkan matriks perbandingan berpasangan untuk kasus pemilihan *supplier* ini ditunjukkan pada Tabel 3.4

Tabel 3.2 Skala Fundamental Saaty

Nilai (i_{ajb})	Penjelasan
1	Sama penting satu sama lain
3	Agak penting dibanding yang lain
5	Lebih penting dibanding yang lain
7	Sangat penting dibanding yang lain
9	Mutlak penting dibanding yang lain
2,4,6,8	Nilai diantara dua penilaian yang berdekatan

$a, b = (1, 2, 3, \dots, n = \text{jumlah kriteria})$

Tabel 3.3 Perbandingan Pasangan Kriteria

i \ j	Kriteria						
	1	2	3	4	...	n	
Kriteria	1	1	i_{1j_2}	i_{1j_3}	i_{1j_4}	...	i_{1j_n}
	2	i_{2j_1}	1	i_{2j_3}	i_{2j_4}	...	i_{2j_n}
	3	i_{3j_1}	i_{3j_2}	1	i_{3j_4}	...	i_{3j_n}
	4	i_{4j_1}	i_{4j_2}	i_{4j_3}	1	...	i_{4j_n}

	n	i_{nj_1}	i_{nj_2}	i_{nj_3}	i_{nj_4}	...	1

Tabel 3.4 Tabel Matriks Perbandingan Berpasangan Memilih Supplier

<i>Indikator</i>	Q	C	D	F	R
<i>Quality (Q)</i>	1	2	3	4	5
<i>Cost (C)</i>	1/2	1	2	3	4
<i>Delivery (D)</i>	1/3	1/2	1	2	3
<i>Flexibility (F)</i>	1/4	1/3	1/2	1	2
<i>Responsiveness (R)</i>	1/5	1/4	1/3	1/2	1

Pada Tabel 3.4, nilai tersebut menunjukkan bahwa indikator yang satu dianggap sedikit lebih penting daripada indikator lainnya, sehingga diberi nilai dengan selisih 1 angka. Sel-sel di bawah diagonal diisi dengan memberikan skala secara kebalikan dari sel-sel bagian atas diagonal.

Tahap berikutnya setelah itu ialah mensintesis perbandingan. Sintesis bertujuan untuk mendapatkan prioritas dari seluruh kriteria berdasarkan matriks perbandingan. Sintesis dilakukan dengan melakukan normalisasi pada matriks perbandingan. Tabel 3.5 menunjukkan hasil dari matriks perbandingan yang telah dinormalisasi.

Tabel 3.5 Matriks Perbandingan yang Dinormalisasi

<i>Indikator</i>	Q	C	D	F	R	Rata-rata
<i>Quality (Q)</i>	0.438	0.490	0.439	0.381	0.333	0.416
<i>Cost (C)</i>	0.219	0.245	0.293	0.286	0.267	0.262
<i>Delivery (D)</i>	0.146	0.122	0.146	0.190	0.200	0.161
<i>Flexibility (F)</i>	0.109	0.082	0.073	0.095	0.133	0.099
<i>Responsiveness (R)</i>	0.088	0.061	0.049	0.048	0.067	0.062
<i>Jumlah</i>	1.000	1.000	1.000	1.000	1.000	1.000

Nilai rata-rata pada setiap baris menunjukkan nilai prioritas relatif indikator (baris) tersebut terhadap indikator lainnya. Nilai itu juga yang nantinya akan digunakan sebagai bobot perhitungan indikator untuk memilih supplier.

Langkah selanjutnya adalah mengukur konsistensi dari perhitungan yang telah dilakukan. Pengukuran ini dilakukan dengan cara mengalikan nilai prioritas dengan setiap entri pada matriks perbandingan. Kemudian hasil perkalian tersebut dijumlahkan perbaris dan dihitung rata-ratanya. Nilai rata-rata itulah yang disebut dengan vektor konsistensi (λ). Tabel 3.6 dan Tabel 3.7 menunjukkan proses dalam penghitungan nilai vektor konsistensi.

Tabel 3.6 Matriks Perbandingan yang Dibobot

	Q	C	D	F	R	Jumlah
Q	1 x 0.416	2 x 0.262	3 x 0.161	4 x 0.099	5 x 0.062	2.129
C	1/2 x 0.416	1 x 0.262	2 x 0.161	3 x 0.099	4 x 0.062	1.337
D	1/3 x 0.416	1/2 x 0.262	1 x 0.161	2 x 0.099	3 x 0.062	0.815
F	1/4 x 0.416	1/3 x 0.262	1/2 x 0.161	1 x 0.099	2 x 0.062	0.495
R	1/5 x 0.416	1/4 x 0.262	1/3 x 0.161	1/2 x 0.099	1 x 0.062	0.314

Tabel 3.7 Nilai Hasil Vektor Konsistensi

Indikator	Jumlah/vektor prioritas	Vektor konsistensi
Quality (Q)	2.129 / 0.416	5.115
Cost (C)	1.337 / 0.262	5.108
Delivery (D)	0.815 / 0.161	5.060
Flexibility (F)	0.495 / 0.099	5.023
Responsiveness (R)	0.314 / 0.062	5.034
	Rata-rata (λ)	5.068

Selanjutnya adalah penghitungan rasio konsistensi (CR) menggunakan persamaan yang ditunjukkan pada Persamaan (3.1) dan Persamaan (3.2).

$$CR = CI/RI \quad (3.1)$$

$$CI = \frac{\lambda_{max} - n}{(n-1)} \quad (3.2)$$

dimana,

CI = consistency index

RI = random index

RI merupakan rata-rata indeks konsistensi yang dihasilkan oleh Saaty menggunakan matriks yang dihasilkan secara acak hingga ukuran 11 x 11 untuk 500 sampel. Tabel 3.8 merupakan nilai RI yang telah standar.

Tabel 3.8 Nilai Indeks Acak (*random index*)

n	3	4	5	6	7	8	9	10
RI	0,58	0,90	1,12	1,24	1,32	1,41	1,45	1,49

Jika nilai CR dari suatu matrik tinggi, berarti penilaian menjadi tidak konsisten atau tidak dapat diandalkan. Umumnya, nilai CR yang diperbolehkan kurang dari 0,10. Jika nilai CR melebihi 0,10 maka penilaian tidak dapat digunakan dan harus dilakukan penilaian ulang.

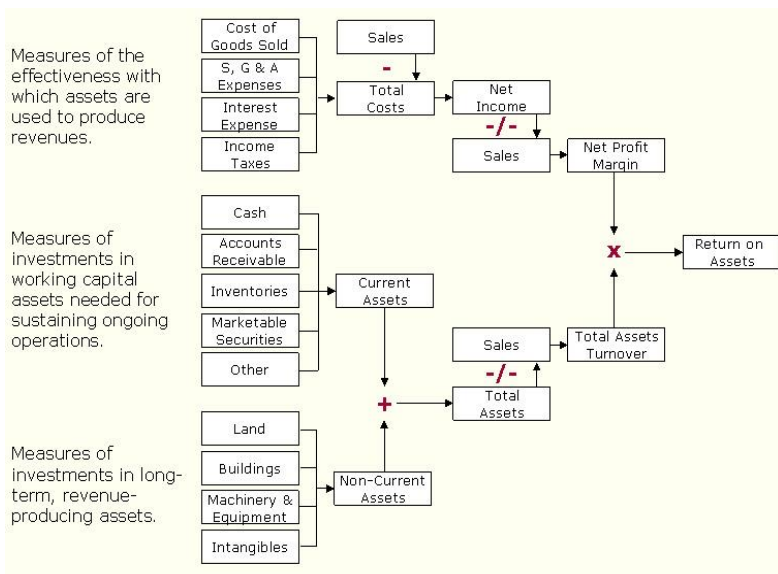
Dengan menggunakan Persamaan (3.1) dan Persamaan (3.2), kita dapat mencari nilai CI dengan $n = 5$ dan $\lambda = 5.068$ yang didapat dari perhitungan sebelumnya. Sehingga diperoleh $CI = 0.017$. Kemudian dengan menggunakan Persamaan (3.2) dan nilai $RI = 1.12$, didapatkan besar nilai $CR = 0.017/1.12 = 0.015$. Karena hasil perhitungan menghasilkan $CR < 0.10$, berarti respon cukup konsisten dan tidak perlu melakukan reevaluasi terhadap matriks perbandingan yang telah dibuat. Sehingga nilai relatif indikator tersebut dapat digunakan sebagai bobot untuk memilih supplier. Tabel 3.9 merupakan rangkuman akhir nilai bobot indikator yang akan digunakan.

Tabel 3.9 Nilai Bobot Indikator

Nilai Bobot Indikator	Bobot
Quality	0,416
Cost	0,262
Delivery	0,161
Flexibility	0,099
Responsiveness	0,062

3.2.2.2. Proses Analisis Keuangan

Proses ini bertujuan untuk mengevaluasi kinerja perusahaan berdasarkan aspek keuangannya. Metode analisis keuangan yang digunakan adalah metode *DuPont Analysis*. Cara analisis menggunakan *DuPont* hampir sama dengan analisis laporan keuangan biasa, tetapi pendekatannya lebih integratif dan menggunakan komposisi laporan keuangan sebagai elemen analisisnya. Gambar 3.2 merupakan *framework* dari *DuPont Analysis*.



Gambar 3.3 Dupont Analysis Framework

Melalui metode ini, pengguna dapat mengetahui seberapa persen keuntungan yang didapatkan dari penjualan. Di samping itu, pengguna juga dapat melihat seberapa efektif penjualan kita terhadap aset perusahaan, serta melihat seberapa efektif keuntungan terhadap aset perusahaan tersebut.

3.2.2.3. Perhitungan EOQ dan ROP

EOQ (*Economical Order Quantity*) merupakan metode yang digunakan untuk menentukan jumlah pemesanan barang yang paling efektif. Sehingga dapat menghemat biaya dan waktu untuk pemesanan barang. Perhitungan EOQ tersebut menggunakan persamaan yang ditunjukkan pada Persamaan (3.3).

$$EOQ = \sqrt{\frac{2DC}{h}} \quad (3.3)$$

dimana,

EOQ = *Economical Order Quantity*

D = penjualan per periode

C = biaya pemesanan setiap kali pesan (bukan per unit)

h = biaya penyimpanan tiap unit barang yang disimpan (dalam rupiah)

Periode yang digunakan untuk EOQ yaitu periode mingguan. Sehingga penjualan yang digunakan adalah penjualan per produk dalam 7 hari. Sedangkan biaya setiap kali pesan merupakan ongkos kirim yang dikenakan saat memesan barang. Ongkos kirim ini mempunyai nilai yang berbeda untuk masing-masing supplier. Untuk biaya penyimpanan tiap unit barang dihitung per satuan hari.

Sedangkan perhitungan ROP digunakan untuk menentukan banyak stok tersisa yang efektif untuk melakukan pemesanan stok kembali. Nilai ROP dapat dihitung dengan menggunakan persamaan yang ditunjukkan pada Persamaan (3.4) dan Persamaan (3.5).

$$ROP = D \times LT + SS, \text{ dimana} \quad (3.4)$$

$$SS = (\sqrt{D} \times \sqrt{LT}) - (\bar{D} \times \bar{LT}) \quad (3.5)$$

dimana,

ROP = *Re Order Point*

D = penjualan per periode

LT = tenggang waktu antara pemesanan dan penerimaan barang

SS = (*safety stock*) persediaan barang yang tergolong aman

Periode yang digunakan untuk perhitungan ROP juga periode mingguan atau per 7 hari. Besarnya tenggang waktu antara pemesanan dan penerimaan barang didapatkan melalui rata-rata perbedaan waktu dari 7 data penerimaan barang terakhir barang tersebut. Sedangkan *safety stock* dapat dihitung melalui maksimal penjualan dikali maksimal tenggang waktu dikurangi rata-rata penjualan dikali tenggang waktu. Data penjualan dan data tenggang waktu juga menggunakan data yang sama yang digunakan untuk perhitungan ROP.

3.2.2.4. Proses *Forecasting* Penjualan

Pada proses *forecasting* penjualan, metode yang digunakan adalah metode *Holt-Winter Multiplicative*. Metode ini dapat diterapkan pada data *time-series* musiman. Pada model ini diasumsikan bahwa komponen-komponen data *time-series* (pemulusan data, *trend*, dan musiman), dikalikan satu sama lain sehingga menghasilkan data *time-series* yang lebih aktif. Persamaan metode *Holt-Winters Multiplicative* ditunjukkan pada Persamaan (3.6), Persamaan (3.7), Persamaan (3.8) dan Persamaan (3.9).

$$F_{t+h_t} = (L_t + hB_t)S_{t-m} \quad (3.6)$$

dimana,

F_{t+h_t} = Hasil peramalan

L_t = nilai tingkat pemulusan pada periode t

h = periode

B_t = nilai *trend* pada periode t

S_{t-m} = nilai musiman pada periode sebelumnya yang sama

$$L_t = \alpha \left(\frac{Y_t}{S_{t-m}} \right) + (1 - \alpha)(L_{t-1} + B_{t-1}) \quad (3.7)$$

dimana,

L_t = Tingkat pemulusan

α = konstanta pemulusan yang berada antara 0 dan 1

Y_t = data aktual pada periode t

L_{t-1} = nilai tingkat pemulusan pada periode t-1

B_{t-1} = nilai trend pada periode t-1

$$B_t = \beta(L_t - L_{t-1}) + (1 - \beta)B_{t-1} \quad (3.8)$$

dimana,

B_t = Nilai *Trend*

β = konstanta pemulusan yang berada antara 0 dan 1

L_t = nilai tingkat pemulusan pada periode t

L_{t-1} = nilai tingkat pemulusan pada periode t-1

B_{t-1} = nilai trend pada periode t-1

$$S_t = \gamma \left(\frac{Y_t}{L_{t-1} - B_{t-1}} \right) + (1 - \gamma)S_{t-m} \quad (3.9)$$

dimana,

B_t = Nilai Musiman

γ = konstanta pemulusan yang berada antara 0 dan 1

Y_t = data aktual pada periode t

S_{t-m} = nilai musiman pada periode sebelumnya yang sama

L_{t-1} = nilai tingkat pemulusan pada periode t-1

B_{t-1} = nilai trend pada periode t-1

3.2.2.5. Perhitungan Harga Optimal

Harga yang optimal dapat dihitung dengan menggunakan *Price Elasticity of Demand* (PED) dan *Marginal Cost*. PED merupakan perhitungan yang digunakan dalam ekonomi untuk menunjukkan respons, atau elastisitas dari kuantitas yang diminta dari suatu barang atau jasa terhadap suatu perubahan harga. Lebih tepatnya, ia memberikan perubahan persentase dalam jumlah yang diminta sebagai respons terhadap perubahan harga satu persen. Sedangkan *Marginal Cost* merupakan biaya untuk menjual satu unit tambahan dari produk.

