**RANCANG BANGUN SISTEM INFORMASI PERSEDIAAN DAN PERAMALAN OBAT DI APOTEK DENGAN METODE EOQ (*ECONOMIC ORDER QUANTITY*) DAN *EXPONENTIAL SMOOTHING***

# TUGAS AKHIR



**Bagus Dhanist Rananta**

**1102001013**

**PROGRAM STUDI INFORMATIKA**

**FAKULTAS TEKNIK DAN ILMU KOMPUTER**

**UNIVERSITAS BAKRIE**

**JAKARTA**

**2016**

# HALAMAN PERNYATAAN ORISINALITAS

**Tugas Akhir ini adalah hasil karya saya sendiri,**

**dan semua sumber baik yang dikutip maupun dirujuk**

**telah saya nyatakan dengan benar.**

**Nama : Bagus Dhanist Rananta**

**NIM : 1102001013**

**Tanda Tangan :**

**Tanggal :**

# HALAMAN PENGESAHAN

Tugas Akhir ini diajukan oleh:

Nama : Bagus Dhanist Rananta

NIM : 1102001037

Program Studi : Informatika

Fakultas : Fakultas Teknik dan Ilmu Komputer

Judul Skripsi : Rancang Bangun Sistem Informasi Persediaan dan Peramalan Obat di Apotek Dengan Metode EOQ (*Economic Order* Quantity) dan *Exponential Smoothing*

**Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai persyaratan yang diperlukan untuk memperoleh gelar Sarjana Komputer (S.Kom) pada Program Studi Informatika Fakultas Teknik dan Ilmu Komputer Universitas Bakrie.**

**DEWAN PEMBIMBING**

Pembimbing : Yusuf Lestanto, ST, M.Sc ( )

Penguji I : Berkah I Susanto ( )

Penguji II : Hoga Saragih, S.T, M.T, Dr. ( )

Ditetapkan di : Jakarta

Tanggal : 19 Agustus 2016

# UNGKAPAN TERIMA KASIH

Segala puji syukur penulis panjatkan atas kehadirat Allah SWT, yang atas limpahan rahmat, taufik, dan hidayah-Nya, penulis mampu menyelesaikan Tugas Akhir ini. Penulisan Tugas Akhir ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gelar Sarjana Komputer Program Studi Informatika Fakultas Teknik dan Ilmu Komputer Universitas Bakrie. Terselesaikanya Tugas Akhir ini tentu saja tidak lepas dari dorongan dan uluran tangan beberapa pihak. Oleh karena itu, tak salah kiranya jikalau penulis mengungkapkan rasa terima kasih dan penghargaan kepada:

1. Bapak Yusuf Lestanto, ST, M.Sc, MBA, selaku dosen pembimbing, yang dengan sabar telah meluangkan waktu untuk membimbing dan mengarahkan penulis.
2. Seluruh dosen program studi Informatika Universitas Bakrie yang telah membimbing selama masa perkuliahan.
3. Orang tua dan keluarga saya yang telah memberikan bantuan dengan dukungan material dan moral.
4. Apotek Nadia Farma yang telah banyak membantu dalam usaha memperoleh data yang dibutuhkan dalam penulisan.
5. Teman-teman Informatika dan semua teman yang telah membantu baik dengan dukungan moral, maupun doanya untuk menyelesaikan skripsi ini.

Akhir kata, penulis berharap Allah SWT berkenan membalas segala kebaikan semua pihak yang telah membantu. Semoga Tugas Akhir ini dapat bermanfaat bagi yang membaca dan dapat dikembangkan lagi lebih jauh di masa depan.

Jakarta, 19 Agustus 2016

Bagus Dhanist Rananta

# HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI

Sebagai sivitas akademik Universitas Bakrie, saya yang bertanda tangan di bawah ini:

Nama : Bagus Dhanist Rananta

NIM : 1102001013

Program Studi : Informatika

Fakultas : Teknik Dan Ilmu Komputer

Jenis Tugas Akhir : Rancang Bangun

demi Pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Bakrie **Hak Bebas Royalti Noneksklusif (*Non-exclusive Royalty-free Right*)** atas karya ilmiah saya yang berjudul:

"**Rancang Bangun Sistem Informasi Persediaan dan Peramalan Obat di Apotek Dengan Metode EOQ (*Economic Order* Quantity) dan *Exponential Smoothing***"

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Universitas Bakrie berhak menyimpan, mengalihmedia/*form*atkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan mempublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta untuk kepentingan akademis.

Demikian pernyataan ini saya buat sebenarnya.

Dibuat di : Jakarta

Pada tanggal : 19 Agustus 2016

Yang Menyatakan

(Bagus Dhanist Rananta)

**RANCANG BANGUN SISTEM INFORMASI PERSEDIAAN DAN PERAMALAN OBAT DI APOTEK DENGAN METODE EOQ (*ECONOMIC ORDER QUANTITY*) DAN *EXPONENTIAL SMOOTHING***

Bagus Dhanist Rananta

# ABSTRAK

Pada penelitian ini dilakukan rancang bangun sistem informasi persediaan dan peramalan obat di apotek dengan metode EOQ (*Economic Order Quantity)* dan *Exponential Smoothing*. Persediaan merupakan salah satu faktor yang memegang pernanan penting di dalam sebuah badan usaha. Pengadaan persediaan yang masih manual dapat menghambat target penjualan, karena tidak menutup kemungkinan suatu saat ada kesalahan dalam perhitungan persediaan yang mengakibatkan penjualan barang tersebut terhenti karena kurangnya persediaan barang bahkan memungkinkan barang habis. Saat ini, sistem yang di terapkan di Apotek Nadia Farma masih manual dengan melakukan pencatatan di buku jurnal. Berdasarkan hal tersebut sistem informasi persediaan dan peramalan dibuat untuk membantu pihak apotek dalam mengontrol persediaan dan jumlah pembelian barang. Sistem ini dibuat dengan menggunakan metode *Exponential Smoothing* untuk melakukan peramalan, EOQ (*Economic Order Quantity*) untuk menghitung pemesanan barang yang paling ekonomis, dan *Iterative Waterfall* sebagai metode pengembangan sistem*.*Sistem ini telah diuji oleh pihak Apotek Nadia Farma. Hasil dari pengujian menunjukkan bahwa sistem telah berjalan sesuai dengan keinginan dan metode EOQ maupun *Exponential Smoothing* berjalan sesuai dengan fungsinya.

Kata Kunci: Sistem Informasi, *Exponential Smoothing,* EOQ(*Economic Order Quantity*), *Iterative Waterfall*

**DESIGN AND DEVELOPMENT INVENTORY AND FORECASTING INFORMATION SYSTEM IN DRUGSTORE USING EOQ (ECONOMIC ORDER QUANTITY) AND EXPONENTIAL SMOOTHING METHOD**

Bagus Dhanist Rananta

# ABSTRACT

*This research study is designing and developing inventory and forecasting information system in drugstore using EOQ (Economic Order Quantity) and Exponential Smoothing method. Inventory is one factor that plays a crucial role in a business. Procurement of supplies that are still manual can hamper sales target, because it is possible when there is a mistake in the calculation of inventory that resulted in the sale of goods are suspended due to lack of inventory. Currently, the system that applied in the Nadia Farma Drugstore still recording manually in ledgers. Based on this case, inventory and forecasting information system created to assist pharmacies in controlling inventories and purchases of goods. The system is made using Exponential Smoothing method for forecasting, EOQ (Economic Order Quantity) to calculate the most economical ordering of goods, and Iterative Waterfall as system development model. This system have been tested by the Nadia Farma Drugstore. The results of the testing indicate that the system has been running in accordance with the requirement and EOQ method and Exponential Smoothing run in accordance with its function.*

*Keywords: Information Systems, Exponential Smoothing, EOQ (Economic Order Quantity), Iterative Waterfall*

# DAFTAR ISI

[HALAMAN JUDUL i](#_Toc459341512)

[HALAMAN PERNYATAAN ORISINALITAS ii](#_Toc459341513)

[HALAMAN PENGESAHAN iii](#_Toc459341514)

[UNGKAPAN TERIMA KASIH iv](#_Toc459341515)

[HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI v](#_Toc459341516)

[ABSTRAK vi](#_Toc459341517)

[ABSTRACT vii](#_Toc459341518)

[DAFTAR ISI viii](#_Toc459341519)

[DAFTAR GAMBAR xi](#_Toc459341520)

[DAFTAR TABEL xii](#_Toc459341521)

[DAFTAR RUMUS xiii](#_Toc459341522)

[DAFTAR SINGKATAN xiv](#_Toc459341523)

[BAB I 1](#_Toc459341524)

[PENDAHULUAN 1](#_Toc459341525)

[1.1 Latar Belakang 1](#_Toc459341526)

[1.2 Perumusan Masalah 4](#_Toc459341527)

[1.3 Pembatasan Masalah 4](#_Toc459341528)

[1.4 Tujuan Penelitian 4](#_Toc459341529)

[1.5 Manfaat Penelitian 5](#_Toc459341530)

[BAB II 6](#_Toc459341531)

[TINJAUAN PUSTAKA 6](#_Toc459341532)

[2.1 Studi Literatur 6](#_Toc459341533)

[2.2 Apotek 10](#_Toc459341534)

[2.3 Persediaan 11](#_Toc459341535)

[2.3.1 Economic Order Quantity (EOQ) 13](#_Toc459341536)

[2.3.2 Just In Time (JIT) 16](#_Toc459341537)

[2.4Peramalan 18](#_Toc459341538)

[2.4.1 Exponential Smoothing 21](#_Toc459341539)

[2.4.2 Moving Average 22](#_Toc459341540)

[2.5 *SDLC (Software Development Life Cycle)* 24](#_Toc459341541)

[2.6 Konsep Dasar Sistem Informasi 28](#_Toc459341542)

[2.7 Metode Pengujian Sistem 29](#_Toc459341543)

[2.8 *Object Oriented Programming (OOP)* 31](#_Toc459341544)

[2.9 Bahasa Pemrograman 32](#_Toc459341545)

[BAB III 35](#_Toc459341546)

[METODOLOGI PENELITIAN 35](#_Toc459341547)

[3. 1 Alat Penelitian 35](#_Toc459341548)

[3. 2 Metode Pengumpulan Data 36](#_Toc459341549)

[3. 3 Pengolahan Data 37](#_Toc459341550)

[3. 4 Metode Pengembangan Sistem 38](#_Toc459341551)

[3.4.1 Analisis dan Definisi Persyaratan 38](#_Toc459341552)

[3.4.2 Perancangan Sistem dan Perangkat Lunak 39](#_Toc459341553)

[3.4.3 Implementasi dan Pengujian Unit 40](#_Toc459341554)

[3.4.4 Integrasi dan Pengujian sistem 40](#_Toc459341555)

[3.4.5 *Operation and Maintenance* 41](#_Toc459341556)

[3.5 Jadwal Penelitian 42](#_Toc459341557)

[BAB IV 43](#_Toc459341558)

[PEMBAHASAN 43](#_Toc459341559)

[4.1 Pengumpulan Informasi 43](#_Toc459341560)

[4.2 Analisis Kebutuhan Sistem 44](#_Toc459341561)

[4.2.1 Desain *Database* 44](#_Toc459341562)

[4.2.3 *Use case* Diagram 53](#_Toc459341563)

[4.2.4 *Activity diagram* 60](#_Toc459341564)

[4.2.5 Deployment Diagram 67](#_Toc459341565)

[4.2.6 *Class diagram* 68](#_Toc459341566)

[4.2.7 Sequence Diagram 69](#_Toc459341567)

[4.3 *User* Interface Sistem 77](#_Toc459341568)

[4.4 Pengujian Sistem 86](#_Toc459341569)

[4.4.1 Metode Exponential Smoothing dan EOQ 86](#_Toc459341570)

[BAB V 89](#_Toc459341571)

[PENUTUP 89](#_Toc459341572)

[5.1 Kesimpulan 89](#_Toc459341573)

[5.2 Saran 90](#_Toc459341574)

[DAFTAR PUSTAKA 91](#_Toc459341575)

# DAFTAR GAMBAR

[Gambar 2.1 Software Development Life Cycle (SDLC) (Maheswari & Jain, 2012) 24](#_Toc459340269)

[Gambar 2.2 *Waterfall Model* (Maheswari & Jain, 2012) 25](#_Toc459340270)

[Gambar 2.3 *Iterative Waterfall* model (Maheswari & Jain, 2012) 26](#_Toc459340271)

[Gambar 2.4 *Prototype* *Model* (Maheswari & Jain, 2012) 26](#_Toc459340272)

[Gambar 2. 5 *Spiral Model* (Maheswari & Jain, 2012) 27](#_Toc459340273)

[Gambar 2. 6 Statistik Penggunaan *Tools* untuk Pengembangan Web 33](file:///D:\Skripsi\Skripsi%20Bagus%20Dhanist%20(2).docx#_Toc459340274)

[Gambar 4.1 *Conceptual database design* dari sistem 47](#_Toc459339466)

[Gambar 4.2 *Logical database design* dari sistem 51](#_Toc459339467)

[Gambar 4.3 Diagram *entity relationship diagram* pada sistem 52](#_Toc459339468)

[Gambar 4.4 *Use case diagram* dari sistem 53](#_Toc459339469)

[Gambar 4.5 *Activity diagram* *Login* 60](#_Toc459339470)

[Gambar 4.6 *Activity diagram* *Logout* 61](#_Toc459339471)

[Gambar 4.7 *Activity diagram* *Create* 62](#_Toc459339472)

[Gambar 4.8 *Activity diagram* *Read* 63](#_Toc459339473)

[Gambar 4.9 *Activity diagram* *Update* 64](#_Toc459339474)

[Gambar 4.10 *Activity diagram* *Delete* 65](#_Toc459339475)

[Gambar 4.11 *Activity diagram* EOQ 66](#_Toc459339476)

[Gambar 4.12 *Deployment Diagram* 67](#_Toc459339477)

[Gambar 4.13 *Class diagram* dari sistem 68](#_Toc459339478)

[Gambar 4.14 Sequence Diagram *Login* 69](#_Toc459339479)

[Gambar 4.15 Sequence Diagram Logut 70](#_Toc459339480)

[Gambar 4.16 Sequence Diagram *Create* 71](#_Toc459339481)

# DAFTAR TABEL

[Tabel 2.1 Penelitian terdahulu 8](#_Toc459339541)

[Tabel 2.2 Perbedaan *Just In Time* dengan Sistem Konvensional (Mayora, 2015) 17](#_Toc459339542)

[Tabel 2.3 Perbedaan *Exponential Smoothing* dengan *Moving Average* 23](#_Toc459339543)

[Tabel 2.4 Perbandingan dari empat metode SDLC (Maheswari & Jain, 2012) 27](#_Toc459339544)

[Tabel 2.5 Perbandingan tiga teknik pengujian (Khan & Khan, 2012) 30](#_Toc459339545)

[Tabel 2.6 Perbedaan antara PHP dan ASP.NET (Nagilla, 2013) 34](#_Toc459339546)