Persamaan perhitungan harga optimal menggunakan PED dan *Marginal Cost* ditunjukkan pada Persamaan (3.10), Persamaan (3.11) dan Persamaan (3.12)

$$OP = MC * (PED / (PED + 1)) \quad (3.10)$$

dimana,

OP = Harga optimal

MC = *Marginal Cost*

PED = *Price Elasticity of Demand*

$$MC = \frac{Ep + (COGS \times D)}{D} \quad (3.11)$$

dimana,

MC = *Marginal Cost*

Ep = biaya pengeluaran operasional

$COGS$ = harga beli barang dagang

D = jumlah barang yang terjual

$$PED = \left[\frac{(Q^1 - Q^0)}{(Q^1 + Q^0)} \right] \times \left[\frac{(P^1 + P^0)}{(P^1 - P^0)} \right] \quad (3.12)$$

dimana,

PED = *Price Elasticity of Demand*

P^0 = harga awal

P^1 = harga akhir

Q^0 = jumlah permintaan pada harga awal

Q^1 = jumlah permintaan pada harga akhir

3.2.2.6. Output

Setelah melalui masing-masing proses, hasil perhitungan akan dimasukkan ke dalam database. Sehingga aplikasi ERP Retail hanya perlu mengambil data dari database dan menampilkannya. Cara penyajian data berbeda untuk masing-masing proses. Untuk proses analisis keuangan, data hasil perhitungan disajikan dalam bentuk grafik *profit margin*, *total asset turnover*, dan *return of asset* yang disediakan pada halaman dashboard. Sedangkan untuk hasil perhitungan *supplier selection*, *EOQ*, *ROP* dan harga optimal disajikan melalui tombol rekomendasi pada halaman *Config*

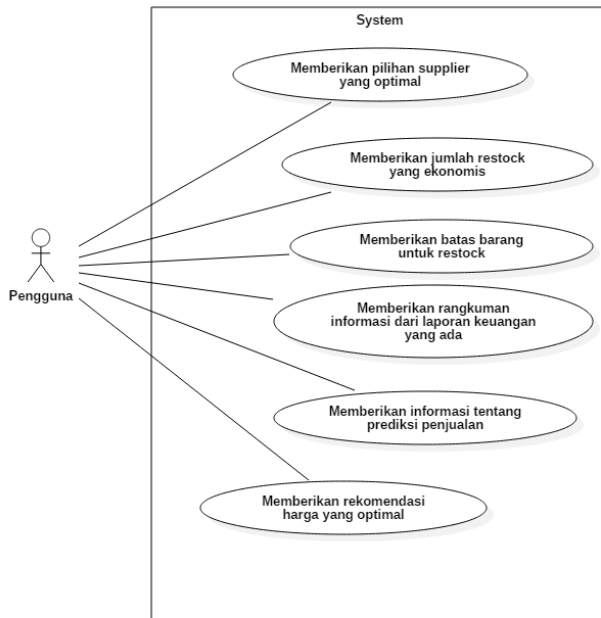
AutoStock. Sehingga pengguna hanya perlu menekan tombol rekomendasi dan kolom isian harga, *supplier*, batas pemesanan dan jumlah pesan menjadi otomatis terisi.

3.2.3. Kasus Penggunaan

Mengacu pada spesifikasi kebutuhan fungsional yang telah dipaparkan, dibuat kasus penggunaan yang selanjutnya akan disimpulkan dalam deskripsi umum sistem, yang diharapkan dapat memenuhi kebutuhan fungsional, berdasar pada kasus penggunaan yang dibuat. Kasus penggunaan dijelaskan lebih lanjut pada Tabel 3.10 dan diagram kasus penggunaan ditunjukkan pada Gambar 3.4.

Tabel 3.10 Daftar Kode Kasus Penggunaan

Kode Kasus Penggunaan	Nama	Aktor
UC-1	Memberikan pilihan <i>supplier</i> yang optimal	Pengguna
UC-2	Memberikan jumlah <i>re-stock</i> yang ekonomis	Pengguna
UC-3	Memberikan batas barang untuk <i>re-stock</i>	Pengguna
UC-4	Memberikan rangkuman informasi dari laporan keuangan yang ada	Pengguna
UC-5	Memberikan informasi tentang prediksi penjualan	Pengguna
UC-6	Memberikan rekomendasi harga barang	Pengguna



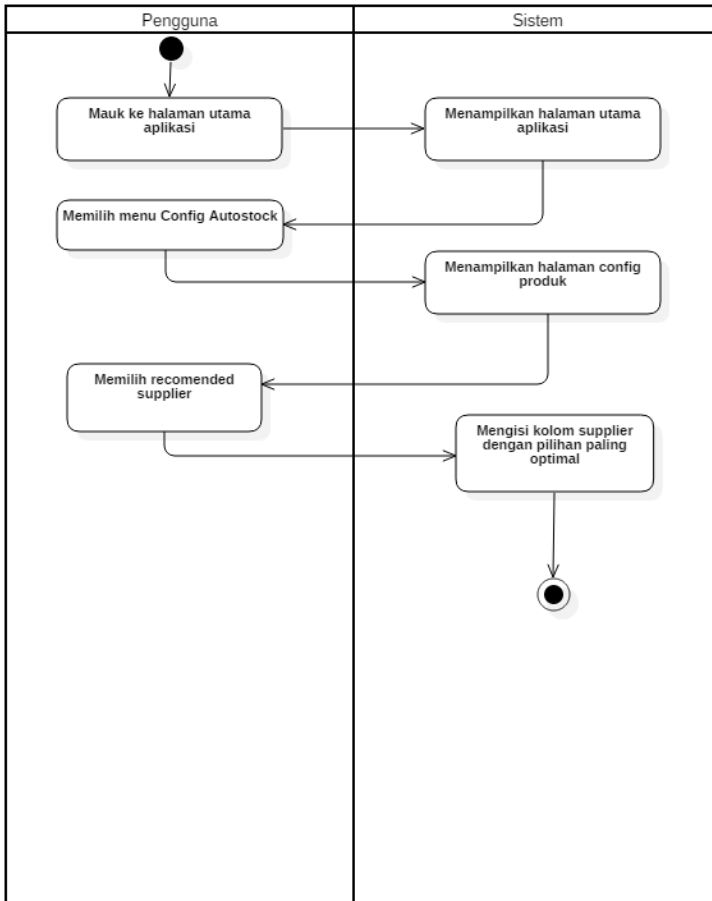
Gambar 3.4 Diagram Kasus Penggunaan

3.2.3.1. Memberikan pilihan *supplier* yang optimal (UC-1)

Kasus penggunaan kode UC-1 diakses pengguna setelah berhasil mengakes halaman pemesanan. Pada tahap ini pengguna akan diberikan pilihan yang dapat memilih secara otomatis supplier mana yang paling optimal. Spesifikasi penggunaan dan diagram aktivitas dapat dilihat pada Tabel 3.11 dan Gambar 3.5.

Tabel 3.11 Spesifikasi Kasus Penggunaan Memberikan Pilihan Supplier Optimal

Nama	Memberikan pilihan <i>supplier</i> yang optimal
Nomor	UC-1
Description	Kasus penggunaan ini digunakan untuk melihat <i>supplier</i> mana yang paling optimal
Tipe	Fungsional
Aktor	Pengguna <i>retail</i>
Kondisi Awal	Pengguna sudah masuk ke sistem
Kondisi Akhir	Sistem mengisi kolom <i>supplier</i> dengan pilihan <i>recommended supplier</i>
Alur Normal	<ol style="list-style-type: none"> 1. Pengguna membuka halaman utama aplikasi 2. Sistem menampilkan halaman utama aplikasi 3. Pengguna memilih menu config autostock 4. Sistem menampilkan halaman config produk 5. Pengguna memilih <i>recommended supplier</i> 6. Sistem mengisi kolom <i>supplier</i> dengan pilihan yang paling optimal 7. Kasus penggunaan berakhir.
Alur Alternatif	-



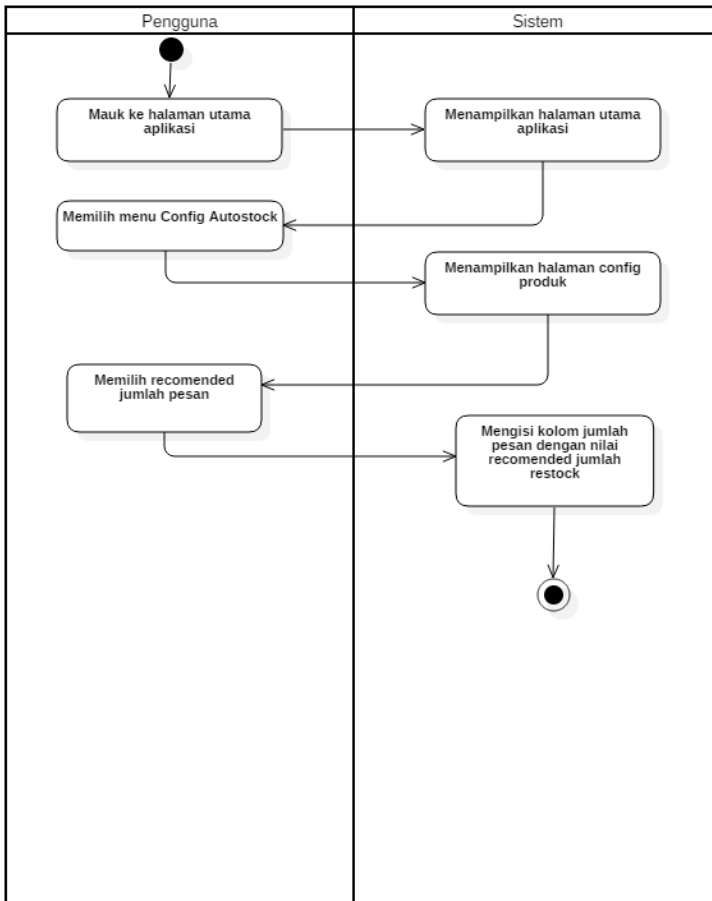
Gambar 3.5 Diagram aktivitas use case UC-1

3.2.3.2. Memberikan jumlah *re-stock* yang ekonomis (UC-2)

Kasus penggunaan kode UC-2 diakses pengguna setelah berhasil mengakes halaman pemesanan. Pada tahap ini pengguna akan diberikan nilai *default* jumlah pesan optimal di kolom jumlah pesan. Namun jumlah pesanan ini bisa diubah secara manual oleh pengguna jika diinginkan. Spesifikasi penggunaan dan diagram aktivitas dapat dilihat pada Tabel 3.12 dan Gambar 3.6.

Tabel 3.12 Spesifikasi Kasus Penggunaan Memberikan Jumlah Re-Stock yang Ekonomis

Nama	Memberikan jumlah <i>re-stock</i> yang ekonomis
Nomor	UC-2
Description	Kasus penggunaan ini digunakan untuk mengetahui berapa jumlah <i>re-stock</i> yang ekonomis
Tipe	Fungsional
Aktor	Pengguna <i>retail</i>
Kondisi Awal	Pengguna sudah masuk ke sistem
Kondisi Akhir	Sistem mengisi kolom jumlah <i>re-stock</i> dengan <i>recomended jumlah re-stock</i>
Alur Normal	<ol style="list-style-type: none"> 1. Pengguna membuka halaman utama aplikasi 2. Sistem menampilkan halaman utama aplikasi 3. Pengguna memilih menu config autostock 4. Sistem menampilkan halaman config produk 5. Pengguna memilih <i>recomended jumlah restok</i> 6. Sistem mengisi kolom jumlah <i>re-stock</i> secara otomatis 7. Kasus penggunaan berakhir.
Alur Alternatif	-



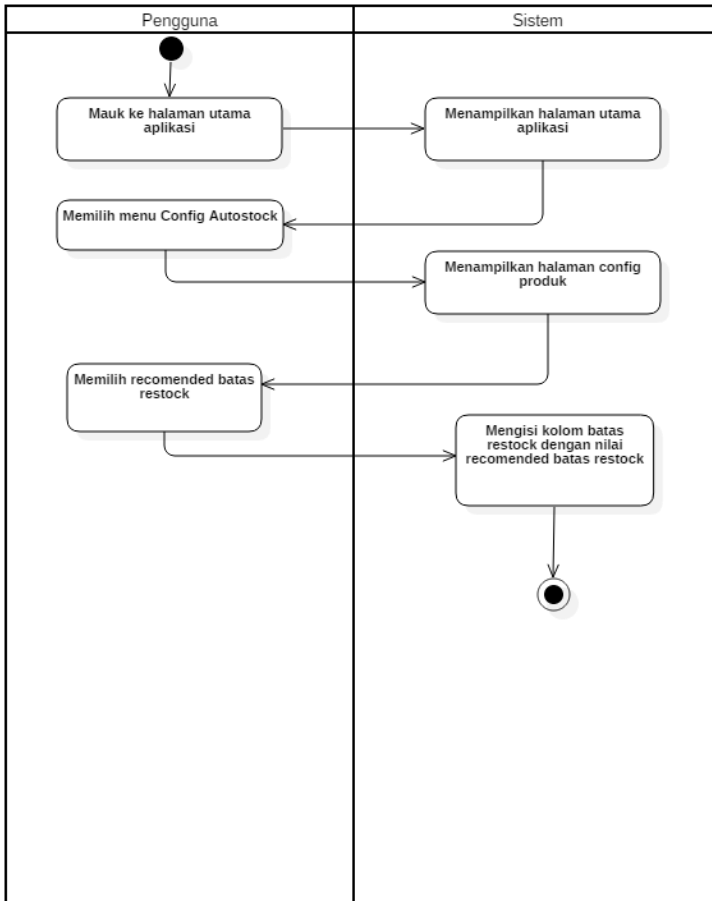
Gambar 3.6 Diagram aktivitas use case UC-2

3.2.3.3. Memberikan batas barang untuk *re-stock* (UC-3)

Kasus penggunaan kode UC-3 diakses pengguna setelah berhasil mengakses halaman stok barang. Pada tahap ini pengguna akan diberikan status ketersediaan barang, apakah itu aman, perlu di *re-stock* atau bahaya. Spesifikasi penggunaan dan diagram aktivitas dapat dilihat pada Tabel 3.13 dan Gambar 3.7.

Tabel 3.13 Spesifikasi Kasus Penggunaan Memberikan Status Ketersediaan Barang

Nama	Memberikan status Ketersediaan Barang
Nomor	UC-3
Description	Kasus penggunaan ini digunakan untuk mengetahui status ketersediaan barang.
Tipe	Fungsional
Aktor	Pengguna <i>retail</i>
Kondisi Awal	Pengguna sudah masuk ke sistem
Kondisi Akhir	Sistem mengisi kolom jumlah <i>re-stock</i> dengan <i>recommended jumlah re-stock</i>
Alur Normal	<ol style="list-style-type: none"> 1. Pengguna membuka halaman utama aplikasi 2. Sistem menampilkan halaman utama aplikasi 3. Pengguna memilih menu config autostock 4. Sistem menampilkan halaman config produk 5. Pengguna memilih <i>recommended batas restock</i> 6. Sistem mengisi kolom batas <i>re-stock</i> secara otomatis 7. Kasus penggunaan berakhir.
Alur Alternatif	-



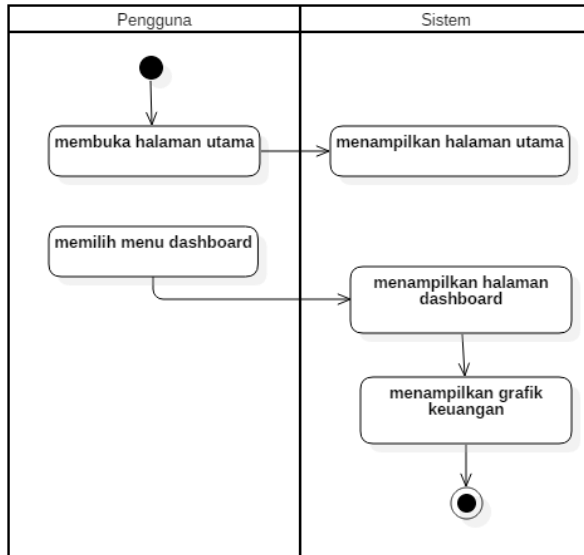
Gambar 3.7 Diagram aktivitas use case UC-3

3.2.3.4. Memberikan informasi dari laporan keuangan (UC-4)

Kasus penggunaan kode UC-4 diakses pengguna setelah berhasil masuk ke dalam sistem. Pada tahap ini pengguna akan diberikan informasi keuntungan bersih yang dibandingkan dari beberapa aspek. Spesifikasi penggunaan dan diagram aktivitas dapat dilihat pada Tabel 3.14 dan Gambar 3.8.

Tabel 3.14 Spesifikasi Kasus Penggunaan Memberikan Informasi Dari Laporan Keuangan

Nama	Memberikan informasi dari laporan keuangan
Nomor	UC-4
Description	Kasus penggunaan ini digunakan untuk mengetahui perbandingan keuntungan bersih dengan beberapa aspek penting.
Tipe	Fungsional
Aktor	Pengguna <i>retail</i>
Kondisi Awal	Pengguna sudah masuk ke sistem
Kondisi Akhir	Sistem menampilkan informasi keuangan pada halaman utama
Alur Normal	<ol style="list-style-type: none"> 1. Pengguna membuka halaman utama aplikasi 2. Sistem menampilkan halaman utama aplikasi 3. Pengguna memilih menu “<i>dashboard</i>” 4. Sistem menampilkan halaman dashboard beserta grafik pendapatan, grafik profit, return of asset, return od investment, dan return of equity. 5. Kasus penggunaan berakhir
Alur Alternatif	-



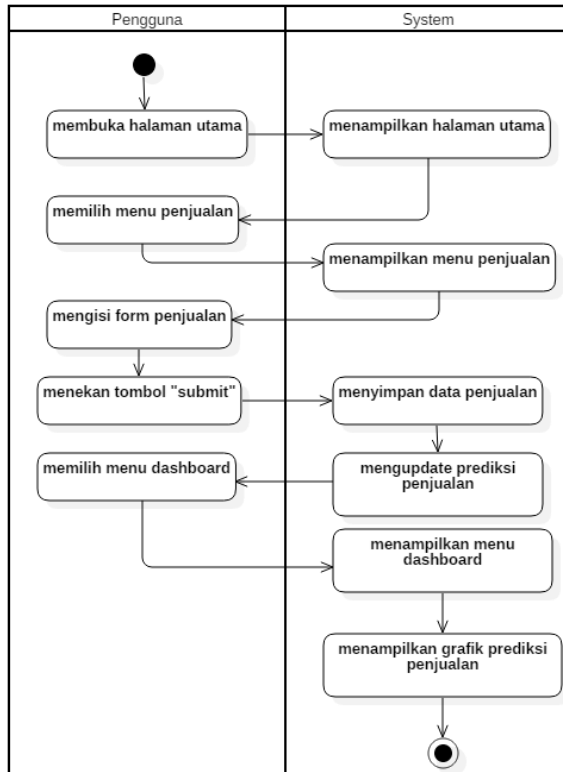
Gambar 3.8 Diagram aktivitas use case UC-4

3.2.3.5. Memberikan informasi tentang prediksi penjualan (UC-5)

Kasus penggunaan kode UC-5 diakses pengguna setelah berhasil melakukan proses pembelian dan penjualan barang. Pada tahap ini pengguna akan diberikan informasi prediksi penjualan dalam bentuk grafik. Spesifikasi penggunaan dan diagram aktivitas dapat dilihat pada Tabel 3.15 dan Gambar 3.9.

Tabel 3.15 Spesifikasi Kasus Penggunaan Memberikan Informasi Tentang Prediksi Penjualan

Nama	Memberikan informasi tentang prediksi penjualan
Nomor	UC-5
Description	Kasus penggunaan ini digunakan untuk mengetahui prediksi penjualan periode depan.
Tipe	Fungsional
Aktor	Pengguna <i>retail</i>
Kondisi Awal	Pengguna sudah masuk ke sistem
Kondisi Akhir	Sistem menampilkan grafik prediksi penjualan
Alur Normal	<ol style="list-style-type: none"> 1. Pengguna membuka halaman utama aplikasi 2. Sistem menampilkan halaman utama aplikasi 3. Pengguna memilih menu penjualan 4. Sistem menampilkan halaman penjualan 5. Pengguna mengisi form penjualan 6. Pengguna menekan tombol "Submit" 7. Sistem menyimpan data penjualan 8. Sistem meng-update prediksi penjualan 9. Pengguna memilih menu "<i>dashboard</i>" 10. Sistem menampilkan halaman dashboard beserta grafik prediksi penjualan. 11. Kasus penggunaan berakhir
Alur Alternatif	-



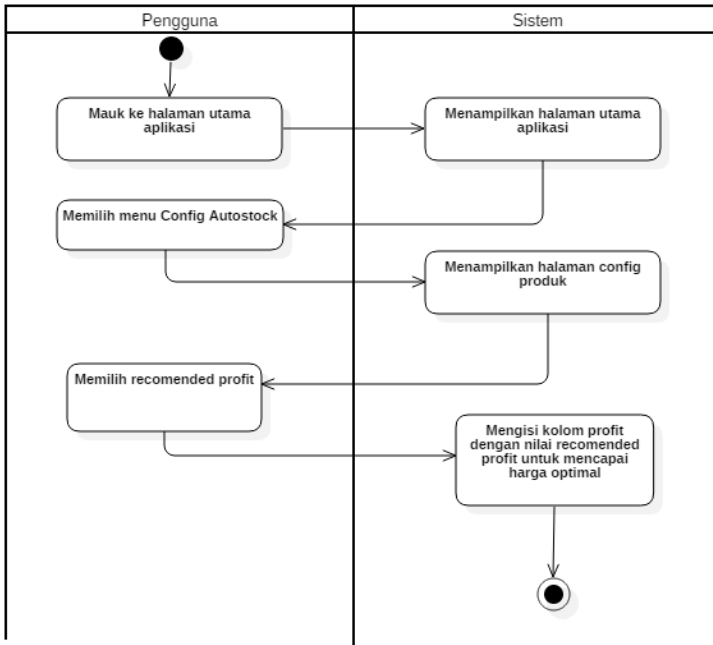
Gambar 3.9 Diagram aktivitas use case UC-5

3.2.3.6. Memberikan rekomendasi harga optimal (UC-6)

Kasus penggunaan kode UC-6 diakses pengguna setelah berhasil mengakes halaman config autostock. Pada tahap ini pengguna akan diberikan pilihan yang dapat merekomendasikan berapa profit optimal dari harga beli. Spesifikasi penggunaan dan diagram aktivitas dapat dilihat pada Tabel 3.16 dan Gambar 3.10.

Tabel 3.16 Spesifikasi Kasus Penggunaan Memberikan Rekomendasi Harga Optimal

Nama	Memberikan Rekomendasi Harga Optimal
Nomor	UC-6
Description	Kasus penggunaan ini digunakan untuk mengetahui rekomendasi harga optimal.
Tipe	Fungsional
Aktor	Pengguna <i>retail</i>
Kondisi Awal	Pengguna sudah masuk ke sistem
Kondisi Akhir	Sistem mengisi kolom profit dengan <i>recommended profit</i> untuk mencapai harga optimal
Alur Normal	<ol style="list-style-type: none"> 1. Pengguna membuka halaman utama aplikasi 2. Sistem menampilkan halaman utama aplikasi 3. Pengguna memilih menu config autostock 4. Sistem menampilkan halaman config produk 5. Pengguna memilih <i>recommended profit</i> 6. Sistem mengisi kolom profit secara otomatis 7. Kasus penggunaan berakhir.
Alur Alternatif	-



Gambar 3.10 Diagram aktivitas use case UC-6

3.3. Perancangan Sistem

Tahap ini meliputi perancangan basis data, tampilan antarmuka, dan perancangan alur proses penggunaan sistem yang diharapkan dapat memenuhi tujuan dari pengembangan aplikasi ini. Lingkungan pengembangan aplikasi adalah sebagai berikut.

Komputer	: Prosesor Intel® Xeon® CPU e5-2650 v4 (2.20GHz), RAM 2 GB
Sistem operasi	: Ubuntu 14.04
Basis Data	: MySQL 5.7

3.3.1. Perancangan Basis Data

Pada subbab ini dijelaskan mengenai perancangan basis data yang dalam hal ini digunakan untuk menyimpan data hasil proses perhitungan. Gambaran perancangan basis data dapat dilihat pada penjelasan di bawah. Saat ini, aplikasi masih menggunakan *database* relasional (MySQL) dikarenakan kekurangan sumber daya manusia, sedangkan seharusnya menggunakan *database* non relasional (NoSQL). Tabel 3.17 adalah gambaran tabel atribut dan tipe data yang diterapkan pada MySQL.

Tabel 3.17 Perancangan Basis Data

No	Tabel	Atribut	Keterangan
1	bi_supplier	id : <i>integer</i>	Tabel untuk menyimpan hasil perhitungan pada proses <i>supplier selection</i>
		supp_id : <i>integer</i>	
		barang_id : <i>varchar</i>	
		quality : <i>float</i>	
		cost : <i>float</i>	
		deliv : <i>float</i>	
		flex : <i>integer</i>	
		respon : <i>integer</i>	
2	bi_supplier_role	id : <i>integer</i>	Tabel untuk menyimpan data bobot indikator dalam <i>supplier selection</i>
		barang_id : <i>varchar</i>	
		quality : <i>float</i>	
		cost : <i>float</i>	
		deliv : <i>float</i>	
		flex : <i>float</i>	
		respon : <i>float</i>	
		id_retail : <i>integer</i>	
3	bi_eoq	id : <i>integer</i>	Tabel untuk menyimpan data hasil perhitungan EOQ
		barang_id : <i>varchar</i>	
		demand : <i>float</i>	
		cost : <i>float</i>	
		rent : <i>float</i>	
		eoq : <i>float</i>	
id_retail : <i>varchar</i>			

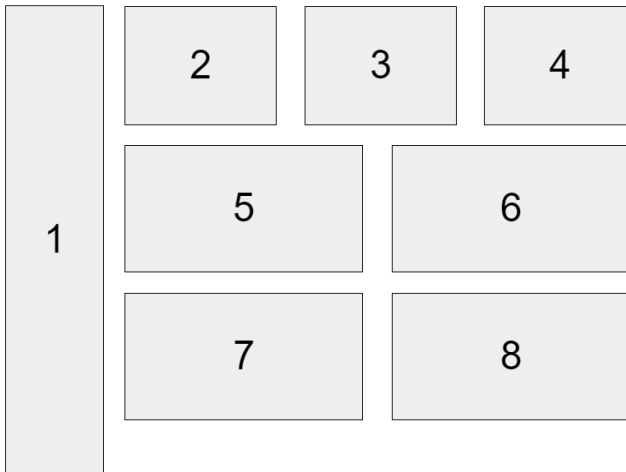
4	bi_rop	id : <i>integer</i>	Tabel untuk menyimpan ata hasil perhitungan ROP
		id_item : <i>varchar</i>	
		penjualan : <i>decimal</i>	
		tenggang_waktu : <i>integer</i>	
		safety_stock : <i>integer</i>	
		id_pemilih: <i>varchar</i>	
5	bi_forecast	id : <i>integer</i>	Tabel untuk menyimpan data hasil perhitungan <i>forecasting</i>
		tanggal_terakhir : <i>datetime</i>	
		day1 : <i>float</i>	
		day2 : <i>float</i>	
		day3 : <i>float</i>	
		day4 : <i>float</i>	
		day5 : <i>float</i>	
		day6 : <i>float</i>	
		day7 : <i>float</i>	
		id_retail : <i>varchar</i>	

3.3.2. Perancangan Antar Muka

Subbab ini menjelaskan bagaimana rancangan antarmuka yang akan berinteraksi secara langsung dengan pengguna pada saat tahap implementasi.

3.3.2.1. Perancangan Antarmuka Halaman Utama

Antarmuka halaman utama dirancang dengan tujuan untuk memudahkan melihat laporan dan informasi mengenai keadaan perusahaan sekarang. Pada halaman utama terdapat dua proses yang hasilnya disajikan dalam bentuk grafik, yaitu grafik untuk perbandingan keuntungan dan grafik untuk *forecasting* penjualan. Rancang antarmuka halaman utama ditunjukkan pada Gambar 3.10.