[Tabel 3.1 Jadwal Penelitian 42](#_Toc459339566)

[Tabel 4.1 Daftar *entity* dan keterangannya 45](#_Toc459339521)

[Tabel 4.2 Hubungan antar *entity* 46](#_Toc459339522)

[Tabel 4.3 Daftar attribut dari masing-masing *entity* 47](#_Toc459339523)

[Tabel 4.4 *Use case Scenario Login* 53](#_Toc459339524)

[Tabel 4.5 *Use case* *Scenario* *Logout* 54](#_Toc459339525)

[Tabel 4.6 *Use case* *Scenario* Lihat Menu Utama 54](#_Toc459339526)

[Tabel 4.7 *Use case* *Scenario* Kelola Obat 55](#_Toc459339527)

[Tabel 4.8 *Use case* *Scenario* Kelola *User* 56](#_Toc459339528)

[Tabel 4 9 *Use case* *Scenario* Kelola *Supplier* 56](#_Toc459339529)

[Tabel 4.10 *Use case* *Scenario* Kelola Penjualan 57](#_Toc459339530)

[Tabel 4.11 *Use case* *Scenario* Kelola Pembelian 58](#_Toc459339531)

[Tabel 4.12 *Use case* *Scenario* Melakukan Peramalan 58](#_Toc459339532)

[Tabel 4.13 *Use case* *Scenario* Menghitung EOQ 59](#_Toc459339533)

[Tabel 4.14 Tabel Model, Controller, dan Fungsi *Create* 71](#_Toc459339534)

[Tabel 4.15 Model, Controller, dan Fungsi dari *Update* 72](#_Toc459339535)

[Tabel 4.16 Model, Controller, dan Fungsi *Read* 74](#_Toc459339536)

[Tabel 4.17 Model, Controller, dan Fungsi *Delete* 75](#_Toc459339537)

# DAFTAR RUMUS

Rumus 2.1 *Economic Order Quantity* (EOQ) 14

Rumus 2.2 Biaya Penyimpanan 14

Rumus 2.3 *Reorder Point* (RoP) 14

Rumus 2.4 *Exponential Smoothing* 21

Rumus 2.5 *Mean Square Error* (MSE) 22

Rumus 2.6 *Mean Absolute Percentage Error* (MAPE) 22

Rumus 2.7 *Moving Average* 23

# DAFTAR SINGKATAN

EOQ *Economic Order Quantity*

JIT *Just In Time*

RoP *Reorder Point*

MSE *Mean Squared Error*

MAPE *Mean Absolute Percentage Error*

SDLC *Software Development Life Cycle*

OOP *Object Oriented Programming*

SQL *Structured Query Language*

CI *CodeIgniter*

# BAB I

# PENDAHULUAN

## Latar Belakang

Proses bisnis yang kompleks dan dengan persaingan bisnis yang semakin ketat membuat pihak apotek harus memiliki strategi yang tepat agar dapat bersaing dengan apotek lainnya. Untuk memenangkan persaingan tersebut, maka suatu badan usaha harus melakukan perubahan-perubahan yang mendorong aktivitas usaha untuk melakukan pengawasan terhadap persediaan dan efisiensi biaya, sehingga badan usaha mampu menekan biaya untuk mendukung keuntungan yang semakin menurun seperti dengan melakukan efektifitas persediaan barang dalam suatu perusahaan (Gessong, 2001).

Obat sebagai salah satu jenis barang yang dijual di apotek, adalah bahan untuk mengurangi, menghilangkan penyakit, atau menyembuhkan seseorang dari penyakit (KBBI). Pada Peraturan Menteri Kesehatan (Permenkes) No. 917/Menkes/Per/X/1993, pengertian obat adalah sediaan atau padu-paduan yang siap digunakan untuk mempengaruhi atau menyelidiki secara fisiologi atau keadaan patologi dalam rangka penetapan diagnosa, pencegahan, penyembuhan, pemulihan, peningkatan kesehatan dan kontrasepsi. Pada Permenkes yang sama juga diatur wajib daftar obat. Sehingga golongan obat adalah penggolongan yang dimaksudkan untuk peningkatan keamanan dan ketetapan penggunaan serta pengamanan distribusi yang terdiri dari obat bebas, obat bebas terbatas, obat wajib apotik, obat keras, psikotropika dan narkotika (Mulyani, 2011).

Permasalahan utama yang dihadapi oleh kebanyakan apotek adalah sistem pencatatan yang masih manual, sehingga banyak informasi yang salah dan hilang. Dari permasalahan tersebut, akan timbul bermacam macam masalah seperti stok obat yang sudah habis dan pelayanan yang lambat dan pada akhirnya akan berimbas pada ketidakpuasan konsumen. Banyak terjadi ketimpangan di dalam sebuah organisasi atau badan usaha dimana tidak adanya keselarasan antara sasaran organisasi atau badan usaha dengan strategi sistem informasi dan teknologi informasi (Yogeswara, et al., 2013). Oleh karena itu, dibutuhkan strategi sistem informasi dan teknologi informasi berbasis komputer di apotek yang dapat membantu pihak apotek dalam melakukan pekerjaannya.

Wawancara dan observasi dilakukan pada tanggal 23 September 2015, pada pukul 19.00 sampai dengan selesai bertempat di Apotek Nadia Farma yang berlokasi di Jl. Ratu Dibalau, Tanjung Senang, Bandar Lampung. Penulis mewawancarai Ibu Lismawati berusia 45 tahun selaku pemilik apotek sebagai informan, hasil dari wawancara tertera pada lampiran 2. Berdasarkan hasil wawancara, diketahui bahwa pada saat ini apotek masih menggunakan sistem konvensional yaitu dengan melakukan pencatatan pada buku. Segala aktivitas baik penjualan, penyimpanan, maupun pembelian dicatat di dalam sebuah buku. Pembelian tidak dilakukan secara besar-besaran dikarenakan oleh sumber daya yang tidak mencukupi, seperti tempat penyimpanan yang kurang, dan pegawai yang sedikit serta sistem belum memadai.

Dengan menggunakan sistem yang konvensional dengan pencatatan pada sebuah buku dan dengan mengandalkan tenaga manusia, maka akan timbul beberapa masalah yang harus dihadapi oleh pihak apotek. Permasalahan yang umum yang timbul apabila menggunakan sistem konvensional adalah pencatatan manual yang sangat menyulitkan pegawai untuk melakukan pencatatan pada semua aktivitas yang terjadi di dalam apotek, baik pada saat pembelian, penyimpanan, maupun penjualan obat. Selain masalah tersebut, masalah lain yang timbul akibat sistem ini adalah banyak kesalahan dalam melakukan pencatatan, seperti adanya data obat yang belum atau lupa tercatat, bahkan tidak menutup kemungkinan terdapat beberapa data yang hilang atau dicuri. Masalah lainnya antara lain sulit untuk melihat ketersediaan obat, tidak dapat diketahui apabila terdapat obat yang persediaannya sudah habis (*stock out*), jadwal pembelian obat yang tidak menentu sehingga lebih mengeluarkan biaya, dan *stock opname*  membutuhkan waktu yang lama dan hanya dilakukan sekali dalam setahun.

Berdasarkan hasil wawancara, permasalahan persediaan menjadi perhatian utama bagi pengelola di Apotek Nadia Farma. Oleh karena itu, dibutuhkan manajemen persediaan yang baik agar pengelolaan persediaan pada apotek menjadi lebih mudah. Manajemen persediaan yang dimaksud adalah dengan cara mencari jumlah persediaan yang optimal sehingga setiap biaya menyangkut persediaan dapat di minimalisir. Selain itu, peramalan terhadap penjualan barang pada peirode berikutnya dapat dijadikan patokan bagi perusahaan untuk mempersiapkan persediaan yang dibutuhkan sehingga tidak mengganggu operasional perusahaan. Tanggung jawab utama dari manajemen persediaan adalah pengelolaan barang dan sistem barang di seluruh pengadaan dan rantai pasokan yang telah diketahui dari pembelian barang sampai pada titik konsumsi (Kyei, et al., 2008). Oleh karena itu, dibutuhkan sistem manajemen dan peramalan terhadap persediaan yang terdapat pada apotek agar kegiatan operasional apotek berjalan dengan lancar karena persediaan memegang peranan yang penting pada apotek.

Tujuan dari penelitian ini adalah membuat sebuah sistem informasi yang dapat digunakan oleh Apotek Nadia Farma dalam pengelolaan persediaan obat yang terdapat pada apotek tersebut dengan menggunakan metode EOQ (*Economic Order Quantity*) dan *exponential smoothing*. EOQ merupakan volume atau jumlah pembelian yang paling ekonomis untuk dilaksanakan pada setiap kali pembelian. Untuk memenuhi kebutuhan itu maka dapat diperhitungkan pemenuhan kebutuhan (pembelian) yang paling ekonomis yaitu sejumlah barang yang akan diperoleh dengan pembelian dengan menggunakan biaya yang minimal (Fatona, 2013). Peramalan akan memperkirakan jumlah penjualan obat dengan menggunakan deret waktu (*time series*) berdasarkan data pada periode sebelumnya. Pola data tersebut mendekati bentuk pola horizontal sehingga peramalan yang sesuai dengan pola data tersebut adalah dengan menggunakan metode *exponential smoothing* (Ardhi, et al., 2013). Dengan adanya sistem informasi ini diharapkan pihak apotek dapat melakukan kegiatan apotek dengan lebih mudah, meminimalisir biaya, dan memberikan pelayanan yang lebih baik kepada pelanggan apotek.

## Perumusan Masalah

Berdasarkan latar belakang permasalahan yang telah dijelaskan sebelumnya, maka permasalahan yang akan dibahas pada penelitian ini adalah:

1. Bagaimana mengetahui informasi mengenai obat dan data persediaan obat pada apotek?
2. Bagaimana cara meramalkan jumlah penjualan obat untuk periode berikutnya?
3. Bagaimana cara untuk membantu pihak apotek dalam menentukan jumlah dan waktu pembelian obat yang paling ekonomis?

## Pembatasan Masalah

Batasan masalah dalam pembuatan sistem ini antara lain:

1. Permasalahan dibatasi pada perancangan sistem informasi persediaan obat pada apotek dengan menggunakan metode EOQ (*Economic Order Quantity*)
2. Sistem tidak menyimpan data seluruh pegawai.
3. Pengujian aplikasi dilakukan oleh staff apotek dengan menggunakan metode *black box testing.*
4. Peramalan jumlah obat hanya dapat diketahui untuk periode berikutnya, tidak dapat dilakukan untuk meramalkan dua periode berikutnya atau lebih.

## Tujuan Penelitian

1. Merancang dan membangun sistem informasi persediaan obat pada apotek.
2. Merancang dan membangun sistem informasi persediaan obat pada apotek dengan menggunakan metode *exponential smoothing* untuk melakukan peramalan obat*.*
3. Merancang dan membangun sistem informasi persediaan obat pada apotek dengan metode EOQ (*Economic Order Quantity*) untuk menghitung jumlah pembelian optimal.

## Manfaat Penelitian

Manfaat dari penelitian ini adalah:

1. Memudahkan pihak apotek dalam mengelola obat.
2. Membantu pihak apotek dalam mengatur persediaan obat yang terdapat pada apotek, baik pada saat pembelian, pemeliharaan, maupun penjualan.
3. Mampu meramalkan jumlah penjualan obat pada periode berikutnya pada apotek.

# BAB II

# TINJAUAN PUSTAKA

## 2.1 Studi Literatur

Sebelum melakukan penelitian ini, penulis melakukan sebuah studi literatur dari penelitian yang telah dilakukan terkait dengan topik yang akan dibahas mengenai sistem informasi persediaan obat di apotek dengan menggunakan metode EOQ (*Economic Order Quantity)* dan *exponential smoothing*. Penelitian sebelumnya akan penulis jadikan referensi dalam melakukan penelitian ini. Setiap penelitian memiliki banyak perbedaan, baik dari objek, metode, alat penelitian, maupun hasilnya. Oleh karena itu, diperlukan pembelajaran terhadap penelitian terkait untuk mendapatkan metode dan alat penelitian yang tepat agar hasil akhir dari penelitian sesuai dengan harapan.

Penelitian terdahulu yang dijadikan studi literatur merupakan penelitian mengenai sistem informasi persediaan obat di apotek dengan metode EOQ dan *exponential smoothing*, karena sistem informasi persediaan obat di apotek, metode EOQ, dan *exponential smoothing* merupakan fokus utama yang akan dijadikan bahan penelitian. Oleh karena itu, dibutuhkan beberapa studi literatur mengenai sistem informasi persediaan obat di apotek, metode EOQ, dan *exponential smoothing*. Berikut ini akan diuraikan tiga penelitian terkait sistem informasi persediaan obat di apotek, metode EOQ, dan *exponential smoothing* yang digunakan sebagai referensi di dalam penelitian ini.

Penelitian yang pertama adalah penelitian dari Christian dan Tiur (2014) yang berjudul “*Aplikasi Apotik X Dengan Penerapan Metode Economic Order Quantity”.* Penelitian ini menghasilkan sebuah aplikasi untuk apotek yang dapat mengolah obat dengan benar dengan menerapkan sebuah metode untuk menentukan jumlah persediaan obat dengan meminimalkan biaya total persediaan di gudang obat. Permasalahan pada penelitian ini adalah pembelian obat yang terlalu banyak sedangkan obat di dalam apotek masih banyak tersisa. Dengan adanya aplikasi ini, diharapkan dapat membantu pihak apotek dalam mengelola obat dan mendapat keuntungan dari penjualan obat yang ada.

Penelitian kedua adalah penelitian dari Afrian (2014) yang berjudul “*Penerapan Metode Economic Order Quantity Probabilistik Menggunakan Model (q,r) Pada Persediaan Obat Anti Mefinal 500Mg*”. Penelitian ini membahas tentang penerapan metode EOQ probabilistic menggunakan model (*q,r*) pada pengendalian dan persediaan obat antinyeri. Penelitian ini dilakukan di Apotek Griya Medika Malang, yang beralamat di Jalan Soekarno Hatta Malang, Provinsi Jawa Timur. Data yang di peroleh pada penelitian ini diasumsikan normal dengan tujuan untuk menentukan *reorder point* dan EOQ untuk meminimumkan biaya total dengan menggunakan model (*q,r*) pada obat antinyeri di Apotek Griya Medika Malang pada Bulan Januari sampai Desember 2013. Metode EOQ memberikan hasil bahwa total biaya minumum secara teori lebih baik dari hasil simulasi yang dilakukan pada apotek.

Penelitian ketiga adalah penelitian dari Immanuel (2013) yang berjudul “*Perancangan Sistem Pengontrolan Stok Barang Pada Blesscom Komputer Dengan Metode Economic Order Quantity*”. Permasalahan yang terjadi pada penelitian ini adalah pengontrolan barang masih menggunakan cara yang manual dimana pengontrolan tersebut masih dibuat di dalam sebuah buku besar yang di dalamnya dituliskan data-data mengenai penjualan, pembelian dan pengontrolan stok barang. Tujuan dari penelitian ini adalah dengan membuat sebuah rancangan pengontrolan stok barang pada Bless Computer dengan menggunakan metode *Economic Order Quantity* (EOQ). Penelitian ini menggunakan bahasa pemrograman PHP dengan menggunakan database *MySql.*

Penelitian keempat adalah penelitian dari Sayed (2015) yang berjudul “*Peramalan Penjualan Obat Menggunakan Metode Exponential Smoothing Pada Toko Obat Bintang Geurogok*”. Permasalahan yang terjadi pada penelitian ini adalah diperlukannya peramalan penjualan obat untuk meningkatkan keuntungan dan menghindari terjadinya kelebihan maupun kekurangan obat. Penelitian ini menggunakan metode *Exponential Smoothing* dengan menggunakan nilai alpha (α) dari 0,1 sampai dengan 0,9. Nilai parameter α yang sesuai akan memberikan ramalan yang optimal dengan nilai kesalahan paling sedikit dengan pencarian menggunakan metode *Mean Square Error* (MSE). Uji coba sampel menggunakan salah satu merek obat yaitu dengan menggunakan data penjualan obat *Ambeven* dari bulan Januari 2013 sampai dengan Maret 2015.

Tabel 2.1 Penelitian terdahulu

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Judul** | **Pengarang** | **Tahun** | **Permasalahan** | **Hasil** |
| 1. | Aplikasi Apotik X Dengan Penerapan Metode *Economic Order Quantity* | Christian dan Tiur | 2014 | Pembelian obat yang terlalu banyak sedangkan obat di dalam apotek masih banyak tersisa. Dengan adanya aplikasi ini, diharapkan dapat membantu pihak apotek dalam mengelola obat dan mendapat keuntungan dari penjualan obat yang ada. | Aplikasi untuk apotek yang dapat mengolah obat dengan benar dengan menerapkan sebuah metode untuk menentukan jumlah persediaan obat dengan meminimalkan biaya total persediaan di gudang obat. |
| 2. | Penerapan Metode *Economic Order Quantity Probabilistik* Menggunakan Model (q,r) Pada Persediaan Obat *Anti Mefinal* | Afrian Satria | 2014 | Penentuan jumlah barang yang akan dipesan serta waktu untuk pemesanan yang kurang efektif, sehingga terjadi penumpukan barang di gudang penyimpanan. | Metode EOQ memberikan hasil bahwa total biaya minumum secara teori lebih baik dari hasil simulasi yang dilakukan pada apotek. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Judul** | **Pengarang** | **Tahun** | **Permasalahan** | **Hasil** |
| 3. | Perancangan Sistem Pengontrolan Stok Barang PadaBlesscom Komputer Dengan Metode *Economic Order Quantity* | Immanuel Ginting | 2013 | Pengontrolan barang masih menggunakan cara yang manual dimana pengontrolan tersebut masih dibuat di dalam sebuah buku besar yang di dalamnya dituliskan data-data mengenai penjualan, pembelian dan pengontrolan stok barang. | Data-data akan dimasukkan dan disimpan didalam *database* dan dikoneksikan ke pemograman *visual Bassic 6.0* sehingga dalam pengendalian dan pengontrolan stok barang lebih mudah ,cepat dan juga efesien. |
| 4. | Peramalan Penjualan Obat Menggunakan Metode *Exponential Smoothing* Pada Toko Obat Bintang Geurogok | Sayed Facrurrazi | 2015 | Permasalahan yang terjadi pada penelitian ini adalah diperlukannya peramalan penjualan obat untuk meningkatkan keuntungan dan menghindari terjadinya kelebihan maupun kekurangan obat. | Uji coba sampel menggunakan salah satu merek obat yaitu dengan menggunakan data penjualan obat *Ambeven* dari bulan Januari 2013 sampai dengan Maret 2015. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Judul** | **Pengarang** | **Tahun** | **Permasalahan** | **Hasil** |
| 5. | Rancang Bangun Sistem Informasi Persediaan dan Peramalan Obat di Apotek dengan Menggunakan Metode *Ecponential Smoothing dan Economic Order Quantity* | Bagus Dhaist Rananta | 2016 | Sulit untuk melihat ketersediaan obat, tidak dapat diketahui apabila terdapat obat yang persediaannya sudah habis (*stock out*), jadwal pembelian obat yang tidak menentu, dan *stock opname*  membutuhkan waktu yang lama. | Aplikasi persediaan dan peramalan apotek dengan menggunakan metode *Exponential smoothing* dan *Economic Order Quantity.* |

Tabel 2.1 menjelaskan tentang perbandingan antara penelitian sebelumnya dengan penelitian yang sedang dilakukan dilihat dari permasalahan dan hasil dari penelitian.

## 2.2 Apotek

Apotek Berasal dari Bahasa Yunani *apotecha* yang secara harfiah memiliki arti “penyimpanan”. Definisi apotek adalah tempat menjual dan terkadang digunakan sebagai tempat membuat atau meramu obat. Apotek dapat mendistribusikan perbekalan farmasi dan perbekalan kesehatan dari *supplier* kepada pelanggan. Apotek memiliki beberapa fungsi kegiatan, yaitu pembelian, gudang, pelayanan dan penjualan, keuangan, dan pembukuan. Apotek tidak hanya mengejar keuntungan saja, tetapi apotek mempunyai fungsi sosial yang menyediakan, menyimpan, dan menyerahkan perbekalan farmasi yang berkualitas baik dan terjamin keabsahannya (Wibisetiadi, 2011)

Apotek adalah suatu tempat yang di dalamnya berisi pekerjaan kefarmasian dan penyaluran pembekalan farmasi kepada masyarakat. Pekerjaan kefarmasian memiliki maksud pengadaan obat, penyimpanan obat, pembuatan sediakan obat, peracikan, penyaluran, dan penyerahan perbekalan farmasi serta memberikan informasi kepada masyarakat mengenai perbekalan kefarmasian (Wijaya, 2014).

## 2.3 Persediaan

Persediaan merupakan suatu istilah yang menunjukkan segala sesuatu dari sumber daya yang terdapat dalam suatu proses yang bertujuan untuk mengantisipasi terhadap segala kemungkinan yang terjadi baik karena adanya permintaan atau masalah lain (Mulyani, 2013).

Persediaan adalah bahan-bahan, bagian yang disediakan, dan bahan-bahan dalam proses yang terdapat dalam perusahaan untuk proses produksi, serta barang-barang jadi atau produk yang disediakan untuk memenuhi permintaan dari konsumen atau pelanggan (Lestari & Setyorini, 2014).

Menurut Sri (2013), Fungsi dari persediaan bagi perusahaan adalah:

1. Agar dapat memenuhi permintaan yang diantisipasi akan terjadi,
2. Untuk menyeimbangkan antara produksi dan distribusi,
3. Untuk memperoleh keuntungan dari potongan kuantitas, karena pembeli dalam jumlah banyak mendapatkan diskon,
4. Untuk menghindari kurangnya persediaan yang dapat terjadi karena cuaca, kekurangan pasokan, mutu, dan ketidaktepatan pengiriman,
5. Untuk menjaga kelangsungan operasi dengan cara persediaan proses.

Mengendalikan persediaan yang tepat bukan hal yang mudah, apabila jumlah persediaan terlalu besar, akan menimbulkan dana menganggur yang besar, meningkatnya biaya penyimpanan, dan resiko kerusakan barang yang lebih besar. Namun, jika persediaan terlalu sedikit dapat mengakibatkan resiko terjadinya kekurangan persediaan karena seringkali bahan/barang tidak dapat didatangkan secara mendadak dan sebesar yang dibutuhkan, yang menyebabkan terhentinya proses produksi, tertundanya penjualan, bahkan hilangnya pelanggan (Meilani & Eka, 2013).

Biaya persediaan terdiri dari seluruh pengeluaran, baik yang langsung maupun yang tidak langsung, yang berhubungan dengan pembelian, persiapan, dan penempatan persediaan untuk dijual. Biaya persediaan termasuk biaya pembelian, pengiriman, penerimaan, penyimpanan, dan seluruh biaya yang terjadi sampai pada saat barang siap dijual. Penentuan besarnya persediaan sangat penting bagi perusahaan, karena persediaan memiliki efek langsung terhadap keuntungan perusahaan. Kesalahan dalam menentukan besarnya investasi (yang ditanamkan) dalam persediaan akan menekankan keuntungan perusahaan (Mulyani, 2013).

Menurut Aulia ishak (2010), biaya dalam sistem persediaan secara umum dapat diklasifikasikan sebagai berikut (Lestari & Setyorini, 2014):

1. Biaya pembelian (*purchasing cost* = c), merupakan biaya pembelian setiap barang jika barang tersebut berasal dari eksternal, atau biaya produksi setiap barang bila barang tersebut berasal dari internal perusahaan atau diproduksi sendiri oleh perusahaan. biaya pembelian ini dapat bervariasi untuk berbagai ukuran pemesanan bila pemasok menawarkan potongan harga untuk ukuran pemesanan yang lebih besar.
2. Biaya pengadaan (*procurement cost*), biaya ini dibedakan atas dua jenis sesuai dengan asal-usul barang, yaitu biaya pemesanan (*ordering cost*) bila barang yang diperlukan diperoleh dari pihak luar (*supplier*) dan biaya pembuatan (*set up cost*) bila barang diperoleh dengan memproduksi sendiri.
3. Biaya pemesanan (*ordering cost* = k), merupakan semua biaya yang timbul untuk mendatangkan barang dari luar.
4. Biaya pembuatan (*set up cost* = P), merupakan semua pengeluaran yang ditimbulkan untuk persiapan memproduksi barang.
5. Biaya penyimpanan (*holding cost* = h), merupakan biaya yang timbul akibat disimpannya suatu barang. Biaya penyimpanan terdiri atas biaya-biaya yang bervariasi secara langsung dengan kuantitas persediaan. Biaya penyimpanan per periode akan semakin besar apabila kuantitas bahan yang dipesan semakin banyak, atau rata-rata persediaan semakin tinggi.
6. Biaya kekurangan persediaan (*shortage cost* = p), merupakan biaya yang timbul apabila perusahaan kehabisan barang pada saat terdapat permintaan, maka akan terjadi kekurangan persediaan. Dari semua biaya-biaya yang berhubungan dengan tingkat persediaan, biaya kekurangan bahan (*stock out*) merupakan biaya yang paling sulit diperkirakan.

### 2.3.1 Economic Order Quantity (EOQ)

EOQ adalah jumlah unit (kuantitas) barang yang dapat dibeli dengan biaya minimal. Tujuan dari metode persediaan ini adalah untuk menentukan jumlah pesanan yang dapat meminimumkan biaya penyimpanan dan biaya pemesanan barang persediaan. Dengan menggunakan metode EOQ, maka persediaan yang terdapat di dalam gudang tidak terlalu banyak dan juga tidak terlalu sedikit, sehingga segala aktivitas perusahaan tidak akan terganggu. Salah satu masalah dalam analisis EOQ adalah sulitnya menentukan titik pemesanan kembali (*reorder point*). Titik pemesanan kembali itu digunakan untuk mencegah terjadinya kehabisan barang (*stock* out) selama waktu antara melakukan pemesanan dan penerimaan pesanan tersebut (Royyan, 2015).

Menurut Ayhari (1995), untuk dapat menentukan tujuan persediaan, maka perusahaan harus memenuhi beberapa faktor tentang persediaan bahan baku (Fatona, 2013). Adapun faktor-faktor tersebut antara lain:

1. Perkiraan Pemakaian, sebelum melakukan pembelian, maka manajemen harus dapat membuat perkiraan bahan baku yang akan dipergunakan dalam proses produksi pada suatu periode.
2. Harga dari bahan baku, harga dari bahan baku merupakan dasar penyusunan perhitungan berapa besar dana perusahaan yang harus disediakan untuk investasi dalam persediaan bahan baku tersebut. Oleh karena itu, maka biaya modal (*cost of capital*) yang dipergunakan dalam persediaan bahan baku tersebut harus diperhitungkan.
3. Biaya-biaya persediaan, merupakan biaya untuk menyelenggarakan persediaan bahan baku dan sudah selayaknya diperhitungkan dalam penentuan besarnya persediaan bahan baku.
4. Pemakaian senyatanya, pemakaian bahan baku dari periode-periode yang lalu (*actual demand*) merupakan salah satu faktor yang perlu diperhatikan karena keperluan proses produksi akan dipergunakan sebagai salah satu dasar pertimbangan dalam pengadaan bahan baku pada periode selanjutnya.
5. Waktu tunggu (*lead time*), merupakan tenggang waktu yang diperlukan (yang terjadi) antara saat pemesanan bahan baku dengan datangnya bahan baku itu sendiri. Waktu tunggu ini perlu diperhatikan karena sangat erat hubungannya dengan penentuan saat pemesanan kembali (*reorder point*).
6. Model pembelian bahan, manajemen perusahaan harus dapat menentukan model pembelian yang paling sesuai dengan situasi dan kondisi bahan baku yang akan dibeli. Model pembelian yang optimal atau *Economic Orer Quantity* (EOQ).
7. Persediaan bahan pengaman (*safety stock*), persediaan pengaman adalah persediaan tambahan yang diadakan untuk melindungi atau menjaga kemungkinan terjadinya kekurangan bahan (*stock out*).
8. Pemesanan kembali (*reorder point*), merupakan saat atau waktu perusahaan harus mengadakan pemesanan bahan baku kembali, sehingga datangnya pemesanan tersebut tepat dengan habisnya bahan baku yang dibeli.

Persamaan untuk kuantitas pembelian optimal (Royyan, 2015):

EOQ = Q\* = √2 CR / H (2.1)

H = P × f (2.2)

B = Q x L / N (2.3)

Keterangan:

R : Permintaan per-periode (unit).

C : Biaya pemesanan setiap pesanan (Rp).

Q : Jumlah setiap pesanan pada setiap pembelian (unit).

H : Biaya penyimpanan perunit (Rp).

P : Harga pembelian (Rp) perunit.

f : Biaya penyimpanan perunit yang dinyatakan dalam persentase.

B : Titik pemesanan kembali (unit).

N : Banyaknya periode lead time dalam periode permintaan.

L : Waktu tunggu (Lead time).

Untuk meminimalkan bahan baku yang tersedia dalam menjamin kelancaran proses produksi dan biaya yang dikeluarkan oleh perusahaan maka tindakan yang perlu dilakukan adalah menentukan *Economic Order Quantity* (EOQ), *Reorder Point* (RoP), dan *Safety Stock.* Menurut Wahyu dan Fatoni (2012), syarat data yang menggunakan metode EOQ antara lain:

1. Tingkat permintaan tidak diketahui dan bersifat konstan (*deterministic*).
2. *Lead time* tidak diketahui.
3. Barang yang dipesan diasumsikan dapat segera tersedia (*instaneously*) atau tingkat produksi (*production rate*) barang yang dipesan berlimpah (tak terhingga).
4. Setiap pesanan diterima dalam sekali pengiriman dan langsung dapat digunakan.
5. Tidak terdapat pesanan ulang (*back order*) karena kehabisan persediaan (*storage*).
6. Harga pembelian atau biaya pembuatan tidak berubah-ubah.
7. Tidak terdapat potongan harga (*quantity discount*).
8. Variabel biaya hanya biaya pesan (*ordering cost*) dan biaya simpan (*holding cost).*

Menurut Kartika Hendra (2009), keunggulan dari metode EOQ adalah (Valerie & Sinuraya, 2011):

1. Dapat digunakan untuk mengetahui berapa banyak persediaan yang harus dipesan, dalam hal ini bahan baku, dan kapan seharusnya pemesanan dilakukan.
2. Dapat mengatasi ketidakpastian permintaan dengan persediaan pengamanan (*safety stock).*
3. Mudah diaplikasikan pada proses produksi secara massal.
4. Lazim digunakan pada rumah sakit, khususnya pada persediaan obat.