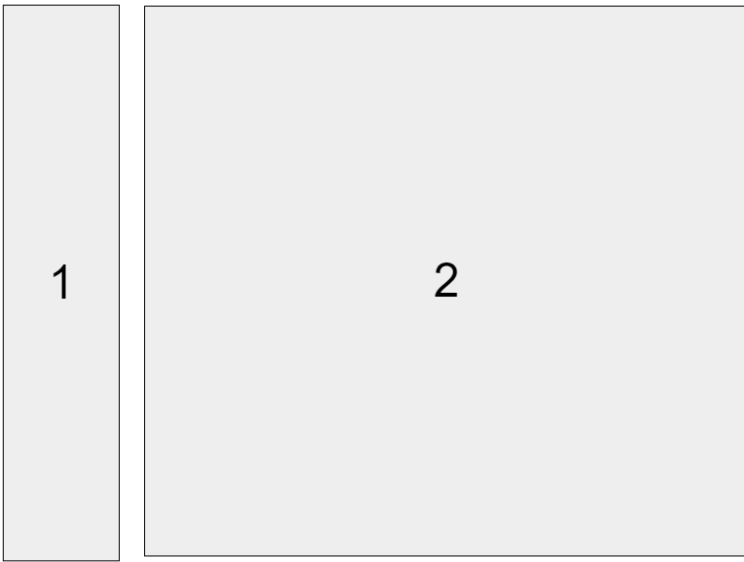
Gambar 3.11 Perancangan tampilan halaman utama

Komponen –komponen yang terdapat pada Gambar 3.10 adalah sebagai berikut:

1. *Sidebar Menu*
2. Persentase ROA
3. Persentase ROI
4. Persentase ROE
5. Grafik total pendapatan
6. Grafik total profit
7. Tabel list barang dengan pendapatan tertinggi
8. Tabel list barang dengan profit tertinggi

3.3.2.2. Perancangan Antarmuka Halaman Config Autostock

Antarmuka halaman config autostock dirancang dengan tujuan untuk mempermudah melakukan proses input pembelian. Pada halaman pembelian terdapat beberapa proses rekomendasi yang berjalan yaitu diantaranya proses *supplier selection* dan proses EOQ. Rancang antarmuka halaman pembelian ditunjukkan pada Gambar 3.11.



Gambar 3.12 Perancangan tampilan halaman pembelian

Komponen –komponen yang terdapat pada Gambar 3.11 adalah sebagai berikut:

1. *Sidebar Menu*
2. Tabel Config produk

[Halaman ini sengaja dikosongkan]

BAB IV IMPLEMENTASI

Bab ini membahas implementasi dari perancangan modul *Business Intelligence* pada ERP Retail. Di dalamnya mencakup penjelasan lingkungan pengembangan sistem, proses implementasi modul BI dan antarmuka pengguna.

4.1. Lingkungan Pengembangan Sistem

Lingkungan pengembangan sistem yang digunakan untuk mengembangkan tugas akhir ini dilakukan pada lingkungan dan kaskas sebagai berikut.

1. Basis data yang digunakan pada *server* adalah *MySQL version 5.7.22*.
2. PC untuk *server* menggunakan Intel® Xeon® CPU e5-2650 v4 (2.20GHz), RAM 2 GB dengan Sistem Operasi Ubuntu 14.04.
3. Mozilla Firefox 37.0.1 dan Chrome 66.0 sebagai antarmuka untuk pengujian aplikasi klien.

4.2. Implementasi Modul *Business Intelligence*

Pada bagian ini akan dijelaskan mengenai implementasi modul *business intelligence* pada aplikasi ERP *retail*. Implementasi modul tersebut akan dibagi menjadi beberapa proses, antara lain: *Supplier Selection*, *Economic Order Quantity*, *Re-Order Point*, Rangkuman Laporan Keuangan dan *Forecasting*. Secara rinci mengenai implementasi modul BI pada aplikasi ERP *retail* dijabarkan sebagai berikut:

4.2.1. Membuat Tabel-Tabel yang Dibutuhkan

Pembuatan tabel-tabel yang dibutuhkan untuk modul ini dilakukan dengan menjalankan “*CREATE TABLE*” melalui *MySQL*. Cara untuk membuat tabel-tabel tersebut yaitu dengan menjalankan perintah yang ditunjukkan pada Kode Sumber 4.1.

```

1. CREATE TABLE `bi_eoq` (
2.   `id` int(11) NOT NULL,
3.   `barang_id` varchar(100) NOT NULL,
4.   `demand` float NOT NULL,
5.   `cost` float NOT NULL,
6.   `rent` float NOT NULL,
7.   `eoq` float NOT NULL,
8.   `id_retail` int(11) NOT NULL
9. );
10.
11. CREATE TABLE `bi_forecast` (
12.   `id` int(11) NOT NULL,
13.   `tanggal_terakhir` date NOT NULL,
14.   `day1` float NOT NULL,
15.   `day2` float NOT NULL,
16.   `day3` float NOT NULL,
17.   `day4` float NOT NULL,
18.   `day5` float NOT NULL,
19.   `day6` float NOT NULL,
20.   `day7` float NOT NULL,
21.   `id_retail` int(11) NOT NULL
22. );
23.
24. CREATE TABLE `bi_rop` (
25.   `id` int(11) NOT NULL,
26.   `id_item` varchar(100) NOT NULL,
27.   `penjualan` decimal(10,2) NOT NULL DEFAULT
   '0.00',
28.   `tenggang_waktu` int(11) NOT NULL,
29.   `safety_stock` int(11) NOT NULL DEFAULT '0',
30.   `id_pemilik` varchar(200) NOT NULL
31. );
32.
33. CREATE TABLE `bi_supplier` (
34.   `id` int(11) NOT NULL,
35.   `supp_id` int(11) NOT NULL,
36.   `barang_id` varchar(20) NOT NULL,
37.   `quality` float NOT NULL,
38.   `cost` float NOT NULL,
39.   `deliv` float NOT NULL,
40.   `flex` int(11) NOT NULL,
41.   `respon` int(11) NOT NULL,
42.   `total` float NOT NULL DEFAULT '0',

```



```
43. `id_retail` int(11) NOT NULL
44. );
45.
46. CREATE TABLE `bi_supplier_rule` (
47.   `id` int(11) NOT NULL,
48.   `barang_id` varchar(100) NOT NULL,
49.   `quality` float DEFAULT NULL,
50.   `cost` float DEFAULT NULL,
51.   `deliv` float DEFAULT NULL,
52.   `flex` float DEFAULT NULL,
53.   `respon` float DEFAULT NULL,
54.   `id_retail` int(11) DEFAULT NULL
55. );
56.
57. ALTER TABLE `bi_eq`
58.   ADD PRIMARY KEY (`id`);
59.
60. ALTER TABLE `bi_forecast`
61.   ADD PRIMARY KEY (`id`);
62.
63. ALTER TABLE `bi_rop`
64.   ADD PRIMARY KEY (`id`);
65.
66. ALTER TABLE `bi_supplier`
67.   ADD PRIMARY KEY (`id`);
68.
69. ALTER TABLE `bi_supplier_rule`
70.   ADD PRIMARY KEY (`id`);
71.
72. ALTER TABLE `bi_eq`
73.   MODIFY `id` int(11) NOT NULL AUTO_INCREMENT;
74. ALTER TABLE `bi_forecast`
75.   MODIFY `id` int(11) NOT NULL AUTO_INCREMENT;
76. ALTER TABLE `bi_rop`
77.   MODIFY `id` int(11) NOT NULL AUTO_INCREMENT;
78. ALTER TABLE `bi_supplier`
79.   MODIFY `id` int(11) NOT NULL AUTO_INCREMENT;
80. ALTER TABLE `bi_supplier_rule`
81.   MODIFY `id` int(11) NOT NULL AUTO_INCREMENT;
```

Kode Sumber 4.1 Membuat Tabel BI

4.2.2. Implementasi Perhitungan ROP

Proses perhitungan ROP dapat dibedakan menjadi 3 tahap, yaitu pembuatan data ROP baru, update data ROP, dan pengambilan hasil ROP. Secara rinci akan dijabarkan sebagai berikut:

4.2.2.1. Pembuatan data ROP baru

Pembuatan data ROP baru akan dilakukan jika barang yang dibeli merupakan barang baru, sehingga perlu dibuat data baru pada tabel ROP. Kode sumber 4.2 merupakan implementasi pembuatan data baru pada tabel ROP.

```

1. public function create_rop($data){
2.     $this->db->reconnect();
3.     $query = $this->db->query("SELECT
   `id`,`id_item`,`id_pemilik` FROM `bi_rop` WHERE
   `id_item` = '$data[id_item]` AND `id_pemilik` =
   '$data[id_pemilik]`");
4.     if($query->num_rows()==0){
5.         $ok = $this->db-
>insert('bi_rop',$data);
6.         return $ok;
7.     }
8.     else{
9.         $child = $query->result();
10.        foreach ($child as $item)
11.        {
12.            $id = $item->id;
13.        }
14.        $this->db->where('id',$id);
15.        $ok = $this->db-
>update('bi_rop',$data);
16.        return $ok;
17.    }
18. })

```

Kode Sumber 4.2 Membuat Data ROP Baru

4.2.2.2. Update data ROP

Update data ROP akan dieksekusi jika barang yang dibeli merupakan barang yang dulu pernah dibeli sebelumnya, sehingga data pembelian baru digunakan sebagai data penunjang agar hasil perhitungan lebih baik. Kode sumber 4.3 merupakan implementasi update data ROP.

```

1. public function update_rop($data){
2.     $this->db->reconnect();
3.     $jumlah = 0;
4.     $query = $this->db->query("SELECT
   `id`,`id_item`,`id_pemilik` FROM `bi_rop` WHERE
   `id_item` = '$data[id_item]' AND `id_pemilik` =
   '$data[id_pemilik]'");
5.     $update_data = $this->db->query("SELECT
   penjualan.tanggal, SUM(detail_penjualan.jumlah) as
   jual, detail_penjualan.id_so FROM penjualan,
   detail_penjualan WHERE penjualan.id_so =
   detail_penjualan.id_so AND detail_penjualan.id_item
   = '$data[id_item]' AND penjualan.id_pemilik =
   '$data[id_pemilik]' GROUP BY penjualan.tanggal
   ORDER BY penjualan.tanggal DESC LIMIT 100 ");
6.     $avg_lt = $this->db->query("SELECT
   AVG(DATEDIFF(penerimaan_barang.tanggal_receive,purc
   hasing.tanggal_po)) as avg_lt FROM purchasing,
   penerimaan_barang WHERE purchasing.id_pemilik =
   '$data[id_pemilik]' AND purchasing.id_po in (SELECT
   detail_purchasing.id_purchasing from
   detail_purchasing where id_item = '$data[id_item]'
   AND id_pemilik = '$data[id_pemilik]') AND
   penerimaan_barang.id_po = purchasing.id_po ORDER BY
   penerimaan_barang.tanggal_receive DESC LIMIT 10");
7.     $data1 = $avg_lt->result();
8.     foreach ($data1 as $item)
9.     {
10.         $av_lt = $item->avg_lt;
11.     }
12.     $max_lt = $this->db->query("SELECT
   MAX(DATEDIFF(penerimaan_barang.tanggal_receive,purc
   hasing.tanggal_po)) as maxlt FROM purchasing,
   penerimaan_barang WHERE purchasing.id_pemilik =

```

```

'$data[id_pemilik]' AND purchasing.id_po in (SELECT
detail_purchasing.id_purchasing from
detail_purchasing where id_item = '$data[id_item]'
AND id_pemilik = '$data[id_pemilik]') AND
penerimaan_barang.id_po = purchasing.id_po");
13.     $data2 = $max_lt->result();
14.     foreach ($data2 as $item)
15.     {
16.         $maxlt = $item->maxlt;
17.     }
18.     $max_penj = $this->db->query("SELECT
MAX(A.max_jual) as max_jual FROM (SELECT
SUM(detail_penjualan.jumlah) as max_jual FROM
penjualan, detail_penjualan WHERE penjualan.id_so =
detail_penjualan.id_so AND detail_penjualan.id_item
= 'data[id_item]' AND penjualan.id_pemilik =
'data[id_pemilik]' GROUP BY penjualan.tanggal)A");
19.     $data3 = $max_penj->result();
20.     foreach ($data3 as $item)
21.     {
22.         $max_jual = $item->max_jual;
23.     }
24.     $jual = $update_data->result();
25.     foreach ($jual as $item) {
26.         $jumlah = $jumlah+(int)$item->jual;
27.     }
28.     $data_jual = $jumlah/$update_data-
>num_rows();
29.     $child = $query->result();
30.     foreach ($child as $item)
31.     {
32.         $id = $item->id;
33.     }
34.     $data_rop = array(
35.         'id_item' => $data['id_item'],
36.         'id_pemilik' => $data['id_pemilik'],
37.         'penjualan' => $data_jual,
38.         'safety_stock' => ($max_jual*$maxlt)-
($data_jual*$av_lt),
39.     );
40.     $this->db->where('id',$id);
41.     $ok = $this->db-
>update('bi_rop',$data_rop);

```

```

42.         return $ok;
43.     }

```

Kode Sumber 4.3 Update Data ROP

4.2.2.3. Pengambilan hasil ROP

Pengambilan hasil ROP dilakukan saat pengguna mengakses halaman *stock items*. Hasil perhitungan yang disimpan dalam bentuk angka diterjemahkan menjadi keterangan untuk pengguna. Kode sumber 4.4 merupakan implementasi pengambilan hasil ROP.

```

1. public function
   stockItem($page,$per_page,$id_pemilik){
2.         $this->db->reconnect();
3.         $this->db-
   >select('item_master.id_item, nama_item,
   sum(jumlah) as stock,
   ((bi_rop.penjualan*bi_rop.tenggang_waktu)+bi_rop.sa
   fety_stock) as rop, bi_rop.safety_stock as ss');
4.         $this->db-
   >where('item_master.id_pemilik',$id_pemilik);
5.         $this->db->from('item_master');
6.         $this->db-
   >join('gudang','gudang.id_item =
   item_master.id_item','INNER');
7.         $this->db-
   >join('bi_rop','bi_rop.id_item =
   item_master.id_item AND bi_rop.id_pemilik
   =' . $id_pemilik, 'LEFT');
8.         $this->db-
   >group_by('item_master.id_item');
9.         $this->db->limit($per_page, $page);
10.        $read = $this->db->get();
11.        foreach ($read->result() as $data) {
12.            $hasil[] = $data;
13.        }
14.        return $hasil;}