### 2.3.2 Just In Time (JIT)

Menurut Hansen & Maryanne (2001), *Just In Time (JIT)* merupakan suatu pendekatan manufaktur yang mempertahankan bahwa produk-produk harus ditarik dari seluruh sistem dengan adanya permintaan, dan bukannya mendorong seluruh sistem dengan skedul yang tetap untuk mengantisipasi permintaan.

*Just In Time (JIT)* dapat berarti banyak hal yang berbeda-beda bagi masyarakat, baik masyarakat bisnis maupun masyarakat umum. Beberapa pihak menganggap *Just In Time (JIT)* adalah suatu pendekatan; bagi pihak lain *JIT* adalah suatu metodologi, atau suatu filosofi, atau suatu konsep atau suatu strategi (Soewarno,2005).

Menurut Hansen & Maryanne (2005), *Just In Time (JIT)* memiliki dua tujuan strategis, yaitu untuk meningkatkan laba dan untuk memperbaiki posisi bersaing perusahaan. Kedua tujuan ini dapat dicapai dengan mengendalikan biaya (yang memungkinkan persaingan harga yang lebih baik dan peningkatan laba), memperbaiki kinerja pengiriman dan meningkatkan kualitas.

Menurut Putu Sulastri (2012), JIT bukan hanya sekedar metode pengendalian persediaan tetapi juga merupakan system produksi yang saling berkaitan dengan semua fungsidan aktivitas. Manfaat JIT antara lain:

1. Mengurangi ruangan gudang untuk penyimpanan barang,
2. Mengurangi waktu setup dan penundaan jadwal produksi,
3. Mengurangi pemborosan barang rusak dan cacat dengan mendeteksi kesalahan pada sumbernya,
4. Penggunaan mesin dan fasilitas secara baik,
5. Menciptakan hubungan yang lebih baik dengan pemasok,
6. Layout pabrik yang lebih baik,
7. Pengendalian kualitas dalam proses.

Tabel 2.2 Perbedaan *Just In Time* dengan Sistem Konvensional (Mayora, 2015)

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Aspek Perbedaan** | **Filosofi Just In Time** | **Filosofi Tradisional** |
| 1 | Kualitas | *Quality is free* | Untuk menghasilkan produk yang berkualitas dibutuhkan biaya |
| 2 | Keahlian | Para pekerja merupakan para ahli, manajer dan insinyur melayani mereka | Manajer dan insinyur adalah orang ahli. Para pekerja melayani apa yang mereka inginkan |
| 3 | Persediaan | Persediaan hanya menyembunyikan masalah yang sesungguhnya muncul di permukaan. | Persediaan berguna untuk menjamin kelancaran produksi, yaitu sebagai penyangga terhadap kerusakan atau masalah lain |
| 4 | Kesalahan | Kesalahan merupakan pelajaran yang dapat menghasilkan perbaikan. | Kesalahan adalah hal yang tidak dapat dihindari dan harus ditelaah |
| 5 | Ukuran *lot* | Ukuran *lot* harus kecil, diharapkan hanya satu. | Ukuran *lot* harus ekonomis, yaitu menggunakan prinsip EOQ |
| 6 | Antrian | Produksi harus *Just In Time,*tidak boleh ada antrian panjang. | Antrian dalam *Work In Process* dibutuhkan untuk memastikan utilisasi mesin tinggi. |
| 7 | Aliran Material | Material harus ditarik ke dalam pabrik. | Material harus dikoordinir dan didorong keluar oleh pabrik. |
| 8 | Nilai Otomatisasi | Otomatisasi bernilai karena memungkinkan terjadinya konsistensi kualitas. | Otomatisasi bernilai karena dapat mengurangi tenaga kerja dalam proses produksi. |
| **No** | **Aspek Perbedaan** | **Filosofi Just In Time** | **Filosofi Tradisional** |
| 9 | Sumber pengurangan biaya | Pengurangan biaya diperoleh dari mempercepat aliran produk didalam pabrik, waktu proses singkat sangat bernilai. | Pengurangan biaya dilakukan dengan mengurangi penggunaan tenaga kerja, dengan utilisasi mesin yang tinggi. Tingkat produksi yang tinggi sanga bernilai. |
| 10 | Fleksibilitas | Fleksibilitas berasal dari memadatkan semua lead time waktu proses pabrik, waktu pengembangan produk baru, order entry, dan production planning cycles, dan sebagainya. | Fleksibilitas membutuhkan biaya kelebihan kapasitas, peralatan yang bersifat kapasitas, peralatan yang bersifat umum, persediaan, overhead, dan sebagainya. |
| 11 | Peran Overhead | Setiap pekerja yang tidak memberikan nilai tambah secara langsung pada produk adalah pemborosan. | Fungsi overhead adalah asensial. Fungsi-fungsi overhead seperti pembelian, industrial engineering, dan material handling dimaksudkan sebagai aspek koordinasi dari proses. |
| 12 | Biaya tenaga kerja | Biaya tenaga kerja merupakan biaya tetap | Biaya tenaga kerja merupakan biaya variabel. |
| 13 | Kecepatan Mesin | Mesin diibaratkan pelari maraton, lambat namun pasti, dan selalu mampu untuk berlari | Mesin diibaratkan pelari cepat. |
| 14 | Pembelian | Membeli dari pemasok yang terbatas. | Membeli dari banyak penjual |
| 15 | Expediting | Expediting dan work around adalah dosa. | Expediting dan work around adalah cara hidup. |

Tabel 2.2 menjelaskan tentang perbandingan antara filosofi *Just In Time* dengan filosofi tradisional dengan membandingkan beberapa aspek.

## 2.4Peramalan

Menurut Zulian (2003), peramalan (*forecasting*) merupakan alat bantu yang penting dalam perencanaan yang efektif dan efisien khususnya dalam bidang ekonomi. Dalam organisasi modern, mengetahui keadaan yang akan datang tidak saja penting untuk melihat yang baik dan buruk tetapi juga bertujuan untuk melakukan persiapan peramalan. Peramalan adalah prediksi, proyeksi, atau estimasi tingkat kejadian yang tidak pasti di masa yang akan datang (Sahara, 2013).

Dalam melakukan peramalan terdapat metode yang banyak digunakan tergantung dari tipe pola data dan tujuan penelitian itu sendiri. Sehingga pengidentifikasian pola data perlu dilakukan. Menurut Merkidarkis (1999), pola data deret waktu dapat dibedakan menjadi empat jenis (Nahar, 2013).

1. Pola stasioner, terjadi bilamana nilai data berfluktuasi di sekitar nilai rata-rata konstan.
2. Pola trend, terjadi bilamana terdapat kenaikan atau penurunan sekuler jangka panjang dalam data.
3. Pola musiman, terjadi bilamana suatu deret dipengaruhi oleh faktor musiman (misalnya kuartal tahun tertentu, bulanan, atau hari-hari pada minggu tertentu).
4. Pola siklis, terjadi bilamana datanya dipengaruhi oleh fluktuasi ekonomi jangka panjang.

Menurut Afni (2013), secara umum metode peramalan dapat diklasifikasikan dalam dua kategori utama, yaitu:

1. Metode Kuantitatif

Metode kuantitatif merupakan metode peramalan yang didasarkan pada data masa lalu dengan metode yang ada. Metode yang digunakan akan sangat mempengaruhi hasil dari peramalan. Metode kuantitatif dapat digunakan apabila terdapat tiga kondisi, yaitu adanya informasi tentang keadaan yang lain, informasi tersebut dapat dikuantifikasikan dalam bentuk data, dapat diasumsikan bahwa pola yang lalu akan berkelanjutan di masa yang akan datang (Himawan, 2014).

Metode kuantitatif dapat dibagi ke menjadi (Sahara, 2013):

1. Deret berkala atau runtun waktu (*time series*)
2. Indikator ekonomi
3. Model ekonometri
4. Metode Kualitatif

Metode kualitatif adalah peramalan yang berdasarkan data masa lalu yang ditentukan berdasarkan pemikiran yang bersifat intuisi, judgment atau pendapat, dan pengetahuan serta pengalaman dari penyusunnya (Himawan, 2014). Metode kualitatif dapat berupa pengumpulan pendapat yang dapat dibagi menjadi (Sahara, 2013):

1. Pengumpulan pendapat para ahli
2. Mengelompokkan dalam metode eksploratoris dan normatif.

Peramalan yang dibuat selalu diupayakan agar dapat meminimalkan pengaruh ketidakpastian terhadap perusahaan. Menurut Subagyo (1986), Peramalan bertujuan untuk mendapatkan peramalan yang dapat meminimalkan kesalahan meramal (*forecast error*) yang dapat diukur dengan *Mean Squared Error* (MSE), *Mean Absolute Error* (MAE), dan lain sebagainya (Himawan, 2014).

Menurut Santoso (2009), agar hasil peramalan dapat secara efektif menjawab masalah yang ada, kegiatan peramalan sebaiknya mengikuti tahapan baku sebagai berikut (Himawan, 2014):

1. Perumusan masalah dan pengumpulan data
2. Persiapan data, dalam prakteknya terdapat beberapa masalah berkaitan dengan data yang telah terkumpul:

* Jumlah data terlalu banyak.
* Jumlah data terlalu sedikit.
* Data harus diproses terlebih dahulu.
* Data tersedia namun rentang waktu data tidak sesuai dengan masalah yang ada.
* Data tersedia namun cukup banyak data yang hilang, yakni data yang tidak lengkap.

1. Membangun model
2. Implementasi model
3. Evaluasi peramalan.

Hasil Peramalan yang telah penulis dapatkan kemudian dibandingkan dengan data aktual. Tidak ada metode peramalan yang dapat secara tepat memprediksi data di masa depan, tetapi yang ada adalah ketepatan prediksi. Untuk itu diperlukan pengukuran kesalahan untuk melihat apakah metode yang digunakan telah memadai untuk memprediksi sebuah data di masa depan (Himawan, 2014).

### 2.4.1 Exponential Smoothing

Menurut Santoso (2009), *Exponential Smoothing* adalah salah satu tipe teknik peramalan rata-rata bergerak yang melakukan penimbangan terhadap data masa lalu dengan cara eksponensial sehingga data paling akhir memiliki timbangan lebih besar dalam rata-rata bergerak (Himawan, 2014). Menurut Markidaris (1999), dalam metode ini dasar pemikiran yang dilakukan adalah pembobotan terhadap data masa lalu dengan nilai data tersebut akan membentuk pola eksponensial. Metode peramalan ini mempertimbangkan aspek pola data yang terdapat dalam data deret waktu (Nahar, 2013).

Pada metode *single exponential smoothing* bobot yang diberikan pada data yang ada adalah sebesar α untuk data yang terbaru, α(1- α) untuk data yang lama, α(1- α)2 untuk data lebih lama, dan seterusnya. Besarnya nilai α adalah antara 0 dan 1. Semakin mendekati 1 berarti data terbaru lebih diperhatikan. Besarnya nilai peramalan menggunakan metode ini adalah dengan menggunakan persamaan (Sahara, 2013):

Ft+1 = α Xt + (1- α) F­t (2.4)

Keterangan:

Ft+1 = Ramalan untuk periode ke t + 1

Xt = Nilai riil periode ke t

Ft = Ramalan untuk periode ke t

Nilai parameter α yang sesuai akan memberikan ramalan yang optimal dengan nilai kesalahan (*error*) terkecil. Untuk mendapatkan nilai α yang tepat pada umumnya dilakukan dengan *trial and error* (coba-coba) untuk menentukan nilai kesalahan terendah (Facrurazzi, 2015). Untuk menentukan model yang paling baik dapat digunakan ukuran statistik untuk menghitung besarnya penyimpangan yaitu dengan menggunakan persamaan berikut (Nahar, 2013):

*Mean Square Error* (MSE) =  (2.5)

*Mean Absolute Error* (MAPE) =  (2.6)

Dengan nilai n adalah jumlah pengamatan.

### 2.4.2 Moving Average

*Moving average* adalah suatu metode peramalan umum dan mudah untuk menggunakan alat-alat yang tersedia untuk analisis teknis. *Moving average* menyediakan metode sederhana untuk pemulusan data masa lalu. Metode ini berguna untuk peramalan ketika tidak terjadi tren. Jika terdapat tren, gunakan estimasi berbeda untuk mempertimbangkannya. Hal ini disebut dengan ”bergerak” karena sebagai data baru yang tersedia, data yang tertua tidak digunakan lagi (Makridakis et all, 1999).

Tujuan utama penggunaan rata-rata bergerak adalah untuk menghilangkan atau mengurangi acakan dalam deret waktu. Teknik rata-rata bergerak dalam deret waktu terdiri dari pengambilan suatu kumpulan nilai-nilai yang diobservasi, mendapatkan rata-rata dari nilai ini, dan kemudian menggunakan nilai rata-rata tersebut sebagai ramalan untuk periode yang akan datang. (Assauri,1984).

Adapun rumus dari metode *moving average* (MA) yang menggunakan rumus sebagai berikut (Sahara, 2013):

Ft+1 (2.7)

Keterangan:

Ft+1 = Ramalan untuk periode ke t + 1

XT = Nilai riil periode ke t

T = Jangka waktu rata-rata bergerak

Dalam model *Moving Average* dapat dilihat bahwa semua data observasi memiliki bobot yang sama yang membentuk rata-ratanya. Padahal data observasi terbaru seharusnya memiliki bobot yang lebih besar dibandingkan dengan data observasi di masa lalu. Hal ini dipandang sebagai kelemahan dalam metode *Moving Average* (Sungkawa & Tri, 2011).

Tabel 2.3 Perbedaan *Exponential Smoothing* dengan *Moving Average*

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Aspek** | **Exponensial Smoothing** | **Moving Average** |
| 1 | Bobot Data | Bobot yang diberikan pada data yang ada adalah sebesar α untuk data yang terbaru, α(1- α) untuk data yang lama, α(1- α)2 untuk data lebih lama, dan seterusnya. | Semua data observasi memiliki bobot yang sama yang membentuk rata-ratanya |
| 2 | Penggunaan | Digunakan untuk melakukan pemulusan terhadap suatu deret berkala dengan membuat rata-rata tertimbang dari sederetan data yang lalu. | Metode ini berguna untuk peramalan ketika tidak terjadi tren. Jika terdapat tren, gunakan estimasi berbeda untuk mempertimbangkannya. |

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Aspek** | **Exponensial Smoothing** | **Moving Average** |
| 3 | Keuntungan | Banyak mengurangi masalah penyimpanan data, sehingga tidak perlu lagi menyimpan semua data historis. | Menghilangkan atau mengurangi acakan dalam deret waktu |

Tabel 2.3 menjelaskan tentang perbedaan antara metode *Exponential Smoothing* dengan metode *Moving Average* dilihat dari bobot data, penggunaan, dan keuntungan dari masing-masing metode.

## 2.5 *SDLC (Software Development Life Cycle)*

*System Development Life Cycle* (SDLC) adalah semua proses dari pembangunan, implementasi, dan penyelesaian sebuah sistem informasi melalui beberapa proses tahapan dari proses inisiasi, analisis, desain, implementasi, dan perawatan untuk penyelesaian. Terdapat beberapa model dan metode dari SDLC, tetapi setiap model secara general terdiri dari beberapa tahap dan fase yang jelas (Radack, 2009).



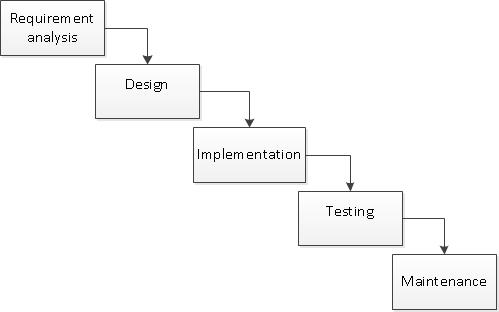
Gambar 2.1 Software Development Life Cycle (SDLC) (Maheswari & Jain, 2012)

Gambar 2.1 merupakan Penjabaran dari aktivitas dasar dalam membangun sebuah sistem. Menurut Shikha dan Dinesh (2012), aktifitas dasar atau fase dasar yang dilakukan untuk membangun sebuah sistem antara lain:

* Menentukan kebutuhan dari sistem
* Mendesain sistem
* Membangun (*coding) software*
* Mengetes sistem

Menurut Shikha dan Dinesh (2012), terdapat empat tipe model dari SDLC, yaitu:

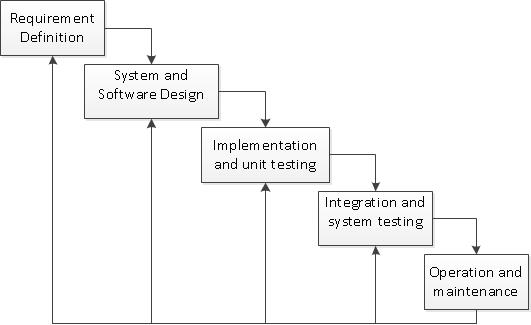
1. *Original Waterfall Model*

**

Gambar 2.2 *Waterfall Model* (Maheswari & Jain, 2012)

Gambar 2.2 adalah model dan tahapan dari *Original Waterfall model. Original Waterfall Model* adalah model klasik di dalam *software engineering.* Model ini merupakan salah satu model tertua dan banyak digunakan dalam proyek-proyek pemerintah dan di banyak perusahaan. Sebagai model yang menekankan perencanaan di tahapan awal, model ini memastikan kekurangan pada desain sebelum melakukan pengembangan. Sebagai tambahan, dokumen dan perencanaan yang intensif membuat model ini bekerja dengan baik untuk proyek-proyek yang perthatian utamanya adalah kualitas kontrol. Seperti gambar diatas, model ini terdiri dari tahapan yang tidak berulang (Munassar & Govardhan, 2010).

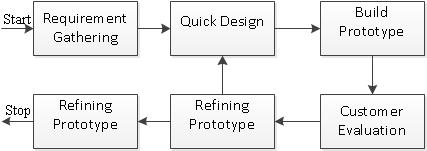
1. *Iterative Waterfall*

**

Gambar 2.3 *Iterative Waterfall* model (Maheswari & Jain, 2012)

Gambar 2.3 adalah model dan tahapan dari *Iterative Waterfall model. Iterative Waterfall* ini menggunakan fase yang sama dengan *original waterfall*, tetapi tidak menggunakan tahapan yang tidak berulang. Pada model ini memungkinkan terjadi pengulangan pada setiap tahap juka diperlukan (Munassar & Govardhan, 2010).

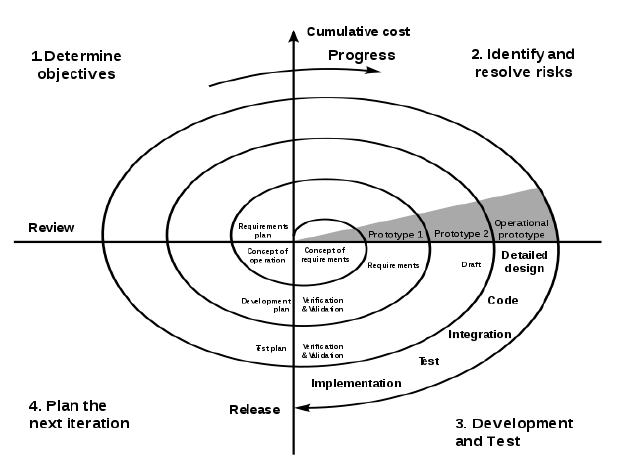
1. *Prototype*

**

Gambar 2.4 *Prototype* *Model* (Maheswari & Jain, 2012)

Gambar 2.4 adalah model dan tahapan dari *Prototype Model. Prototype model* adalah pendekatan pengembangan dari kegiatan selama proses pengembangan perangkat lunak, penciptaan prototipe, merupakan versi lengkap dari sistem yang sedang dikembangkan (Maheswari & Jain, 2012).

1. *Spiral*

**

Gambar 2. 5 *Spiral Model* (Maheswari & Jain, 2012)

Gambar 2.5 adalah model dan tahapan dari *Spiral model. Spiral Model* lebih melakukan penekanan pada resiko analisis. Model ini memiliki empat tahap, *planning, risk analysis, engineering, and evaluation.* Sebuah proyek berulang kali melewati fase-fase di dalam iterasi (pada model ini disebut *spiral*) (Munassar & Govardhan, 2010).

Tabel 2.4 Perbandingan dari empat metode SDLC (Maheswari & Jain, 2012)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Features** | **Original Waterfall Model** | **Iterative Waterfall** | **Prototyping Model** | **Spiral Model** |
| Requirement specification | Beginning | Beginning | Frequently changed | Beginning |
| Understanding requirements | Well understood | Not well understood | Not well understood | Well understood |
| Availability of reusable component | No | Yes | Yes | Yes |
| Complexity of the system | Simple | Simple | Complex | Complex |
| Risk analysis | Only at the beginning | No risk analysis | No risk analysis | Yes |
| **Features** | **Original Waterfall Model** | **Iterative Waterfall** | **Prototyping Model** | **Spiral Model** |
| *User* involvement in all phases SDLC | Only at beginning | Intermediate | High | High |
| Guarantee of success | Less | High | Good | High |
| Overlapping phases | No overlapping | No overlapping | Yes overlapping | Yes overlapping |
| Implementation time | Long | Less | Less | Depends on project |
| Flexibility | Rigid | Less flexible | Highly flexible | Flexible |
| Changes incorporated | Difficult | Easy | Easy | Easy |
| Expertise required | High | High | Medium | High |
| Cost control | Yes | No | No | Yes |
| Resource control | Yes | Yes | No | Yes |

Tabel 2.4 menjelaskan tentang perbandingan dari berbagai macam model di dalam SDLC yang dibandingkan dari berbagai macam aspek.

## 2.6 Konsep Dasar Sistem Informasi

Sistem adalah suatu kesatuan yang terdiri dari interaksi subsistem yang berusaha untuk mencapai tujuan yang sama, beroperasi dan berinteraksi dengan lingkungannya untuk mencapai tujuan tertentu. Suatu sistem menunjukkan tingkah lakunya melalui interaksi diantara komponen-komponen di dalam sistem dan diantara lingkungannya (Priyadna & Yulianto, 2012).

Adapun penjelasan tentang elemen-elemen dari sistem adalah (Royyan, 2015):

1. Tujuan, sistem dapat berupa tujuan usaha, kebutuhan pemecahan masalah, dan lain sebagainya.
2. Batasan, merupakan batasan-batasan yang terdapat dalam mencapai tujuan dari sistem yang dapat berupa peraturan-peraturan, permasalahan yang dibahas, peralatan, personil, dan lain sebagainya.
3. Penghubung, merupakan media antara satu subsistem dengan subsistem lainnya sehingga *output* (keluaran) dari subsistem akan dapat menjadi *input* (masukan) bagi subsistem lain.
4. *Input* (masukan), merupakan bagian yang bertugas untuk menerima data masukan, dimana data dapat berupa asal masukan, frekuensi pemasukan data dan jenis pemasukan data.
5. Proses, merupakan bagian yang memproses masukan data menjadi informasi yang sesuai dengan keinginan penerima.
6. *Output* (keluaran), merupakan keluaran atau tujuan akhir dari sistem yang dapat berupa laporan, tabel, atau grafik.

Informasi adalah data yang telah diklasifikasikan atau diolah atau diinterpretasikan untuk digunakan dalam proses pengambilan keputusan. Sistem pengolahan informasi akan mengolah data dari data yang tidak berguna menjadi data yang berguna bagi yang menerimanya (Aprilian, 2014).

Sistem informasi merupakan sistem yang berisi jaringan SPD (Sistem Pengolahan Data) yang dilengkapi dengan kanal-kanal komunikasi yang digunakan dalam sistem organisasi data. Elemen proses dari sistem informasi antara lain mengumpulkan data (*data gathering*), mengolah data yang tersimpan, dan menyebarkan informasi (Damayanti, et al., n.d.). Setiap sistem informasi menyajikan tiga hal pokok (Liliek, 2012):

1. Pengumpulan dan pemasukan data
2. Penyimpanan dan pengambilan kembali (*retrieval*) data
3. Penerapan data, yang dalam hal ini sistem informasi terkomputerisasi termasuk penayangan (*display*)

## 2.7 Metode Pengujian Sistem

Pengujian sistem adalah proses dari menjalankan program dengan tujuan untuk mencari kesalahan dalam program. Pengujian sendiri memiliki arti sebuah aktivitas yang dilakukan untuk mengevaluasi dan meningkatkan kualitas dari sebuah perangkat lunak (Jovanovic, 2009). Pengujian sistem mengidentifikasi kelemahan dan kesalahan di dalam kode aplikasi yang harus diperbaiki. Tujuan dari pengujian sistem dapat berupa jaminan kualitas dan memberikan ulasan tentang spesifikasi, desain, dan kode. Tujuan utama dari pengujian sistem adalah untuk memperkuat kualitas dari sistem dengan menguji sistem secara sistematis dalam keadaan yang terkontrol, tujuan lainnya adalah untuk mengidentifikasi kelengkapan dan kebenaran dari sistem, dan kemudian ditemukan kesalahan dari sistem (Khan & Khan, 2012).

Menurut Ehmer & Farmeena (2012), tiga teknik penting yang digunakan untuk mencari kesalahan adalah:

1. Teknik *White Box Testing*, merupakan investigasi secara mendalam mengenai logika internal dan struktur dari kode sumber. Di dalam teknik ini penguji harus memiliki pengetahuan tentang keseluruhan kode sumber yang terdapat di dalam aplikasi.
2. Teknik *Black Box Testing,* merupakan teknik dari pengujian yang tidak memiliki pengetahuan sama sekali terhadap pekerjaan yang terdapat dalam aplikasi. Teknik ini hanya menguji aspek dasar dari sistem dan tidak memiliki atau sedikit hubungan dengan struktur logika internal dari sistem.
3. Teknik *Grey Box Testing,* gabungan dari *White box dan Black box,* merupakan teknik untuk menguji aplikasi dengan pengetahuan yang terbatas mengenai pekerjaan di dalam aplikasi dan memiliki pengetahuan aspek dasar dari sistem.

Tabel 2.5 Perbandingan tiga teknik pengujian (Khan & Khan, 2012)

|  |  |  |
| --- | --- | --- |
| **Black Box Testing** | **Grey Box Testing** | **White Box Testing** |
| Menganalisa aspek dasar , tidak memiliki pengetahuan dari pekerjaan internal | Memiliki sebagian pengetahuan dari pekerjaan internal | Memiliki pengetahuan secara keseluruhan mengenai pekerjaan internal |
| Rincian yang rendah | Rincian yang sedang | Rincian yang tinggi |

|  |  |  |
| --- | --- | --- |
| **Black Box Testing** | **Grey Box Testing** | **White Box Testing** |
| Dilakukan oleh *end user* dan oleh penguji dan pengembang (*user acceptance testing)* | Dilakukan oleh *end user* dan oleh penguji dan pengembang (*user acceptance testing)* | Dilakukan oleh pengembang dan penguji |
| Memakan waktu yang sedikit | Memakan waktu yang tidak terlalu sedikit dan lama | Memakan waktu yang lama |
| Pengetesan berdasarkan keluaran dari sistem, internal dari sistem dihiraukan. | Pengetesan di desain berdasarkan tinggi level diagram database, diagram alur, *internal states,* pengetahuan dari algoritma dan arsitektur | Internal dari sistem diketahui secara keseluruhan |
| Dapat diuji dengan metode *trial and error* | Domain data dan *internal boundaries* dapat diuji dan melalui alur, jika diketahui | Pengujian lebih baik, domain data dan *internal boundaries* |
| Tidak sesuai dengan pengujian algoritma | Tidak sesuai dengan pengujian algoritma | Sesuai dengan pengujian algoritma (sesuai dengan semuanya) |

Tabel 2.5 merupakan penjelasan mengenai perbadaan antara tiga macam pengujian sistem, yaitu *Black Box Testing, Grey Box Testing,* dan *White Box Testing.*

## 2.8 *Object Oriented Programming (OOP)*

Pemrograman berorientasi objek merupakan paradigma pemrograman yang berorientasikan kepada objek, semua data dan fungsi di dalam paradigma ini dibungkus dalam kelas-kelas atau objek-objek. Setiap objek dapat mengirim pesan, memproses data, dan mengirim pesan ke objek lainnya (Rolliawati, 2013).