```

Kode Sumber 4.4 Mengambil hasil ROP

4.2.3. Implementasi Perhitungan EOQ

Proses perhitungan EOQ dilakukan pada setelah pengguna melakukan penjualan. Perhitungan dilakukan secara otomatis dengan cara mengambil data dari penjualan, detail penjualan serta data aset. Data tersebut kemudian dihitung berdasarkan rumus dan hasilnya disimpan ke dalam tabel bi_eoq. Kode 4.5 merupakan implementasi dari perhitungan EOQ.

```

1. public function update_eoq($data){
2.     $this->db->reconnect();
3.     $jumlah = 0;
4.     $query = $this->db->query("SELECT
   `id`,`barang_id`,`id_retail` FROM `bi_eoq` WHERE
   `barang_id` = '$data[id_item]` AND `id_retail` =
   '$data[id_pemilik]`");
5.     $update_data = $this->db->query("SELECT
   penjualan.tanggal, SUM(detail_penjualan.jumlah) as
   jual, detail_penjualan.id_so FROM penjualan,
   detail_penjualan WHERE penjualan.id_so =
   detail_penjualan.id_so AND detail_penjualan.id_item
   = '$data[id_item]` AND penjualan.id_pemilik =
   '$data[id_pemilik]` GROUP BY penjualan.tanggal
   ORDER BY penjualan.tanggal DESC LIMIT 100 ");
6.     $hold = $this->db->query("SELECT
   (`asset_value`/`asset_duration`)/`asset_capacity`
   FROM `asset` WHERE `asset_name` = 'Gedung' AND
   `id_retail` = '$data[id_pemilik]` AND `is_delete`=0
   AND `status` = 1 ");
7.     $data1 = $hold->result();
8.     if (is_null($data1)) {
9.         $hold_c = 1;
10.    }
11.    else{
12.        foreach ($data1 as $item)
13.        {
14.            $hold_c = $item->hold;
15.        }
16.    }
17.    $jual = $update_data->result();

```

```

18.         foreach ($jual as $item) {
19.             $jumlah = $jumlah+(int)$item->jual;
20.         }
21.         $data_jual = $jumlah/$update_data-
>num_rows();
22.         $demand = $data_jual*365;
23.         $child = $query->result();
24.         foreach ($child as $item)
25.         {
26.             $id = $item->id;
27.         }
28.         $data_eoq = array(
29.             'barang_id' => $data['id_item'],
30.             'id_retail' => $data['id_pemilik'],
31.             'demand' => $data_jual,
32.             'cost' => $data['ongkir'],
33.             'rent' => $hold_c
34.             'eoq' =>
sqrt((2*$data_jual*$data['ongkir'])/$hold_c),
35.         );
36.         $this->db->where('id',$id);
37.         $ok = $this->db-
>update('bi_eoq',$data_rop);
38.         return $ok;
39.     }
40.

```

Kode Sumber 4.5 Implementasi Perhitungan EOQ

4.2.1. Implementasi *Supplier Selection*

Proses perhitungan *supplier selection* dilakukan pada saat pengguna membuka halaman utama. Perhitungan dilakukan secara otomatis dengan cara mengambil data dari supplier, detail supplier serta supplier rule. Data tersebut kemudian dioperasikan dan hasilnya dimasukkan kedalam tabel *bi_supplier*. Kode 4.6 merupakan implementasi dari perhitungan *supplier selection*.

```

1. public function insert_data()
2.     {
3.         $id_retail = $this->session-
>userdata('id_retail');
4.         $this->db->reconnect();
5.         $this->db->select('suplier.id_suplier as
supp_id, jangka_waktu, ongkir, config_value,
id_item, harga, quality, cost, deliv, flex,
respon');
6.         $this->db->where('suplier.id_pemilik',0);
7.         $this->db->from('suplier');
8.         $this->db-
>join('detail_suplier','detail_suplier.id_suplier =
suplier.id_suplier','LEFT');
9.         $this->db-
>join('config','config.config_code =
suplier.return_barang','LEFT');
10.        $this->db-
>join('bi_supplier_rule','barang_id = id_item AND
id_retail='.$id_retail,'LEFT');
11.        $read = $this->db->get();
12.        foreach ($read->result() as $data)
13.            {
14.                $hasil[] = $data;
15.            }
16.        foreach ($hasil as $item) {
17.            $data = array(
18.                'supp_id' => $item->supp_id,
19.                'barang_id' => $item->id_item,
20.                'quality' => $item-
>config_value*$item->quality,
21.                'cost' => ($item->ongkir+$item-
>harga)*(-($item->cost)),
22.                'deliv' => $item->jangka_waktu*(-
($item->deliv)),
23.                'flex' => $item-
>config_value*$item->flex,
24.                'respon' =>$item-
>config_value*$item->respon,
25.                'total' => ($item-
>config_value*$item->quality)+(($item-
>ongkir+$item->harga)*(-($item->cost)))+($item-
>jangka_waktu*(-($item->deliv)))+($item-

```



```

>config_value*$item->flex)+($item-
>config_value*$item->respon),
26.         'id_retail' => $id_retail,
27.         );
28.         $this->db->insert('bi_supplier',$data);
29.     }
30.     return 'success';
31. }]}
32.

```

Kode Sumber 4.6 Implementasi *Supplier Selection*

4.2.2. Implementasi Rangkuman Laporan Keuangan

Rangkuman laporan keuangan dibagi menjadi 3 grafik yaitu, *profit margin*, *total assets turn over*, dan *return of asset*. Masing-masing grafik menggunakan fungsi yang berbeda-beda. Kode 4.7 merupakan implementasi perhitungan *profit margin*. Sedangkan kode 4.8 merupakan implementasi perhitungan *total assets turn over*. Kode 4.9 merupakan implementasi perhitungan *return of assets*. Ketiga perhitungan ini dilakukan secara otomatis saat pengguna membuka halaman *dashboard* menggunakan data keuangan yang tersimpan di tabel *gl_jurnal_l* dan *gl_jurnal_h*.

```

1. public function roa($uid){
2.     $this->db->reconnect();
3.     $game_id = $this->session-
>userdata('game_id');
4.     $query = $this->db->query("select
(A.total-(B.expenses+D.total)) as net, C.total as
totalc from (SELECT (sum(l.line_credit)-
sum(l.line_debit)) as total from gl_jurnal_l l,
gl_jurnal_h h where l.journal_id=h.id and
l.acc_id=4000 and h.uid='$uid' and
h.game_id='$game_id')A, (SELECT
(sum(e.expense_jumlah*e.expense_value)) as expenses
from expense e where e.id_pemilik='$uid' and
e.game_id='$game_id')B, (SELECT (sum(l.line_debit)-

```

```

sum(l.line_credit)) as total from gl_journal_l l,
gl_journal_h h where l.journal_id=h.id and l.acc_id
like '1%' and h.uid='$uid' and
h.game_id='$game_id')C, (SELECT (sum(l.line_debit)-
sum(l.line_credit)) as total from gl_journal_l l,
gl_journal_h h where l.journal_id=h.id and l.acc_id
like '5%' and h.uid='$uid' and
h.game_id='$game_id')D");
5.         if ($query->num_rows() > 0)
6.         {
7.             foreach ($query->result() as $row)
8.             {
9.                 $hasil[] = $row;
10.            }
11.            return $hasil;
12.        }
13.        else{
14.            return 0;
15.        }
16.    }
17.

```

Kode Sumber 4.7 Implementasi Perhitungan *Return of Assets*

```

1. public function roi($uid){
2.     $this->db->reconnect();
3.     $game_id = $this->session-
>userdata('game_id');
4.     $query = $this->db->query("select (A.total-
(B.expenses+D.total)) as net, C.total as totalc
from (SELECT (sum(l.line_credit)-
sum(l.line_debit)) as total from gl_journal_l l,
gl_journal_h h where l.journal_id=h.id and
l.acc_id=4000 and h.uid='$uid' and
h.game_id='$game_id')A, (SELECT
(sum(e.expense_jumlah*e.expense_value)) as expenses
from expense e where e.id_pemilik='$uid' and
e.game_id='$game_id')B, (SELECT
(sum(l.line_credit)- sum(l.line_debit)) as total
from gl_journal_l l, gl_journal_h h where
l.journal_id=h.id and l.acc_id like '2%' and

```

```

h.uid='$uid' and h.game_id='$game_id')C, (SELECT
(sum(l.line_debit)- sum(l.line_credit)) as total
from gl_journal_l l, gl_journal_h h where
l.journal_id=h.id and l.acc_id like '5%' and
h.uid='$uid' and h.game_id='$game_id')D");
5.     if ($query->num_rows() > 0)
6.         {
7.         foreach ($query->result() as $row)
8.         {
9.             $hasil[] = $row;
10.        }
11.        return $hasil;
12.    }
13.    else{
14.        return 0;
15.    }
16. }
17.
18.

```

Kode Sumber 4.8 Implementasi Perhitungan ROI

```

1. public function roe($uid){
2.     $this->db->reconnect();
3.     $game_id = $this->session-
>userdata('game_id');
4.     $query = $this->db->query("select
(A.total-(B.expenses+D.total)) as net, C.total as
totalc from (SELECT (sum(l.line_credit)-
sum(l.line_debit)) as total from gl_journal_l l,
gl_journal_h h where l.journal_id=h.id and
l.acc_id=4000 and h.uid='$uid' and
l.game_id='$game_id')A, (SELECT
(sum(e.expense_jumlah*e.expense_value)) as expenses
from expense e where e.id_pemilik='$uid' and
e.game_id='$game_id')B, (SELECT
(sum(l.line_credit)- sum(l.line_debit)) as total
from gl_journal_l l, gl_journal_h h where
l.journal_id=h.id and l.acc_id like '3%' and
h.uid='$uid' and h.game_id='$game_id')C, (SELECT
(sum(l.line_debit)- sum(l.line_credit)) as total

```

```

from gl_journal_l l, gl_journal_h h where
l.journal_id=h.id and l.acc_id like '5%' and
h.uid='$uid' and h.game_id='$game_id')D");
5.         if ($query->num_rows() > 0)
6.         {
7.             foreach ($query->result() as $row)
8.             {
9.                 $hasil[] = $row;
10.            }
11.            return $hasil;
12.        }
13.        else{
14.            return 0;
15.        }
16.    }
17.

```

Kode Sumber 4.9 Implementasi Perhitungan *Return of Equity*

4.2.3. Implementasi *Forecasting* Penjualan

Proses perhitungan *forecasting* dilakukan pada saat pengguna membuka halaman utama. Perhitungan dilakukan secara otomatis dengan cara mengambil data dari basis data dan disimpan langsung menuju basis data. Proses perhitungan menggunakan *Holt-Winter* ini dibangun dalam bentuk bahasa python. Data tersebut kemudian dioperasikan dan hasilnya dimasukkan kedalam tabel *bi_forecast*. Kode 4.10 merupakan implementasi dari perhitungan *forecast* penjualan.

```

1. #!/usr/bin/env python
2. import pandas as pd
3. import numpy as np
4. import math
5. import sys;
6. import gc
7. import pprint as pprint
8. import datetime
9. import MySQLdb
10. import matplotlib.pyplot as plt

```

```

11. from statsmodels.tsa.holtwinters import
    ExponentialSmoothing
12. uid = sys.argv[1];
13. game_id = sys.argv[2];
14. id_item = sys.argv[3];
15. conn = MySQLdb.connect(host= "localhost",
16.                         user="root",
17.                         passwd="hominatest",
18.                         db="retail_game")
19. #print(sys.argv[1]);
20.
21. if uid != None :
22.     sql = 'SELECT * FROM (SELECT
    penjualan.tanggal as ds,
    SUM(detail_penjualan.jumlah) as y FROM penjualan,
    detail_penjualan WHERE penjualan.id_so =
    detail_penjualan.id_so AND penjualan.id_pemilik =
    %s and penjualan.game_id= %s and
    detail_penjualan.id_item = %s GROUP BY
    penjualan.tanggal ORDER BY penjualan.tanggal DESC
    LIMIT 30) A ORDER BY ds ASC'
23.     df = pd.read_sql(sql, con=conn,
    params=(uid,game_id,id_item), index_col='ds');
24.     df.index.freq = 'D'
25.     train= df.iloc[:30,0]
26.     start = train.index[-1]
27.     end = train.index[-1]+ datetime.timedelta(7)
28.     model = ExponentialSmoothing(train,
    seasonal='mul', seasonal_periods=7).fit()
29.     pred = model.predict(start=start, end=end)
30.     a_list = [];
31.     for p in range (0, 7):
32.         a_list.append(pred[p]);
33.     sdate = pred.index[0]
34.     try:
35.         cur = conn.cursor();
36.         cur.execute("""SELECT `id` FROM
    `bi_forecast` WHERE `id_pemilik`=%s and `game_id` =
    %s and id_item=%s""", (uid,game_id,id_item));
37.         row = cur.fetchone();
38.         if row is not None:
39.             for ind in cur:
40.                 up_id = ind[0];

```

```

41.             cur.execute("""UPDATE
    `bi_forecast` SET
    `start_date`=%s,`day1`=%s,`day2`=%s,`day3`=%s,`day4`
    `=%s,`day5`=%s,`day6`=%s,`day7`=%s WHERE
    `id`=%s""",(sdate,a_list[0],a_list[1],a_list[2],a_l
    ist[3],a_list[4],a_list[5],a_list[6],up_id));
42.             # print("lala");
43.             else :
44.                 cur.execute("""INSERT INTO
    `bi_forecast`(`id_item`,`start_date`,
    `id_pemilik`,`game_id`,`day1`,`day2`,`day3`,
    `day4`,`day5`,`day6`,`day7`) VALUES
    (%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s)""",(id_item,sdat
    e,uid,game_id,a_list[0],a_list[1],a_list[2],a_list[
    3],a_list[4],a_list[5],a_list[6]));
45.                 print('Success');
46.             except MySQLdb.Error as e:
47.                 print(e);
48.                 sys.exit(1);
49.             finally:
50.                 if conn:
51.                     conn.commit();
52.                     conn.close()
53.