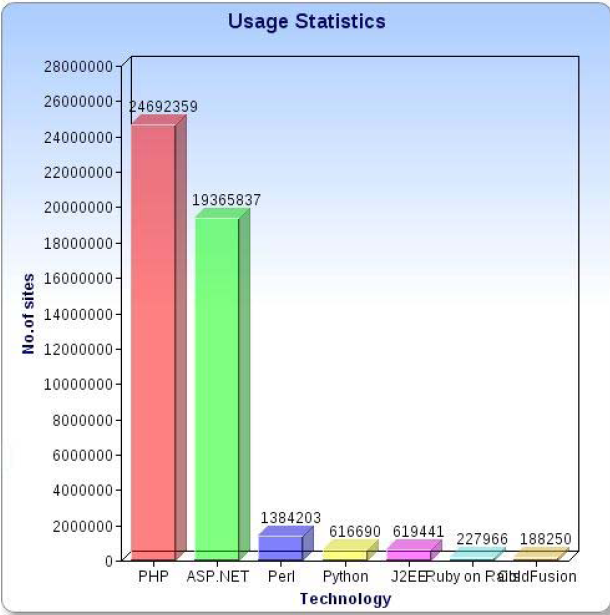
OOP (*Object Oriented Programming*) merupakan sebuah istilah yang diberikan kepada bahasa pemrograman yang menggunakan teknik berorientasi atau berbasis pada sebuah objek dalam pembangunan program aplikasi, pembuatan program tidak hanya menggunakan orientasi linear melainkan berorientasi pada objek-objek yang terpisah. Suatu perintah dalam bahasa ini diwakili oleh sebuah objek yang di dalamnya berisi beberapa perintah-perintah sederhana. Objek ini kemudian dikumpulkan dalam modul *form,* laporan, atau modul lain dan disusun dalam sebuah objek (Danuri, 2009).

Menurut Dewi (2009), pemrograman berorientasi objek memiliki beberapa keuntungan, seperti:

1. *Maintenance,* program lebih mudah dibaca dan dipahami, dan pemrograman berorientasi objek mengontrol kerumitan program hanya dengan mengijinkan rincian yang dibutuhkan untuk programmer.
2. Pengubahan program, seperti penambahan atau penghapusan fitur tertentu, perubahan yang dilakukan antara lain berupa penambahan dan penghapusan dalam suatu database program.
3. *Reusable,* dapat digunakannya objek-objek sesering yang diinginkan, dapat menyimpan objek-objek yang dirancang dengan baik ke dalam sebuah *toolkit* rutin yang bermanfaat dan dapat disisipkan ke dalam kode yang baru dengan sedikit perubahan pada kode tersebut.

## 2.9 Bahasa Pemrograman

Dalam pembuatan suatu sistem, diperlukan beberapa *tools* yang digunakan dalam merancang dan membangun sistem informasi berbasis web. Berikut ini akan diberikan beberapa *tools* yang dapat digunakan untuk pembuatan sistem informasi persediaan pada apotek. Beberapa tools dan perangkat lunak yang paling sering digunakan dalam membangun aplikasi web antara lain ASP.NET, PHP, ColdFusion, Perl, Python, J2EE, dan Ruby on Rails (Nagilla, 2013). Perbedaan penggunaan web teknologi tersebut dapat dijelaskan dalam diagram berikut ini:

  
 Gambar 2.6 menampilkan grafik penggunaan *tools* dalam melakukan pengembangan web. Di dalam statistik terlihat bahwa PHP dan ASP.NET mendominasi penggunaan tools dalam pengembangan web. Berikut penjelasan secara singkat mengenai PHP dan ASP.NET:

Gambar 2. 6 Statistik Penggunaan *Tools* untuk Pengembangan Web (Nagilla, 2013)

* PHP, merupakan singkatan dari "PHP: *Hypertext Preprocessor*", yang merupakan sebuah bahasa *scripting* yang terpasang pada *HyperText Markup Language* (HTML). Sebagian besar sintaks mirip dengan bahasa C, Java dan Perl, ditambah beberapa fungsi PHP yang spesifik (Suprianto, 2008).
* ASP.NET, merupakan singkatan dari *Active Server Pages. NET* dan juga dikenal sebagai *server side programming.* ASP.NET diluncurkan oleh *Microsoft Corporation.* Kebutuhan dari proyek ASP.NET adalah bahasa pendesain web (HTML), bahasa skrip *client side* (*Javascript*), bahasa pemrograman (C#.NET/VB.NET) dan *web server (*IIS/ASP.NET *Development Server)* (Nagilla, 2013)*.*

Tabel 2.6 Perbedaan antara PHP dan ASP.NET (Nagilla, 2013)

|  |  |  |
| --- | --- | --- |
| **Fitur** | **ASP.NET** | **PHP** |
| Biaya | Harus dibeli | Gratis |
| Kecepatan | Ya | Tidak |
| Dukungan bahasa | Banyak | Tidak ada |
| *Editor* | *Visual Studio* | *Zend, Eclipse, Code Lobster, Adobe Dreamweaver, Net beans, context*, dll |
| Kompatibel database | MS-SQL, MySQL | MS-SQL, MySQL |
| *Coding* | 325 baris kode untuk fungsi utama | 386 baris kode untuk fungsi utama |
| Konektivitas *platform* | *Windows* | Banyak (*Windows*, LINUX, UNIX) |
| *Support* | Perbaruan tersedia dari *Microsoft* | Banyak dukungan dari forum atau blog tentang PHP |
| *Tools* | Tersedia di dalam *Visual Studio* | Harus mengumpulkan dari pihak ketiga |
| *Space* | 23MB | 3MB |
| *Designing Code* | Otomatis | Manual |
| Kontrol data | Ya | Tidak |
| Kontrol validasi | Ya | Tidak |

Tabel 2.6 merupakan perbedaan antara dua macam tools di dalam pengembangan sistem yang memiliki banyak pengguna, yaitu PHP dan AHP.

# BAB III

# METODOLOGI PENELITIAN

## 3. 1 Alat Penelitian

Dalam melakukan ini, dibutuhkan alat penelitian yang digunakan dalam pembuatan sistem ini. Alat yang digunakan dalam pembuatan sistem ini dibagi menjadi dua kelompok, yaitu:

1. Perangkat Keras (*Hardware*)

Perangkat keras yang digunakan dalam penelitian ini adalah sebuah *laptop Lenovo® Thinkpad™ 4421s,* dengan spesifikasi sebagai berikut :

* Prosesor : 2,5 GHz
* Memory : 6 GB
* Hardisk : 512GB
* Network Interface : 100/1000 Mbps*.*

Perangkat keras ini digunakan dalam seluruh proses penelitian, dari proses analisis, proses desain sistem, dan implementasi sistem.

1. Perangkat Lunak (*Software*)

Perangkat lunak yang digunakan dalam melakukan penelitian ini antara lain:

* *Microsoft® Office™ Word 2007,* digunakan untuk pengolahan kata untuk mencatat informasi dan data-data yang digunakan untuk membuat sistem ini.
* *Visual Paradigm™,* digunakan sebagai pengolah desain dalam membuat perancangan dari sistem yang akan dibuat.
* *XAMPP,* sebagai W*eb server apache,* digunakan sebagai server dan pengolah database MySQL.
* *Sublime Text ™* 3*,* digunakan untuk penulisan dan penyuntingan bahasa pemrograman. Bahasa pemrograman yang digunakan pada penelitian ini adalah PHP.
* *Google® Chrome,* sebagai penjelajah web untuk melihat implementasi dari sistem yang telah dirancang.

## 3. 2 Metode Pengumpulan Data

Dalam mengerjakan penelitian ini, diperlukan data-data dan informasi yang digunakan untuk menganalisa permasalahan yang akan dibahas agar sesuai dengan tujuan akhir penelitian ini. Data-data dan informasi diperoleh dengan menggunakan dua metode, yaitu penelitian lapangan dan studi pustaka.

1. Penelitian lapangan, yaitu sebuah bentuk penelitian yang dilakukan dengan melakukan wawancara dan observasi. Hal ini dilakukan untuk mendapatkan data yang lebih tepat dan dapat terpercaya sehingga sesuai dengan kebutuhan yang diperlukan untuk mendukung penelitian ini. Data tersebut merupakan data primer dan data sekunder.
2. Data primer, yaitu data yang diperoleh secara langsung di lapangan dengan menggunakan metode wawancara secara langsung kepada pengelola apotek. Wawancara dilakukan pada tanggal 23 September 2015 di Apotek xyz dengan pemilik apotek sebagai narasumber. Wawancara yang dilakukan bertujuan untuk mendapatkan informasi mengenai keadaan sistem pada saat ini, keinginan pelanggan terhadap sistem yang akan dibuat, dan masalah yang dihadapi oleh pihak apotek.
3. Data sekunder, yaitu data yang telah diolah sebelumnya, dan didapat dari dokumentasi yang telah dilakukan oleh perusahaan. dalam penelitian in, data sekunder yang dibutuhkan adalah:

* Data yang digunakan untuk peramalan berupa data permintaan obat pada tahun sebelumnya.
* Biaya penyimpanan obat.
* Biaya pemesanan obat.
* Waktu tenggang pemesanan.

1. Studi Pustaka, yaitu mengambil data dari literatur, modul, buku panduan, dan juga beberapa kepustakaan yang dapat digunakan untuk mendukung penelitian ini. Teknik ini dilakukan dengan tujuan untuk mendapatkan data-data dan informasi yang digunakan sebagai panduan dan pendukung untuk menyelesaikan penelitian ini. Data-data yang didapatkan merupakan data yang mencakup tentang peramalan, dan pengendalian persediaan.

## 3. 3 Pengolahan Data

Pengolahan data dilakukan berdasarkan tahapan-tahapan berikut ini:

1. Peramalan jumlah obat untuk periode berikutnya

Peramalan dilakukan dengan metode *exponential smoothing* dengan mengacu pada data penjualan pada periode-periode sebelumnya. Hal yang perlu dilakukan apabila menggunakan metode ini adalah dengan menghitung setiap α dimana nilai dari α adalah 0,1 sampai 0,9. Setelah mendapatkan hasil peramalan dari setiap nilai α, maka dicari α dengan deviasi terkecil dengan menggunakan *Mean Square Error* (MSE) dan *Mean Absolute Error* (MAPE). Nilai α dengan MSE dan MAPE terkecil yang akan dipilih untuk menghitung peramalan.

1. Perhitungan total biaya persediaan dengan menggunakan metode EOQ.

Pada tahapan ini, selain menghitung total biaya persediaan dengan menggunakan metode EOQ, dilakukan perhitungan lainnya seperti perhitungan titik pemesanan kembali*.* Data peramalan obat yang di dapat dari proses sebelumnya kemudian dijadikan sebagai permintaan disetiap periode yang merupakan salah satu data yang digunakan untuk menghitung jumlah EOQ.

1. Pengembangan sistem.

Pembuatan sistem dilakukan dengan melakukan beberapa tahapan. Tahapan tersebut dipilih berdasarkan kesesuaian dengan situasi dan kondisi penelitian pada saat ini. Pada penelitian ini dipilih metode *iterative waterfall* untuk pengembangan sistem karena sistem yang dibuat sederhana, dan biaya yang dimiliki terbatas. Oleh karena itu, metode *iterative waterfall* merupakan metode yang cocok digunakan dalam pembuatan sistem ini.

1. Kesimpulan dan Saran.

Setelah diperoleh hasil dan cara pemecahan masalah, maka pada tahapan ini dilakukan penarikan kesimpulan berdasarkan hasil pengolahan data dan memberikan saran-saran yang membangun bagi pihak apotek maupun untuk penelitian lebih lanjut.

## 3. 4 Metode Pengembangan Sistem

Dalam melakukan sebuah penelitian, diperlukan sebuah model perancangan sistem yang digunakan untuk memecahkan masalah yang sebenarnya. Oleh karena itu dalam pengembangan sistem ini menggunakan *iterative waterfall* sebagai model acuan dalam pembuatan sistem. *Iterative waterfall* dipilih karena dari perbandingan antara beberapa metode yang ada, metode ini merupakan metode yang paling sesuai dengan keadaan dan kondisi dari penelitian ini.

Sesuai dengan tahapan di dalam model *iterative waterfall*, perancangan sistem yang dilakukan dalam penelitian ini adalah sebagai berikut:

### 3.4.1 Analisis dan Definisi Persyaratan

Tahap ini merupakan tahapan awal dari penelitian. Tahap ini merupakan tahap pencarian topik dan ide penelitian. Topik yang dicari adalah mengenai sistem informasi persediaan obat pada apotek dan mengenai metode EOQ (*Economic Order Quantity*), dan *exponential smoothing*. Setelah mendapatkan topik, peneliti mencari subjek dan objek untuk melakukan penelitian yaitu Apotek xyz dan mencari tahu apakah penelitian yang dilakukan berguna bagi mereka.

Pada definisi persyaratan, dilakukan pendefinisian dari seluruh kebutuhan yang dapat dilayani oleh sistem yang akan dibangun. Kebutuhan dari sistem dilakukan dengan mengumpulkan data-data dan informasi yang diperlukan dalam penelitian ini. Data-data dan informasi didapat dari tiga metode, yaitu dengan studi pustaka, wawancara, dan observasi. Studi pustaka dilakukan sebagai acuan dalam melakukan penelitian dengan melihat penelitian sejenis yang telah dilakukan sebelumnya. Studi pustaka dilakukan dengan mempelajari teori-teori dari jurnal-jurnal, buku, dan dokumen yang memiliki keterkaitan dengan penelitian. Metode wawancara dan observasi dilakukan untuk melihat kebutuhan dari pihak apotek, sehingga kebutuhan dari pengguna terpenuhi dan sesuai dengan keinginan.

Pada tahap analisis difokuskan untuk mengumpulkan kebutuhan sistem dengan lebih spesifik. Hal yang harus dilakukan pada tahap ini adalah memahami domain informasi, fungsi, tingkah laku, dan antarmuka yang dibutuhkan. Kebutuhan untuk sistem didokumentasikan dan ditinjau kembali pada tahap analisis. Hasil akhir yang diperoleh pada tahap ini adalah *Data Flow Diagram* (DFD)*, Entity-Relationship Diagram* (ERD)*,* dan *Data Dictionary* (Kamus Data).

### 3.4.2 Perancangan Sistem dan Perangkat Lunak

Pada tahap ini dilakukan perancangan dan penggambaran tentang sistem yang akan dibuat. Perancangan ini melibatkan identifikasi dan deskripsi dari abstraksi sistem yang mendasar dan hubungan-hubungannya. Perancangan sistem dan perangkat lunak terdiri dari perancangan data, perancangan arsitektur, dan perancangan antarmuka.

1. Perancangan data, merupakan perbaikan terhadap model domain informasi menjadi struktur data. Model domain informasi yang dibuat selama analisis ditransformasikan ke dalam struktur data yang akan diperlukan untuk mengimplementasikan sistem. Objek dan hubungan yang telah di tetapkan di dalam ERD, dan data rinci yang digambarkan di dalam kamus data dijadikan dasar bagi aktivitas perancangan data.
2. Perancangan Arsitektur, merupakan proses merubah aliran informasi (DFD) menjadi struktur data perangkat lunak (*Structure chart).* Tujuan utama dari perancangan ini adalah untuk membangun struktur program modular dan mempresentasikan keterkaitan antar modul, memadukan struktur program, struktur data yang mendefinisikan antar muka yang memungkinkan data dapat mengalir pada seluruh sistem.
3. Perancangan Antarmuka, merupakan penggambaran mengenai cara sistem berkomunikasi di dalam dirinya sendiri, dengan sistem yang berinteraksi dengannya, dan dengan pengguna yang menjalankan sistem.