```

Kode Sumber 4.10 Implementasi *Forecast* Penjualan

4.2.4. Implementasi Perhitungan Harga Optimal

Proses perhitungan harga optimal dilakukan pada saat pengguna membuka halaman utama. Perhitungan dilakukan secara otomatis dengan cara mengambil data dari basis data dan disimpan langsung menuju basis data. Kode 4.11 merupakan implementasi perhitungan *price elasticity of demand*. Sedangkan kode 4.12 merupakan implementasi perhitungan *marginal cost*.

```
1. public function searchHargaMin($id_item)
2.     {
3.         $uid = $this->session-
>userdata('id_retail');
4.         $game_id = $this->session-
>userdata('game_id');
5.         $this->db->reconnect();
6.         $query2 = $this->db->query("SELECT harga,
round(avg(total)) as q FROM (SELECT p.tanggal,
d.harga, sum(d.jumlah) as total FROM
detail_penjualan d, penjualan p where
p.id_so=d.id_so and d.id_item='$id_item' and
p.id_pemilik='$uid' and p.game_id='$game_id' group
by p.tanggal, d.harga)A GROUP by harga ORDER BY
`A`.`harga` ASC limit 1");
7.         if ($query2->num_rows() > 0)
8.         {
9.             return $query2->row();
10.        }
11.        else{
12.            return 0;
13.        }
14.    }
15.
16.    public function searchHargaMax($id_item)
17.    {
18.        $uid = $this->session-
>userdata('id_retail');
19.        $game_id = $this->session-
>userdata('game_id');
20.        $this->db->reconnect();
21.        $query2 = $this->db->query("SELECT harga,
round(avg(total)) as q FROM (SELECT p.tanggal,
d.harga, sum(d.jumlah) as total FROM
detail_penjualan d, penjualan p where
p.id_so=d.id_so and d.id_item='$id_item'and
p.id_pemilik='$uid' and p.game_id='$game_id' group
by p.tanggal, d.harga)A GROUP by harga ORDER BY
`A`.`harga` DESC limit 1");
22.
23.        if ($query2->num_rows() > 0)
24.        {
25.            return $query2->row();
```

```

26.         }
27.         else{
28.             return 0;
29.         }
30.     }
31.
32.     public function ped($uid, $data){
33.         $this->db->reconnect();
34.         $game_id = $this->session-
>userdata('game_id');
35.         $this->db->select();
36.         $this->db->from('bi_ped');
37.         $this->db->where('id_pemilik',$uid);
38.         $this->db->where('game_id',$game_id);
39.         $tes = $this->db->get();
40.         if ($tes->num_rows()==0) {
41.             $d_item = $this->db->query("SELECT *
FROM `item_master`");
42.             foreach ($d_item->result() as $item) {
43.                 $max = $this->searchHargaMax($item-
>id_item);
44.                 $min = $this->searchHargaMin($item-
>id_item);
45.                 $ok = $this->db->query("INSERT INTO
`bi_ped`(`id_item`, po, pi, qo, qi, `id_pemilik`,
`game_id`) VALUES ('$item->id_item','$min->harga,
$max->harga, $min->q, $max->q,'$uid','$game_id')");
46.             }
47.         }
48.         $this->db->select();
49.         $this->db->from('bi_ped');
50.         $this->db->where('id_pemilik',$uid);
51.         $this->db->where('game_id',$game_id);
52.         $this->db-
>where('id_item',$data['id_item']);
53.         $data2 = $this->db->get();
54.         $ped = $data2->result();
55.
56.         $mc = $this->real_mc($data['id_item']);
57.         if ($data['price']<$ped[0]->po) {
58.             $ped_val = (($ped[0]->qo-
$data['quantity'])/($ped[0]->pi-
$data['price']))*(($ped[0]-

```



```

    >pi+$data['price'])/($ped[0]-
    >qi+$data['quantity']));
59.         $op_val = $mc*($ped_val/($ped_val+1));
60.         $this->db->query("UPDATE `bi_ped` SET
    `po`=$data[price]', `qo`=$data[quantity]', `ped`=$
    ped_val', `op`=$op_val WHERE
    `id_item`=$data[id_item]' AND `id_pemilik` =
    '$uid'");
61.     } elseif ($data['price']>$ped[0]->pi) {
62.         $ped_val = (($data['quantity']-$ped[0]-
    >qo)/($data['price']-$ped[0]-
    >po))*(($data['price']+$ped[0]-
    >po)/($data['quantity']+$ped[0]->qi));
63.         $op_val = $mc*($ped_val/($ped_val+1));
64.         $this->db->query("UPDATE `bi_ped` SET
    `pi`=$data[price]', `qi`=$data[quantity]', `ped`=$
    ped_val', `op`=$op_val WHERE
    `id_item`=$data[id_item]' AND `id_pemilik` =
    '$uid'");
65.     }
66.     return true;
67. }

```

Kode Sumber 4.11 Implementasi Perhitungan *Price Elasticity of Demand*

```

1. public function hpp_satu($uid,$id_item){
2.     $this->db->reconnect();
3.     $game_id = $this->session-
    >userdata('game_id');
4.     $query2 = $this->db->query("SELECT
    s.harga as harga FROM detail_supplier s,
    config_product c where s.id_item=c.id_item and
    s.id_supplier= c.supplier and c.id_item = '$id_item'
    and c.id_pemilik = '$uid' and c.game_id =
    '$game_id'");
5.     if ($query2->num_rows() > 0)
6.     {
7.         return $query2->row();
8.     }
9.     else{

```

```

10.         return 0;
11.     }
12. }
13.
14.     public function fore_all($uid){
15.         $this->db->reconnect();
16.         $game_id = $this->session-
>userdata('game_id');
17.         $query2 = $this->db->query("SELECT
sum(day1) as total FROM `bi_forecast` where
id_pemilik='$uid' and game_id='$game_id'");
18.         if ($query2->num_rows() > 0)
19.         {
20.             return $query2->row();
21.         }
22.         else{
23.             return 0;
24.         }
25.     }
26.
27.     public function fore_satu($uid,$id_item){
28.         $this->db->reconnect();
29.         $game_id = $this->session-
>userdata('game_id');
30.         $query2 = $this->db->query("SELECT day1
FROM `bi_forecast` where id_pemilik='$uid' and
game_id='$game_id' and id_item='$id_item'");
31.         if ($query2->num_rows() > 0)
32.         {
33.             return $query2->row();
34.         }
35.         else{
36.             return 0;
37.         }
38.     }
39.
40.     public function expenses($uid){
41.         $this->db->reconnect();
42.         $game_id = $this->session-
>userdata('game_id');
43.         $query2 = $this->db->query("SELECT
(sum(e.expense_jumlah*e.expense_value)) as expenses

```

```

from expense e where e.id_pemilik='$uid' and
e.game_id='$game_id'");
44.         if ($query2->num_rows() > 0)
45.         {
46.             foreach ($query2->result() as $row)
47.             {
48.                 $hasil2 = $row;
49.             }
50.             return $hasil2;
51.         }
52.         else{
53.             return 0;
54.         }
55.     }
56.
57.     public function real_mc($id_item){
58.         $this->db->reconnect();
59.         $uid = $this->session-
>userdata('id_retail');
60.         $game_id = $this->session-
>userdata('game_id');
61.         $exp = $this->expenses($uid);
62.         $fore_tot = $this->fore_all($uid);
63.         $hpp = $this->hpp_satu($uid,$id_item);
64.         $fore_one = $this->fore_satu($uid,
$id_item);
65.         if ($fore_tot->total==0) {
66.             return 0;
67.         } else {
68.             $exp_satu = $exp->expenses/$fore_tot-
>total;
69.             $hpp_tot = $hpp->harga*$fore_one->day1;
70.             $mc = ($exp_satu + $hpp_tot)/$fore_one-
>day1;
71.             return $mc;
72.         }
73.     }
74.

```

Kode Sumber 4.12 Implementasi Perhitungan *Marginal Cost*

4.3. Implementasi Antarmuka

Pada subbab ini dijelaskan implementasi antarmuka yang terdapat pada Tugas Akhir ini. Implementasi antarmuka ditunjukkan sebagai berikut:

4.3.1. Antarmuka Halaman Utama

Setelah berhasil *login*, pengguna akan diarahkan langsung menuju halaman ini. Pada halaman ini terdapat grafik keuangan dan grafik *forecast*. Tampilan antarmuka ini dapat dilihat pada Lampiran Gambar A.1.

4.3.2. Antarmuka tampilan halaman Config Autostock

Pada halaman ini, pengguna dapat melakukan pengaturan untuk autostock. Halaman ini juga dilengkapi dengan rekomendasi dari hasil perhitungan modul BI. Tampilan antarmuka ini dapat dilihat pada Lampiran Gambar A.2.

BAB V

PENGUJIAN DAN EVALUASI

Bab ini membahas pengujian dan evaluasi pada aplikasi yang dikembangkan. Pengujian yang dilakukan adalah pengujian terhadap kebutuhan fungsionalitas sistem yang telah dijabarkan pada Bab III dan terhadap tujuan dibuatnya aplikasi ini, yakni agar musik memiliki informasi yang lebih mendetail dan makin lengkap.

5.1. Lingkungan Pengujian

Lingkungan pengujian sistem pada pengerjaan Tugas Akhir ini dilakukan pada lingkungan dan alat kaku sebagai berikut:

Prosesor : Prosesor Intel® Xeon® CPU-E5-2650v4
RAM : 2 GB
Jenis *Device* : Virtual Server
Sistem Operasi : Ubuntu 16.04 LTS

5.2. Skenario Pengujian

Pada bagian ini akan dijelaskan tentang skenario pengujian yang dilakukan. Pengujian dilakukan dalam beberapa tahapan yang dijelaskan dalam subbab ini.

5.2.1. Pengujian Fungsionalitas

Pengujian fungsionalitas aplikasi dilakukan secara mandiri dengan melakukan skenario yang sama dengan rancangan alur proses aplikasi sebagai tolok ukur keberhasilan pengujian, dan mengacu pada kasus penggunaan yang sebelumnya telah dijelaskan pada Bab III. Pengujian pada kebutuhan fungsionalitas dapat dijabarkan pada subbab berikut.

5.2.1.1. Pengujian Menampilkan Rekomendasi Supplier

Pengujian ini merupakan pengujian menampilkan rekomendasi supplier dari sistem. Pengujian ini dilakukan untuk menguji apakah sistem mampu menampilkan hasil rekomendasi supplier. Rincian skenario pengujian pada kasus penggunaan dapat dilihat pada Tabel 5.1.

Tabel 5.1 Pengujian Menampilkan Rekomendasi Supplier

ID	SP-UC1
Referensi Kasus Penggunaan	UC-1
Nama	Menampilkan rekomendasi supplier.
Tujuan Pengujian	Menguji kemampuan sistem dalam menampilkan rekomendasi supplier.
Skenario	Menampilkan rekomendasi supplier
Kondisi Awal	Pengguna berada pada menu <i>dashboard</i> , memilih menu config autostock.
Data Uji	Data masukkan pengguna
Langkah Pengujian	<ol style="list-style-type: none"> 1. Pengguna memilih menu config autostock. 2. Pengguna memilih tombol <i>recommended supp.</i> 3. Sistem mengambil data rekomendasi supplier berdasarkan nama item. 4. Sistem mengisi kolom supplier dengan rekomendasi supplier secara otomatis.
Hasil Yang Diharapkan	Data rekomendasi supplier dapat ditampilkan.
Hasil Yang Didapatkan	Data rekomendasi supplier dapat ditampilkan.

Hasil Pengujian	Berhasil.
Kondisi Akhir	Aktor melihat kolom supplier berisi rekomendasi supplier.

5.2.1.2. Pengujian Menampilkan Jumlah *Re-stock* yang Ekonomis

Pengujian ini dilakukan untuk menguji apakah sistem mampu menampilkan jumlah *re-stock* yang ekonomis kepada pengguna. Rincian skenario pengujian pada kasus penggunaan dapat dilihat pada Tabel 5.2.

Tabel 5.2 Pengujian Menampilkan Jumlah *Re-stock*

ID	SP-UC2
Referensi Kasus Penggunaan	UC-2
Nama	Menampilkan Jumlah <i>Re-stock</i> Ekonomis.
Tujuan Pengujian	Menguji kemampuan sistem dalam menampilkan jumlah <i>re-stock</i> yang ekonomis kepada pengguna
Skenario	Menampilkan jumlah <i>re-stock</i> ekonomis
Kondisi Awal	Pengguna berada pada menu <i>dashboard</i> , memilih menu config autostock.
Data Uji	Data masukan pengguna
Langkah Pengujian	<ol style="list-style-type: none"> 1. Pengguna memilih menu config autostock. 2. Pengguna memilih tombol <i>recommended</i> jumlah. 3. Sistem mengambil data rekomendasi jumlah <i>re-stock</i> berdasarkan nama item. 4. Sistem mengisi kolom jumlah dengan rekomendasi jumlah secara otomatis.

Hasil Yang Diharapkan	Data jumlah <i>re-stock</i> berhasil ditampilkan
Hasil Yang Didapatkan	Data jumlah <i>re-stock</i> berhasil ditampilkan
Hasil Pengujian	Berhasil.
Kondisi Akhir	Aktor melihat kolom jumlah berisi rekomendasi jumlah <i>re-stock</i> .

5.2.1.3. Pengujian Menampilkan Batas *Re-stock*

Pengujian ini dilakukan untuk menguji kemampuan sistem untuk menampilkan data rekomendasi batas *re-stock*. Rincian skenario pengujian pada kasus penggunaan dapat dilihat pada Tabel 5.3.

Tabel 5.3 Pengujian Menampilkan Batas *Re-stock*

ID	SP-UC3
Referensi Kasus Penggunaan	UC-3
Nama	Menampilkan status <i>stock</i> .
Tujuan Pengujian	Menguji kemampuan sistem dalam menampilkan batas <i>re-stock</i> .
Skenario	Menampilkan batas <i>re-stock</i>
Kondisi Awal	Pengguna berada pada menu <i>dashboard</i> , memilih menu config autostock.
Data Uji	Data masukan pengguna
Langkah Pengujian	<ol style="list-style-type: none"> 1. Pengguna memilih menu config autostock. 2. Pengguna memilih tombol <i>recommended</i> batas. 3. Sistem mengambil data rekomendasi batas <i>re-stock</i> berdasarkan nama item. 4. Sistem mengisi kolom batas dengan rekomendasi batas <i>re-stock</i> secara otomatis.

Hasil Yang Diharapkan	Data batas <i>re-stock</i> berhasil ditampilkan
Hasil Yang Didapatkan	Data batas <i>re-stock</i> berhasil ditampilkan
Hasil Pengujian	Berhasil.
Kondisi Akhir	Aktor melihat kolom batas berisi rekomendasi batas <i>re-stock</i> .

5.2.1.4. Pengujian Menampilkan Rangkuman Laporan Keuangan.

Pengujian ini dilakukan untuk menguji kemampuan sistem dalam menampilkan rangkuman laporan keuangan. Rincian skenario pengujian pada kasus penggunaan dapat dilihat pada Tabel 5.4.

Tabel 5.4 Pengujian Menampilkan Rangkuman Laporan Keuangan

ID	SP-UC4
Referensi Kasus Penggunaan	UC-4
Nama	Menampilkan rangkuman keuangan
Tujuan Pengujian	Menguji kemampuan sistem dalam menampilkan rangkuman laporan keuangan
Skenario	Menampilkan rangkuman dalam bentuk grafik.
Kondisi Awal	Pengguna berada pada menu neraca lajur, dan sudah melakukan transaksi jual beli.
Data Uji	Data keuangan
Langkah Pengujian	<ol style="list-style-type: none"> 1. Pengguna memilih menu <i>dashboard</i>. 2. Sistem membaca data keuangan pengguna. 3. Sistem menampilkan hasil rangkuman dalam bentuk grafik.

	4. Pengguna melihat grafik total pendapatan, grafik total pengeluaran, persentase <i>return of asset</i> , persentase <i>return of investment</i> , persentase <i>return of equity</i> .
Hasil Yang Diharapkan	Informasi rangkuman keuangan dapat ditampilkan
Hasil Yang Didapatkan	Informasi rangkuman keuangan dapat ditampilkan
Hasil Pengujian	Berhasil.
Kondisi Akhir	Aktor melihat informasi rangkuman keuangan.

5.2.1.5. Pengujian Menampilkan Prediksi Penjualan

Pengujian ini dilakukan untuk menguji kemampuan sistem dalam menampilkan prediksi penjualan. Rincian skenario pengujian pada kasus penggunaan dapat dilihat pada Tabel 5.5.

Tabel 5.5 Pengujian Menampilkan Prediksi Penjualan

ID	SP-UC5
Referensi Kasus Penggunaan	UC-5
Nama	Menampilkan prediksi penjualan.
Tujuan Pengujian	Menguji kemampuan sistem dalam menampilkan prediksi penjualan.
Skenario	Menampilkan grafik prediksi
Kondisi Awal	Pengguna berada pada menu penjualan dan telah berhasil melakukan transaksi penjualan.
Data Uji	Data transaksi penjualan
Langkah Pengujian	<ol style="list-style-type: none"> 1. Pengguna memilih menu <i>dashboard</i>. 2. Sistem membaca data transaksi penjualan pengguna.

	<ol style="list-style-type: none"> 3. Sistem menampilkan prediksi penjualan dalam 7 hari kedepan. 4. Pengguna melihat grafik prediksi penjualan.
Hasil Yang Diharapkan	Grafik prediksi penjualan dapat ditampilkan
Hasil Yang Didapatkan	Grafik prediksi penjualan dapat ditampilkan
Hasil Pengujian	Berhasil.
Kondisi Akhir	Aktor melihat grafik prediksi penjualan.

5.2.1.6. Pengujian Menampilkan Harga Optimal

Pengujian ini dilakukan untuk menguji kemampuan sistem untuk menampilkan data rekomendasi profit. Rincian skenario pengujian pada kasus penggunaan dapat dilihat pada Tabel 5.6.

Tabel 5.6 Pengujian Menampilkan Harga Optimal

ID	SP-UC6
Referensi Kasus Penggunaan	UC-6
Nama	Menampilkan harga optimal.
Tujuan Pengujian	Menguji kemampuan sistem dalam menampilkan harga optimal.
Skenario	Menampilkan harga optimal
Kondisi Awal	Pengguna berada pada menu <i>dashboard</i> , memilih menu config autostock.
Data Uji	Data masukan pengguna
Langkah Pengujian	<ol style="list-style-type: none"> 1. Pengguna memilih menu config autostock. 2. Pengguna memilih tombol <i>recomended</i> harga. 3. Sistem mengambil data rekomendasi harga berdasarkan nama item.

	4. Sistem mengisi kolom profit dengan rekomendasi harga secara otomatis.
Hasil Yang Diharapkan	Data harga optimal berhasil ditampilkan
Hasil Yang Didapatkan	Data harga optimal berhasil ditampilkan
Hasil Pengujian	Berhasil.
Kondisi Akhir	Aktor melihat kolom harga berisi rekomendasi harga.

5.2.1.7. Pengujian Hasil Pemilihan Supplier

Pengujian ini dilakukan untuk menguji kebenaran hasil dari pemilihan supplier oleh sistem. Pengujian dilakukan dengan membandingkan hasil perhitungan sistem dengan hasil perhitungan melalui manual dengan menggunakan rumus perhitungan yang telah dijelaskan sebelumnya. Data pengujian menggunakan data supplier terhadap barang dagang. Sedangkan barang dagang yang digunakan sebagai contoh pengujian merupakan Dancow Balita 150 gram. Tabel 5.7 merupakan rincian data supplier terhadap Dancow Balita 150 gram.

Tabel 5.7 Data Supplier Terhadap Dancow Balita 150 gram

Id	Nama Suplier	Barang Rusak	Harga	Lama Pengiriman	Bentuk Pengembalian
1	Sinar Abadi	50	12240	1 hari	Uang
2	Lancar Jaya	50	11270	1 hari	Uang
3	Nikmat	50	12400	2 hari	Uang
4	Sakinah	50	11600	3 hari	Uang
5	Maju Terus	50	11600	1 hari	Uang

Dengan menggunakan bobot indikator pada Tabel 3.8, supplier tersebut di hitung nilai masing indikatornya dengan mengalikan terhadap bobot yang sudah ditentukan. Khusus indikator harga dan lama pengiriman dibuat menjadi nilai negatif untuk mempermudah perangkingan di akhir pembobotan.

Sedangkan untuk bentuk pengembalian yang berupa uang diberikan nilai *responsiveness* = 50 dan *flexibility* = 50. Jika bentuk pengembalian berupa barang namun pengiriman barang bersama dengan pembelian selanjutnya, diberikan nilai *responsiveness* = 80 dan nilai *flexibility* = 80. Namun jika dikembalikan dalam bentuk barang dan dikirimkan saat itu juga, diberikan nilai *responsiveness* = 100 dan *flexibility* = 100. Tabel 5.8 dan Tabel 5.9 merupakan hasil total nilai masing-masing supplier dari perkalian nilai indikator dengan bobotnya masing-masing.

Tabel 5.8 Hasil Perkalian Indikator Dengan Bobot

Id	Nama Suplier	Q	C	D	F	R
1	Sinar Abadi	20.81	-4513.22	-0.16	4.93	3.12
2	Lancar Jaya	20.81	-4259.29	-0.16	4.93	3.12
3	Nikmat	20.81	-4555.11	-0.32	4.93	3.12
4	Sakinah	20.81	-4345.68	-0.48	4.93	3.12
5	Maju Terus	20.81	-4345.68	-0.16	4.93	3.12

Tabel 5.9 Peringkat Supplier Berdasarkan Total Nilai

Id	Nama Suplier	Total Nilai	Peringkat
1	Sinar Abadi	-4484.53	4
2	Lancar Jaya	-4230.59	1
3	Nikmat	-4526.58	5
4	Sakinah	-4317.31	3
5	Maju Terus	-4316.98	2

Dari perhitungan manual tersebut, dapat diketahui bahwa supplier paling baik adalah supplier Lancar Jaya. Perhitungan nilai tersebut harus sesuai dengan perhitungan nilai yang dilakukan oleh sistem. Gambar 5.1 merupakan hasil query ranking supplier pada basis data sistem.

supp_id	nama_suplier	total
1	Sinar Abadi	-4505.38
2	Lancar Jaya	-4250.27
3	Nikmat	-4547.62
4	Sakinah	-4337.38
5	Maju Terus	-4337.06

Gambar 5.1 Hasil Perhitungan oleh Sistem

Berdasarkan Gambar 5.1, dapat diketahui bahwa dari perhitungan sistem juga menghasilkan supplier Lancar Jaya yang paling baik. Perbedaan nilai antara sistem dengan perhitungan manual dipengaruhi oleh pembulatan angka yang tidak sama.

5.2.1.8. Pengujian Hasil Perhitungan EOQ dan ROP

Pengujian ini dilakukan untuk menguji kebenaran hasil rekomendasi batas stok dan jumlah pesan oleh sistem. Pengujian dilakukan dengan membandingkan hasil perhitungan sistem dengan hasil perhitungan melalui manual dengan menggunakan rumus perhitungan yang telah dijelaskan sebelumnya. Data pengujian yang digunakan merupakan data rata-rata penjualan suatu produk per hari. Tabel 5.10 merupakan rincian data yang akan digunakan sebagai ujicoba perhitungan.

Tabel 5.10 Data Sampel Rata-rata Penjualan Barang Per Hari

Id Barang	Nama Barang	Rata-rata penjualan per hari	Stok Pengaman
1200031121	Lactogen Pbio 1 180 gr	190.92	1521
1400061121	Nestle Koko Krunch 25 gr	156.50	1619
2200041121	Formula Tp Junior 50 gr Orange	190.31	966

Data penjualan tersebut dapat digunakan untuk menghitung EOQ menggunakan rumus pada Persamaan (3.4) dengan biaya pemesanan setiap pesan sebesar 5000 dan biaya penyimpanan tiap unit barang sebesar 1. Dengan menggunakan persamaan yang ada didapatkan nilai EOQ pada produk pertama dengan proses sebagai berikut.

$$EOQ = \sqrt{\frac{2DC}{h}} = \sqrt{\frac{2 \times 190,92 \times 5000}{1}} = 1381.73$$

Sedangkan untuk produk sampel kedua, EOQ nya adalah sebagai berikut.

$$EOQ = \sqrt{\frac{2DC}{h}} = \sqrt{\frac{2 \times 156,50 \times 5000}{1}} = 1250.99$$

Dan untuk produk sampel ketiga, EOQ nya adalah sebagai berikut.

$$EOQ = \sqrt{\frac{2DC}{h}} = \sqrt{\frac{2 \times 190,31 \times 5000}{1}} = 1379.52$$

Dari ketiga perhitungan manual tersebut, kita bandingkan dengan perhitungan oleh sistem. Gambar 5.2 merupakan hasil perhitungan yang dilakukan oleh sistem dan disimpan pada basis data.

barang_id	eoq
1200031121	1381.73
1400061121	1251
2200041121	1379.53

Gambar 5.2 Hasil Perhitungan EOQ oleh Sistem

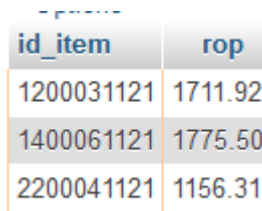
Berdasarkan Gambar 5.2, dapat diketahui bahwa perhitungan manual dan hasil perhitungan sistem menghasilkan angka yang mirip. Sehingga dapat dikatakan bahwa perhitungan sistem sudah sesuai dengan rumus yang digunakan.

Data penjualan pada Tabel 5.9 juga dapat digunakan untuk menghitung ROP yang nantinya digunakan untuk rekomendasi batas stok. Dengan menggunakan rumus pada Persamaan (3.4) dan rata-rata waktu pengiriman (LT) sebesar 1 hari, maka nilai ROP dapat dihitung dengan proses yang disajikan pada Tabel 5.11.

Tabel 5.11 Hasil Perhitungan ROP Manual

Id Barang	Proses Perhitungan	ROP
1200031121	$190.92 * 1 + 1521$	1711.92
1400061121	$156.50 * 1 + 1619$	1775.50
2200041121	$190.31 * 1 + 966$	1156.31

Setelah mendapatkan nilai ROP secara manual, nilai-nilai tersebut dibandingkan dengan hasil perhitungan dari sistem. Gambar 5.3 merupakan hasil perhitungan dari sistem yang didapatkan dari *query* pada basis data.



id_item	rop
1200031121	1711.92
1400061121	1775.50
2200041121	1156.31

Gambar 5.3 Hasil Perhitungan ROP oleh Sistem

Dikarenakan hasil dari perhitungan manual dan hasil dari perhitungan oleh sistem menghasilkan angka yang sesuai, maka dapat disimpulkan bahwa perhitungan sistem sudah sesuai dengan rumus yang digunakan.

5.2.1.9. Pengujian Hasil Perhitungan Harga Optimal (*Optimal Price*)

Pengujian ini dilakukan untuk mengetahui kebenaran dari perhitungan harga optimal oleh sistem. Pengujian dilakukan dengan cara membandingkan hasil perhitungan manual dengan hasil perhitungan sistem, jika hasil keduanya sama maka dapat dikatakan perhitungan sistem sudah sesuai. Data yang digunakan untuk uji coba ini adalah data penjualan dari sampel produk dan data pengeluaran operasional dari perusahaan. Tabel 5.12 merupakan rincian data yang akan digunakan sebagai pengujian.