### 3.4.3 Implementasi dan Pengujian Unit

Pada tahap ini, perancangan sistem yang telah dibuat diterapkan sebagai serangkaian program atau unit program. Sistem diterapkan dengan melakukan penulisan kode yang merupakan terjemahan dari hasil perancangan ke dalam perangkat keras. Penulisan kode menggunakan bahasa pemrograman PHP dan menggunakan MySQL sebagai penyimpan data.*.*

Pada tahap ini dilakukan pengujian dari unit sistem yang telah dibangun. Pengujian dilakukan dengan menguji setiap unit dari sistem. Pengujian dari unit sistem ini digunakan untuk memverifikasi bahwa semua bagian unit dari sistem telah berjalan dengan baik dan telah memenuhi spesifikasinya.

### 3.4.4 Integrasi dan Pengujian sistem

Pada tahap ini, setiap unit sistem yang telah diuji diintegrasikan satu sama lain. Setelah diintegrasikan, maka akan dilakukan pengujian pada sistem secara keseluruhan. Pengujian ini dilakukan untuk menjamin bahwa semua persyaratan telah dipenuhi dan tiap-tiap fungsi dari sistem sudah terhubung dengan baik. Apabila pengujian sistem telah selesai dilakukan dan sistem telah berjalan sesuai dengan keinginan, maka sistem akan dikirim ke pelanggan dan dilakukan pengoperasian sistem. Pengujian sistem menggunakan teknik pengujian *black box.*

### 3.4.5 *Operation and Maintenance*

Tahapan ini tidak dilakukan dikarenakan pada tahap ini memerlukan waktu yang lama sedangkan penelitian ini bertujuan untuk mengerjakan tugas akhir. Oleh karena itu, tahapan ini tidak dilakukan dalam penelitian ini.

## 3.5 Jadwal Penelitian

Adapun jadwal kegiatan pada penelitian ini dapat pembaca lihat pada Tabel 3.1 berikut ini :

**Tabel 3.1 Jadwal Penelitian**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Jenis Kegiatan** | **2015** | | | | | | | | | | | | | | | | **2016** | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| **Sept** | | | | **Oct** | | | | **Nov** | | | | **Dec** | | | | **Jan** | | | | | **Feb** | | | | | **Mar** | | | | | **April** | | | | | **Mei** | | | | | **June** | | | | | **Juli** | | | |
| **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** | **1** | | **2** | **3** | **4** | **1** | | **2** | **3** | **4** | **1** | | **2** | **3** | **4** | **1** | | **2** | **3** | **4** | **1** | | **2** | **3** | **4** | **1** | | **2** | **3** | **4** |
| 1 | Analisis persyaratan |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |
| 2 | Definisi persyaratan |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |
| 3 | Perancangan sistem |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |
| 4 | Perancangan *software* |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |
| 5 | Implementasi unit |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |
| 6 | Pengujian unit |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |
| 7 | Integrasi sistem |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |
| 7 | Pengujian sistem |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |
| 8 | Penulisan tugas akhir |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |

Tabel 3.1 menjelaskan tentang penjadwalan dari beberapa aktifitas yang dilakukan di dalam penelitian ini

# BAB IV

# PEMBAHASAN

Bab ini menjelaskan pembahasan tentang pembuatan sistem informasi apotek dengan menggunakan metode *Exponential Smoothing* dan *Economic Order Quintity* (EOQ) pada Apotek Nadia Farma. Pembuatan sistem dilakukan dengan menggunakan metode *Iterative Waterfall.*

## 4.1 Pengumpulan Informasi

Pengumpulan informasi merupakan tahapan awal dalam penelitian ini yang termasuk ke dalam tahap analisis dan definisi sistem di dalam metode *Iterative Waterfall.* Pada tahap ini dilakukan observasi dan wawancara kepada pemilik Apotek Nadia Farma, yang berlokasi di Jl Ratu Dibalau, Tajung Senang, Bandar Lampung. pada tahapan ini didapatkan beberapa temuan yang langsung berinteraksi dengan sistem, yaitu:

1. Tipe Pengguna

Dalam melakukan proses bisnis, hanya melibatkan satu orang pengguna, yaitu staff yang bekerja di dalam apotek termasuk pemliknya. Pengguna memiliki kewenangan untuk melakukan penambahan, perubahan, dan penghapusan pada data-data yang terdapat di dalam sistem.

1. Proses Bisnis

Proses bisnis yang terjadi di dalam apotek didapatkan berdasarkan wawancara dan observasi secara langsung. Berikut merupakan proses bisnis yang terjadi pada Apotek Nadia Farma:

* Pembelian

Pada proses pembelian, belum terdapat jadwal yang tetap yang diberikan untuk melakukan pembelian. Pembelian obat dilakukan apabila obat tersebut stocknya tersisa sedikit atau bahkan sudah habis. Dalam melakukan pembelian, data pembelian dicatat ke dalam sebuah buku khusus pembelian yang berisi tentang semua informasi pembelian obat dan *supplier* obat tersebut.

* Penjualan

Pada proses penjualan, dilakukan pencatatan setiap kali ada obat yang terjual. Pencatatan penjualan ditulis di dalam sebuah buku khusus penjualan yang di dalamnya terdapat data obat-obat yang telah terjual. Data-data obat yang telah dicatat setiap harinya kemudian akan dirangkum setiap bulannya untuk mengetahui jumlah penjualan obat pada bulan tersebut.

* Data Obat dan *Supplier*

Data obat dan *supplier* berisi tentang nama obat, harga obat dan nama *supplier* tempat pembelian obat. Pendataan ini dilakukan oleh staff apotek dan dicatat dalam sebuah buku data obat dan dilakukan pendataan apabila ada obat baru yang masuk ke dalam apotek.

## 4.2 Analisis Kebutuhan Sistem

Penelitian ini mempunyai kebutuhan fungsional pada sistem yang berguna untuk dapat menjalankan perintah sesuai dengan kebutuhan sistem. Kebutuhan fungsional sistem dibuat berdasarkan hasil pengumpulan data melalui observasi dan wawancara yang dibuat dalam elisitasi tahap I, II, III yang telah disetujui oleh pemilik dan pengelola Apotek Nadia Farma yang terlampir pada lampiran.

### 4.2.1 Desain *Database*

Perancangan *database* dilakukan berdaasarkan analisis kebutuhan yang telah dilakukan pada proses analisis dan definisi sistem. Perancangan *database* masuk ke dalam tahapan perancangan sistem dan perangkat lunak pada metode *Iterative Waterfall.* Rancangan model ini berupa *entitiy*, atribut yang ada di dalamnya, dan relasi antar *entity*. Setelah melakukan analisis kebutuhan, maka diperoleh beberapa *entity* yang dibutuhkan di dalam membangun sistem. Semua *entity* ysang diperoleh mewakili nama tabel *database* pada sistem yang akan dibuat. Daftar *entity* dan keterangan masing-masing *entity* dapat dilihat pada tabel 4.1.

Tabel 4.1 Daftar *entity* dan keterangannya

|  |  |  |
| --- | --- | --- |
| **No** | **Nama *Entity*** | **Keterangan** |
| 1. | *User* | *Entity* yang bertindak sebagai pengguna dari sistem dan memiliki hak untuk mengakses semua informasi dan fitur yang disediakan oleh sistem. |
| 2. | *Supplier* | *Entity* yang berisi tentang informasi mengenai *supplier* obat. *Supplier* sendiri digunakan apabila dilakukan proses pembelian obat. |
| 3. | Barang | *Entity* yang berisi tentang informasi mengenai obat yang dijual di dalam apotek. |
| 4. | Pembelian | *Entity* yang berisi tentang data pembelian barang. Pada *entity* ini merinci pembelian obat yang terjadi di dalam apotek. |
| 5. | Penjualan | *Entity* yang berisi tentang data penjualan barang. Pada *entity* ini merinci penjualan obat yang terjadi di dalam apotek. |
| 6. | Ramalan | *Entity* ini berisi tentang informasi yang dibuthkan untuk melakukan peramalan data obat yang selanjutnya akan dilakukan perhitungan lebih lanjut. |
| 7. | Persediaan | *Entity* ini berisi tentang informasi mengenai persediaan obat di dalam apotek. |

Tabel 4.1 Menjelaskan tentang daftar *entity* dan penjelasan dari masing-masing *entity*. *Entity* tersebut akan mewakili informasi yang terdapat di dalam sistem yang akan dibuat. *Entity* tersebut memiliki relasi, relasi antar *entity* dijelaskan pada tabel 4.2 berikut:

Tabel 4.2 Hubungan antar *entity*

|  |  |  |
| --- | --- | --- |
| **No.** | **Nama *Entity*** | **Hubungan Antar *Entity*** |
| 1 | *User* | *one-to many* dengan *supplier* |
| *one-to many* dengan pembelian |
| *one-to many* dengan barang |
| *one-to many* dengan penjualan |
| 2 | *Supplier* | *one-to many* dengan pembelian |
| *many-to-one* dengan *user* |
| 3 | Barang | *one-to many* denganpembelian |
| one-to many dengan penjualan |
| *many-to-one* dengan ramalan |
| *many-to-one* dengan *user* |
| 4 | Pembelian | *many-to-one* dengan *supplier* |
| *many-to-one* dengan barang |
| *many-to-one* dengan *user* |
| 7 | Penjualan | *many-to-one* dengan barang |
| *many-to-one* dengan *user* |
| 8 | Ramalan | *one-to many* denganbarang |
| 9 | Persediaan | *one-to one* denganbarang |

Tabel 4.2 Menjelaskan tentang hubungan yang terjalin antar satu *entity* dengan lainnya. Berikut ini merupakan *conceptual database design* dari *database* yang dibangun berdasarkan hubungan yang telah ditentukan sebelumnya:

C:\Users\ThinkPad\Pictures\Conceptual Database.png

Gambar 4.1 *Conceptual database design* dari sistem

Gambar 4.1 merupakan sebuah *conceptual database design* dari sistem yang akan dibuat. Seperti yang telah dijelaskan sebelumnya, pada rancangan model akan diperoleh *entity,* hubungan antar *entity*, dan *attribute* dari *entity. Entity* dan hubungannya telah dijabarkan sebelumnya, *attribute* pada masing-masing *entity* dapat dilihat pada tabel 4.3:

Tabel 4.3 Daftar attribut dari masing-masing *entity*

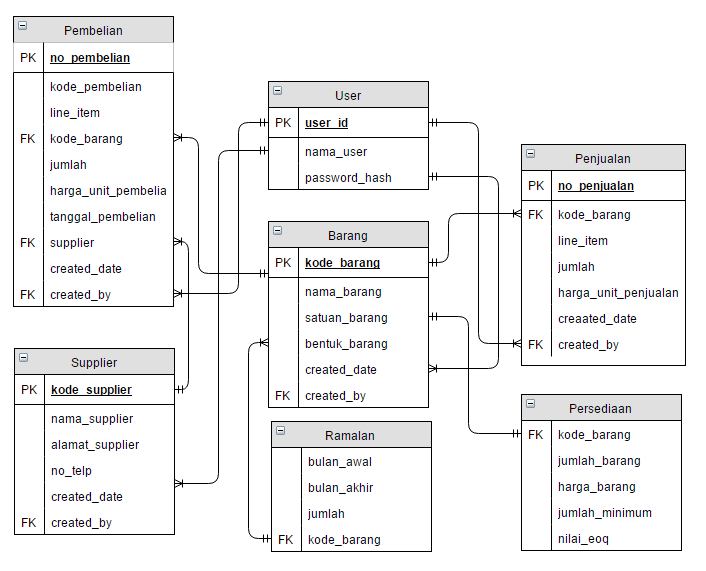
|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Nama *Entity*** | ***Attribute*** | **Keterangan** |
| 1 | *User* | *user*\_id | *Primary key* pada *entity user.* |
| nama\_*user* | Nama yang digunakan untuk autentikasi *user*. |
| *password*\_hash | Kata sandi yang digunakan untuk autentikasi pada sistem. |
| 2 | *Supplier* | kode\_*supplier* | *Primary key* pada *entity supplier.* |
| nama\_*supplier* | Informasi mengenai nama dari *supplier.* |
| alamat\_*supplier* | Informasi mengenai alamat dari *supplier.* |
| no\_telp | Informasi mengenai nomor telepon dari *supplier.* |

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Nama *Entity*** | ***Attribute*** | **Keterangan** |
|  |  | *create*d\_date | Informasi mengenai tanggal pembuatan  data *supplier*  di dalam sistem. |
| *create*d\_by | informasi tentang siapa yang membuat data *supplier* di dalam sistem. |
| 3 | Barang | kode\_barang | *Primary key* pada *entity* barang. |
| nama\_barang | Informasi mengenai nama dari obat. |
| satuan\_barang | Informasi mengenai satuan dari obat. |
| jenis\_barang | Informasi mengenai jenis barang. |
| *create*d\_date | Informasi mengenai tanggal pembuatan data obat di dalam sistem. |
| *create*d\_by | informasi tentang siapa yang membuat data obat di dalam sistem. |
| 4 | Pembelian | no\_pembelian | *Primary key* pada *entity* pembelian*.* |
| kode\_pembelian | Informasi tentang kode pembelian obat. |
| line\_item | *Attribute* ini digunakan agar dapat dilakukan pembelian lebih dari satu macam obat. |
| Jumlah | Informasi mengenai jumlah dari masing-masing jenis obat yang akan di beli. |
| kode\_barang | Informasi mengenai kode barang, merupakan *foreign key* dari tabel barang. |
| harga\_unit\_pembelian | Informasi mengenai harga satuan dari masing-masing jenis obat yang akan di beli. |
| tanggal\_pembelian | Informasi mengenai tanggal pembelian obat. |
| *Supplier* | Informasi mengenai *supplier* obat, merupakan *foreign key* dari tabel *supplier*. |

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | ***Entity*** | ***Attribute*** | **Keterangan** |
|  |  | *create*d\_date | Informasi mengenai tanggal pembuatan data pembelian obat di dalam sistem. |
| *create*d\_by | Informasi tentang siapa yang membuat data pembelian obat di dalam sistem. |
| 5 | Penjualan | no\_penjualan | *Primary key* pada *entity* penjualan*.* |
| kode\_barang | Informasi mengenai kode barang, merupakan foreign key dari tabel barang. |
| line\_item | *Attribute* ini digunakan agar dapat dilakukan penjualan lebih dari satu macam obat. |
| Jumlah | Informasi mengenai jumlah dari masing-masing jenis obat yang akan di jual. |
| harga\_unit\_penjualan | Informasi mengenai harga satuan dari masing-masing jenis obat yang akan di jual. |
| total\_harga | Informasi mengenai total harga dari penjualan obat berdasarkan kode pembelian obat. |
| *create*d\_date | Informasi mengenai tanggal pembuatan data penjualan obat di dalam sistem. |
| *create*d\_by | Informasi tentang siapa yang membuat data penjualan obat di dalam sistem. |
| 6 | Ramalan | bulan\_awal | Berisi Informasi tentang tanggal dimulainya suatu periode. |
| Bulan\_akhir | Berisi informasi mengenai tanggal berakhirnya suatu periode |
| kode\_barang | Informasi mengenai kode barang, merupakan foreign key dari tabel barang. |
|  |  | Jumlah | Infromasi tentang jumlah penjualan obat pada bulan yang telah ditentukan. |