Tabel 5.12 Rincian Data Objek Pengujian Harga Optimal

<i>Keterangan</i>	<i>Nilai</i>
<i>Id Barang</i>	1500091121
<i>Nama Barang</i>	Supermie Kari Ayam Sedap 70gr
<i>Harga Jual Minimal</i>	Rp 1.788,00
<i>Jumlah jual saat harga minimal</i>	115
<i>Harga Jual Maksimal</i>	Rp 2.074,00
<i>Jumlah jual saat harga maksimal</i>	57
<i>Harga beli (COGS)</i>	Rp 1.430,00
<i>Perkiraan penjualan berikutnya</i>	139
<i>Perkiraan Total Barang Terjual</i>	4854
<i>Total pengeluaran operasional</i>	Rp 28.711.110,00

Dari data sampel tersebut, dapat dihitung harga optimal dengan cara menghitung *Marginal Cost* (MC) terlebih dahulu. Perhitungan MC dapat dilakukan menggunakan Persamaan (3.11). Proses perhitungan MC dapat dijabarkan sebagai berikut.

$$\begin{aligned}
 MC &= \frac{Ep + (COGS \times D)}{D} \\
 &= \frac{\left(\frac{28711110}{4854}\right) + (1430 \times 139)}{139} \\
 &= \frac{5914.94 + 198770}{139} = 1472.553
 \end{aligned}$$

Setelah mendapatkan nilai MC, langkah berikutnya adalah mencari nilai *price elasticity of demand* (PED). Nilai PED dapat ditentukan menggunakan Persamaan (3.12). Proses perhitungan PED dapat dijabarkan sebagai berikut.

$$\begin{aligned} PED &= \left[\frac{(Q^1 - Q^0)}{(Q^1 + Q^0)} \right] \times \left[\frac{(P^1 + P^0)}{(P^1 - P^0)} \right] \\ &= \left[\frac{(57 - 115)}{(57 + 115)} \right] \times \left[\frac{(2074 + 1788)}{(2074 - 1788)} \right] \\ &= \left[\frac{-58}{172} \right] \times \left[\frac{3862}{286} \right] = -4.553 \end{aligned}$$

Nilai negatif pada PED menunjukkan bahwa barang tersebut merupakan barang yang bersifat elastis, artinya perubahan harga akan berpengaruh pada perubahan permintaan. Kemudian setelah mendapatkan nilai PED, langkah terakhir adalah mencari nilai *Opimal Price* (OP) menggunakan Persamaan (3.10). Proses perhitungan OP dapat dijabarkan sebagai berikut.

$$\begin{aligned} OP &= MC \times \left(\frac{PED}{(PED + 1)} \right) \\ &= 1472.553 \times \left(\frac{-4.553}{(-4.553 + 1)} \right) \\ &= 1472.553 \times 1.281 = 1886.34 \end{aligned}$$

Setelah mendapatkan nilai OP secara manual, nilai tersebut dibandingkan dengan hasil perhitungan dari sistem. Gambar 5.4 merupakan hasil perhitungan dari sistem yang didapatkan dari *query* pada basis data.

id_item	ped	op
1500091121	-4.5535	1886.51

Gambar 5.4 Hasil Perhitungan PED dan OP oleh Sistem

Dikarenakan hasil dari perhitungan manual dan hasil dari perhitungan oleh sistem menghasilkan angka yang sesuai, maka dapat disimpulkan bahwa perhitungan sistem sudah sesuai dengan rumus yang digunakan.

5.3. Evaluasi Pengujian

Pada subbab ini membahas hasil evaluasi dari pengujian-pengujian yang telah dilakukan. Evaluasi yang diberikan meliputi evaluasi pengujian kebutuhan fungsional.

5.3.1. Evaluasi Pengujian Fungsionalitas

Rangkuman mengenai hasil pengujian fungsionalitas dapat dilihat pada Tabel 5.13. Berdasarkan data pada tabel tersebut, semua skenario pengujian berhasil dan program berjalan dengan baik. Sehingga bisa ditarik kesimpulan bahwa fungsionalitas dari aplikasi telah dapat bekerja sesuai dengan yang diharapkan.

Tabel 5.13 Hasil Pengujian Fungsionalitas

Kode Pengujian	Skenario Pengujian	Hasil
SP-UC1	Menampilkan rekomendasi <i>supplier</i>	Berhasil
SP-UC2	Menampilkan jumlah <i>re-stock</i> ekonomis	Berhasil
SP-UC3	Menampilkan batas <i>re-stock</i> .	Berhasil
SP-UC4	Menampilkan rangkuman informasi keuangan.	Berhasil
SP-UC5	Menampilkan prediksi penjualan	Berhasil
SP-UC6	Menampilkan harga optimal	Berhasil

[Halaman ini sengaja dikosongkan]

BAB VI

KESIMPULAN DAN SARAN

Pada bab ini akan diberikan kesimpulan yang diperoleh selama pengerjaan Tugas Akhir dan saran mengenai pengembangan yang dapat dilakukan terhadap Tugas Akhir ini di masa yang akan datang.

6.1. Kesimpulan

Dari hasil pengamatan selama proses perancangan, implementasi, dan pengujian perangkat lunak yang dilakukan, dapat diambil kesimpulan sebagai berikut:

1. Pemilihan *supplier* yang optimal, dapat menggunakan metode pembobotan indikator. Penentuan nilai bobot masing-masing indikator dapat menggunakan metode AHP dan *Expert Choice 11*.
2. Penentuan waktu pembelian stok barang yang ekonomis dapat dilakukan dengan menggunakan perhitungan *Re-Order Point*. Sedangkan untuk banyaknya jumlah yang dipesan dapat ditentukan dengan perhitungan *Economic Order Quantity*.
3. Penyajian keadaan keuangan perusahaan dapat menggunakan grafik, sehingga pengguna lebih mudah informasi keadaan keuangan perusahaan mereka.
4. Peramalan penjualan dalam sebuah perusahaan dapat berguna sebagai penentuan kedepan akan permintaan barang oleh pelanggan. Peramalan penjualan dapat diimplementasikan menggunakan *Holt-Winter Multiplication* yang menggunakan *Exponential Smoothing*.
5. Penentuan harga yang optimal dapat dilakukan dengan menggunakan perhitungan yang melibatkan nilai dari *Price Elasticity of Demand* sebagai rasio pengaruh perubahan harga terhadap permintaan dan *Marginal Cost* sebagai biaya dasar untuk melakukan satu unit penjualan.

6.2. Saran

Berikut merupakan beberapa saran untuk pengembangan sistem di masa yang akan datang. Saran-saran ini didasarkan pada hasil perancangan, implementasi dan pengujian yang telah dilakukan:

1. Penambahan fitur untuk mengetahui popularitas barang dagang.
2. Mengganti *database* relasional dengan *database* non relasional (NoSQL). Untuk pengerjaan masa datang disarankan memakai MongoDB.
3. Peningkatan tampilan antarmuka khususnya pada halaman *dashboard*

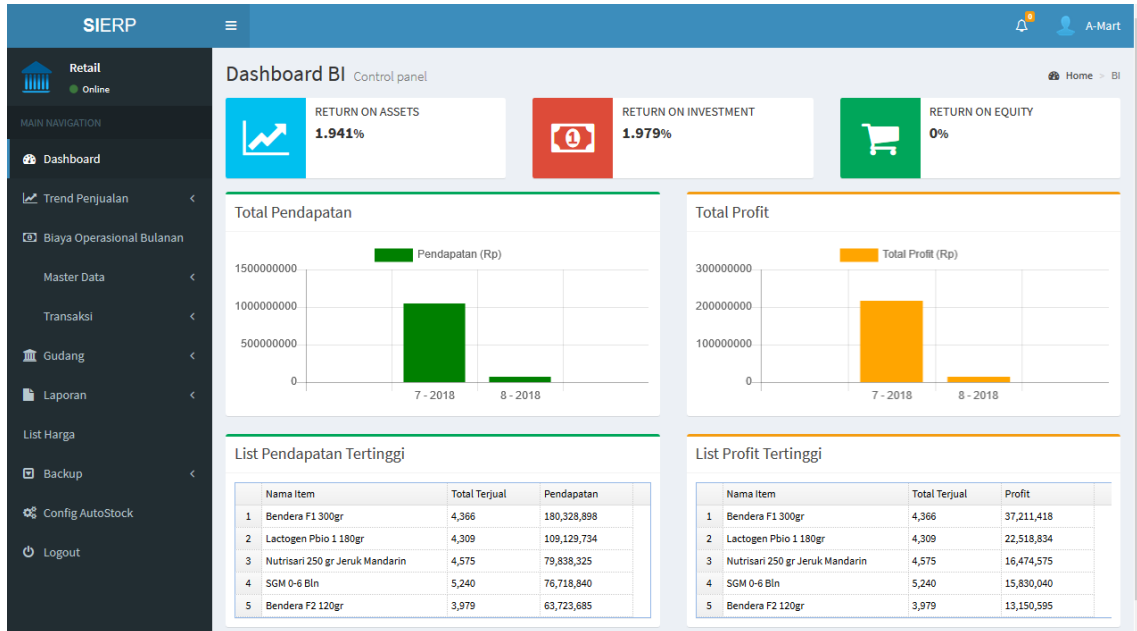
DAFTAR PUSTAKA

- [1] M. A. Adhitama, Rancang Bangun Sistem Enterprise Resource Planning Pada Modul General Ledger Berorientasikan Multitenancy dengan Menggunakan Sistem basis Data Terdistribusi. ITS Surabaya: Jurusan Teknik Informatika-ITS, 2016.
- [2] C. M. Mukwasi dan L. F. Seymour, “Interdependent Enterprise Resource Planning risks in Small and Medium-sized Enterprises in developing countries,” in 2015 IST-Africa Conference, 2015, hal. 1–10.
- [3] Barry Berman dan Joel R. Evans, Retail Management, A Strategic Approach, 8th edition. New Jersey : Prentice Hall, 2001.
- [4] “Accurate User Manual,” Google Docs. [Daring]. Tersedia pada:
https://drive.google.com/file/d/0BxBKmsKfG3YVvKkNKZ116V21ZaTA/view?usp=embed_facebook. [Diakses: 23-Jun-2018].
- [5] S. Negash, “BUSINESS INTELLIGENCE,” Commun. Assoc. Inf. Syst., vol. 13, hal. 177–195, 2004.
- [6] M. I. Nofal dan Z. M. Yusof, “Integration of Business Intelligence and Enterprise Resource Planning within Organizations,” Procedia Technol., vol. 11, hal. 658–665, 2013.
- [7] D. Jacoby, Guide to supply chain management. New York: Bloomberg Press, 2009.
- [8] A. Purwanto, “Penerapan Balanced Scorecard Sebagai Indikator Komprehensif Pengelolaan Sumber Daya Alam – Lingkungan Hidup,” Semin. Sist. Manaj. Pengelolaan Sumber Daya Alam –Lingkungan Hidup, 8 Oktober 2003.
- [9] E. Turban, J. E. Aronson, dan T.-P. Liang, Decision support systems and intelligent systems. New Delhi: Prentice-Hall of India, 2005.

[Halaman ini sengaja dikosongkan]

LAMPIRAN

Gambar



Gambar A.1 Antarmuka Halaman Utama

SIERP ☰ 🔔 👤 A-Mart

Retail
Online

MAIN NAVIGATION

- Dashboard
- Trend Penjualan <
- Biaya Operasional Bulanan
- Master Data <
- Transaksi <
- Gudang <
- Laporan <
- List Harga
- Backup <
- Config AutoStock
- Logout

Pengaturan Produk

Home Master Data Pengaturan produk

Pengaturan Produk

Save Cancel

No	Nama	Harga Beli	Profit (%)	Harga Jual	Iklan	Jumlah Pesan	Supplier	Batas Stock	Recommended Profi	Recommended Jum	Recommended Supp	Recommended Ba
1	Dancow Balita 150 gr	11,270	29%	14,538	3%	746	2	3,303	Menyala	Hidupkan	Menyala	Hidupkan
2	SGM 0-6 Bln	11,620	29%	14,990	3%	666	5	2,597	Menyala	Hidupkan	Menyala	Hidupkan
3	Lactogen Pbio 1 180gr	20,100	29%	25,929	3%	663	1	3,250	Menyala	Hidupkan	Menyala	Hidupkan
4	Bendera F1 120gr	12,390	29%	15,983	3%	710	4	5,267	Menyala	Hidupkan	Menyala	Hidupkan
5	Bendera F2 120gr	12,710	29%	16,396	3%	718	2	3,488	Menyala	Hidupkan	Menyala	Hidupkan
6	Bendera F1 300gr	32,780	29%	42,286	3%	602	1	3,140	Menyala	Hidupkan	Menyala	Hidupkan
7	Nutrisari 250 gr Jeruk Manis	13,180	29%	17,002	3%	708	1	4,594	Menyala	Hidupkan	Menyala	Hidupkan
8	Nutrisari 250 gr Jeruk Mandi	13,850	29%	17,867	3%	648	2	3,106	Menyala	Hidupkan	Menyala	Hidupkan
9	Nestle Cookie Crips 20 gr	3,220	29%	4,154	1%	651	2	3,688	Menyala	Hidupkan	Menyala	Hidupkan
10	Milo Balls 25 gr	3,500	29%	4,515	1%	671	1	2,928	Menyala	Hidupkan	Menyala	Hidupkan
11	Nestle Honey Star 20 gr	3,340	29%	4,309	1%	686	2	4,189	Menyala	Hidupkan	Menyala	Hidupkan
12	Nestle koko Krunch 25 gr	3,320	29%	4,283	1%	592	4	5,013	Menyala	Hidupkan	Menyala	Hidupkan
13	Simba Choco Rillas 20 gr Put	5,550	29%	7,160	2%	704	1	2,918	Menyala	Hidupkan	Menyala	Hidupkan
14	Pop Mie Ayam 56gr	2,990	29%	3,857	1%	655	3	4,010	Menyala	Hidupkan	Menyala	Hidupkan
15	Pop Mie goreng Spc 80gr	3,780	29%	4,876	1%	665	2	3,749	Menyala	Hidupkan	Menyala	Hidupkan
16	Supermie Kari Ayam Sedap	1,430	29%	1,845	0%	582	2	2,584	Menyala	Hidupkan	Menyala	Hidupkan
17	Supermie Grng Sedap	1,470	29%	1,896	0%	743	1	2,560	Menyala	Hidupkan	Menyala	Hidupkan
18	Kit Kat 2f 17gr	2,700	29%	3,483	1%	654	2	2,890	Menyala	Hidupkan	Menyala	Hidupkan

Gambar A.2 Antarmuka Halaman Config Autostock

BIODATA PENULIS



Penulis, **Renanda Agustiantoro**, lahir di Pati, 5 Agustus 1995. Penulis menempuh pendidikan sekolah menengah atas di SMA Negeri 1 Pati. Selanjutnya penulis melanjutkan pendidikan sarjana di Departemen Teknik Informatika, Fakultas Teknologi dan Informasi, Institut Teknologi Sepuluh Nopember Surabaya.

Memiliki pengalaman organisasi sebagai Staf Departemen Pemberdayaan Sumber Daya Manusa HMTTC 2014-2015, Panitia Publikasi dan Dokumentasi YES Summit 2015. Dalam menyelesaikan pendidikan S1, penulis mengambil bidang minat Manajemen Informasi (MI) dan memiliki ketertarikan di bidang *Web Application Development*. Penulis dapat dihubungi melalui *e-mail*: rectarenanda@gmail.com