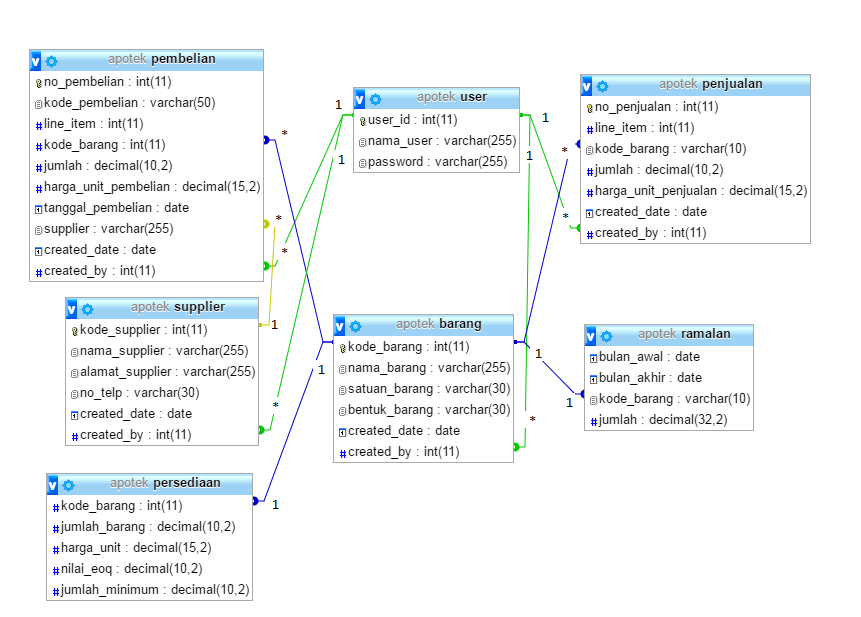
|  |  |  |  |
| --- | --- | --- | --- |
| **No** | ***Entity*** | ***Attribute*** | **Keterangan** |
| 7. | Persediaan | jumlah\_barang | Informasi mengenai jumlah dari obat yang terdapat di gudang. |
| harga\_unit | Informasi mengenai harga satuan dari obat. |
| id\_barang | Informasi mengenai id barang, merupakan *foreign key* dari tabel barang. |
| jumlah\_minimum | Informasi mengenai jumlah minimum dari barang, didapatkan setelah melakukan perhitungan EOQ. |
| Nilai\_eoq | Informasi mengenai nilai dari eoq yang telah dihitung di dalam sistem. |

Tabel 4.3 memberi penjelasan tentang *attribute* yang dimiliki dari masing-masing *entity*. Berdasarkan yang telah ditentukan sebelumnya baik dari *entity, attribute,* dan hubungannya, maka berikut ini adalah *logical database design* dari sistem. Pada skema ini telah ditentukan *primary, foreign, attribute,* dan hubungan dari masing-masing *entity.*



Gambar 4.2 *Logical database design* dari sistem

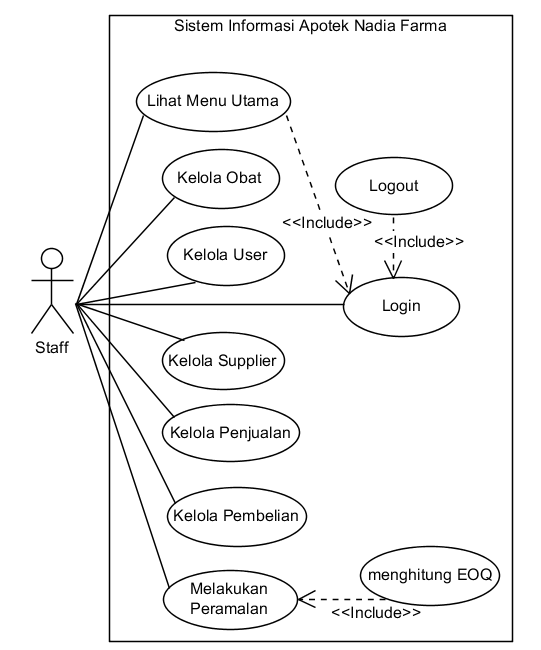
Gambar 4.2 merupakan gambaran dari physical *database* design yang telah dilengkapi dengan tipe data yang digunakan pada masing-masing *attribute*. Skema ini merupakan desain yang sebenarnya digunakan dalam pembuatan sistem. Terdapat tiga master tabel di dalam skema ini, yaitu tabel *user, supplier,* dan barang.



Gambar 4.3 Diagram *entity relationship diagram* pada sistem

Gambar 4.3 merupakan diagram *entity relationship* dari sistem yang akan dibuat. Dalam ERD sistem informasi apotek pada Apotek Nadia Farma terdapat delapan buah *entity*. Pengguna dalam sistem hanya diwakili oleh tabel *user* dan memiliki relasi dengan tabel barang, penjualan dan pembelian dan digunakan untuk melakukan *login* untuk masuk ke dalam sistem*.* Tabel barang memiliki hubungan dengan lima buah tabel, yaitu tabel pembelian, penjualan, *user*, persediaan, dan ramalan. Selain memiliki hubungan dengan tabel barang, tabel pembelian juga memiliki hubungan dengan tabel *supplier* dan *user*. Begitu pula dengan tabel penjualan, selain berhubungan dengan tabel barang, tabel penjualan memiliki hubungan dengan ramalan, dan *user*. Dalam *database* ini juga memiliki tabel *supplier* yang berhubungan tabel pembelian dan *user*. Terdapat tabel ramalan yang hanya memiliki hubungan dengan tabel penjualan. Terakhir, di dalam *database* ini terdapat tabel persediaan yang memiliki hubungan dengan tabel barang.

### 4.2.3 *Use case* Diagram



Gambar 4.4 *Use case diagram* dari sistem

Penjelasan dari gambar 4.4di atas dijelaskan melalui *use case scenario table*. *Use case scenario table* dapat menjelaskan perilaku setiap *use case* dan reaksi aktor serta sistem.

Tabel 4.4 *Use case Scenario Login*

|  |  |  |
| --- | --- | --- |
| ***Use case Name*** | *Login* | |
| ***Description*** | *Use case* ini menjelaskan bagaimana pengguna dapat masuk ke dalam sistem. | |
| ***Actor*** | *User* | |
| ***Pre-condition*** | Pengguna mengakses alamat website sistem informasi apotek | |
| ***Typical of Events*** | ***Actor Action*** | ***System Response*** |
| 1. Membuka Website | 2. Menampilkan *Form* |
| 3. Pengguna memasukkan *user id* dan *password* | 4. Menampilkan Menu Utama |
| ***Alternate Course*** | Jika *user id* dan *password* salah, maka sistem akan menampilkan kesalahan dan *user* harus memasukkan kembali *user id* dan *password*nya | |
| ***Post-condition*** | Sistem menampilkan menu utama | |

Tabel 4.4 merupakan penjelasan dari *use case scenario* yang dilakukan pengguna dalam melakukan proses *login.*

Tabel 4.5 *Use case* *Scenario* *Logout*

|  |  |  |
| --- | --- | --- |
| ***Use case Name*** | *Logout* | |
| ***Description*** | *Use case* ini menjelaskan bagaimana pengguna keluar dari sistem. | |
| ***Actor*** | *User* | |
| ***Pre-condition*** | Pengguna mengakses alamat website sistem informasi apotek | |
| ***Typical of Events*** | ***Actor Action*** | ***System Response*** |
| 1. Membuka website | 2. Menampilkan *form* *login* |
| 3. Pengguna memasukkan *user id* dan *password* | 4. Menampilkan menu utama |
| 5. Memilih menu "*logout*" | 6. Menampilkan *form* *login* |
| ***Alternate Course*** | - | |
| ***Post-condition*** | Sistem menampilkan *form* *login* | |

Tabel 4.5 merupakan penjelasan dari *use case scenario* yang dilakukan pengguna dalam melakukan proses *logout.*

Tabel 4.6 *Use case* *Scenario* Lihat Menu Utama

|  |  |  |
| --- | --- | --- |
| ***Use case Name*** | Lihat Menu Utama | |
| ***Description*** | *Use case* ini menjelaskan bagaimana pengguna masuk dan melihat menu utama dalam sistem. | |
| ***Actor*** | *User* | |
| ***Pre-condition*** | Pengguna berhasil masuk ke dalam sistem | |
| ***Typical of Events*** | ***Actor Action*** | ***System Response*** |
|  | 1. Menampilkan menu utama di dalam sistem |
| 2. Melihat dan memilih menu. |  |
| ***Alternate Course*** | - | |
| ***Post-condition*** | Sistem menampilkan menu utama | |

Tabel 4.6 merupakan penjelasan dari *use case scenario* yang dilakukan pengguna dalam melakukan proses untuk melihat menu utama*.*

Tabel 4.7 *Use case* *Scenario* Kelola Obat

|  |  |  |
| --- | --- | --- |
| ***Use case Name*** | Kelola Obat | |
| ***Description*** | *Use case* ini menjelaskan bagaimana pengguna mengelola data obat | |
| ***Actor*** | *User* | |
| ***Pre-condition*** | Pengguna berhasil masuk ke dalam sistem dan melihat menu utama | |
| ***Typical of Events*** | ***Actor Action*** | ***System Response*** |
| 1. Memilih menu "Data Obat" | 2. Menampilkan semua data obat. |
| 3. Melakukan manipulasi data (CRUD) data obat yang telah dipilih. | 4. Menjalankan fungsi manipulasi data sesuai dengan perintah pengguna. |
| ***Alternate Course*** | - | |
| ***Post-condition*** | Sistem menampilkan semua data obat dan menampilkan perubahan data obat setelah dilakukan manipulasi. | |

Tabel 4.7 merupakan penjelasan dari *use case scenario* yang dilakukan pengguna dalam melakukan proses untuk mengelola obat*.*

Tabel 4.8 *Use case* *Scenario* Kelola *User*

|  |  |  |
| --- | --- | --- |
| ***Use case Name*** | Kelola *User* | |
| ***Description*** | *Use case* ini menjelaskan bagaimana pengguna mengelola data *user* | |
| ***Actor*** | *User* | |
| ***Pre-condition*** | Pengguna berhasil masuk ke dalam sistem dan melihat menu utama | |
| ***Typical of Events*** | ***Actor Action*** | ***System Response*** |
| 1. Memilih menu "Data *User*" | 2. Menampilkan semua data *user*. |
| 3. Melakukan manipulasi data (CRUD) data *user* yang telah dipilih. | 4. Menjalankan fungsi manipulasi data sesuai dengan perintah pengguna. |
| ***Alternate Course*** | - | |
| ***Post-condition*** | Sistem menampilkan semua data *user* dan menampilkan perubahan data *user* setelah dilakukan manipulasi. | |

Tabel 4.8 merupakan penjelasan dari *use case scenario* yang dilakukan pengguna dalam melakukan proses untuk mengelola *user.*

Tabel 4 9 *Use case* *Scenario* Kelola *Supplier*

|  |  |  |
| --- | --- | --- |
| ***Use case Name*** | Kelola *Supplier* | |
| ***Description*** | *Use case* ini menjelaskan bagaimana pengguna mengelola data *supplier* | |
| ***Actor*** | *User* | |
| ***Pre-condition*** | Pengguna berhasil masuk ke dalam sistem dan melihat menu utama | |
| ***Typical of Events*** | ***Actor Action*** | ***System Response*** |
| 1. Memilih menu "Data *Supplier*" | 2. Menampilkan semua data *supplier*. |
| 3. Melakukan manipulasi data (CRUD) data *supplier* yang telah dipilih. | 4. Menjalankan fungsi manipulasi data sesuai dengan perintah pengguna. |
| ***Alternate Course*** | - | |
| ***Post-condition*** | Sistem menampilkan semua data *supplier* dan menampilkan perubahan data *supplier* setelah dilakukan manipulasi. | |

Tabel 4.9 merupakan penjelasan dari *use case scenario* yang dilakukan pengguna dalam melakukan proses untuk mengelola *supplier.*

Tabel 4.10 *Use case* *Scenario* Kelola Penjualan

|  |  |  |
| --- | --- | --- |
| ***Use case Name*** | Kelola Penjualan | |
| ***Description*** | *Use case* ini menjelaskan bagaimana pengguna mengelola data penjualan | |
| ***Actor*** | *User* | |
| ***Pre-condition*** | Pengguna berhasil masuk ke dalam sistem dan melihat menu utama | |
| ***Typical of Events*** | ***Actor Action*** | ***System Response*** |
| 1. Memilih menu "Data Penjualan" | 2. Menampilkan semua data penjualan. |
| 3. Melakukan manipulasi data (CRUD) data penjualan yang telah dipilih. | 4. Menjalankan fungsi manipulasi data sesuai dengan perintah pengguna. |
| ***Alternate Course*** | - | |
| ***Post-condition*** | Sistem menampilkan semua data penjualan dan menampilkan perubahan data penjualan setelah dilakukan manipulasi. | |

Tabel 4.10 merupakan penjelasan dari *use case scenario* yang dilakukan pengguna dalam melakukan proses untuk mengelola penjualan*.*

Tabel 4.11 *Use case* *Scenario* Kelola Pembelian

|  |  |  |
| --- | --- | --- |
| ***Use case Name*** | Kelola Pembelian | |
| ***Description*** | *Use case* ini menjelaskan bagaimana pengguna mengelola data pembelian | |
| ***Actor*** | *User* | |
| ***Pre-condition*** | Pengguna berhasil masuk ke dalam sistem dan melihat menu utama | |
| ***Typical of Events*** | ***Actor Action*** | ***System Response*** |
| 1. Memilih menu "Data Pembelian" | 2. Menampilkan semua data pembelian. |
| 3. Melakukan manipulasi data (CRUD) data pembelian yang telah dipilih. | 4. Menjalankan fungsi manipulasi data sesuai dengan perintah pengguna. |
| ***Alternate Course*** | - | |
| ***Post-condition*** | Sistem menampilkan semua data pembelian dan menampilkan perubahan data pembelian setelah dilakukan manipulasi. | |

Tabel 4.11 merupakan penjelasan dari *use case scenario* yang dilakukan pengguna dalam melakukan proses untuk mengelola pembelian*.*

Tabel 4.12 *Use case* *Scenario* Melakukan Peramalan

|  |  |  |
| --- | --- | --- |
| ***Use case Name*** | Melakukan Peramalan | |
| ***Description*** | *Use case* ini menjelaskan bagaimana pengguna melakukan peramalan penjualan obat | |
| ***Actor*** | *User* | |
| ***Pre-condition*** | Pengguna berhasil masuk ke dalam sistem dan melihat menu utama | |
| ***Typical of Events*** | ***Actor Action*** | ***System Response*** |
| 1. Memilih menu "Peramalan" | 2. Menampilkan pilihan obat yang ingin diramal. |
|  | 3. Memilih obat yang akan diramal. | 4. Menampilkan semua data peramalan obat yang telah dipilih setiap bulannya dengan menggunakan *exponential smoothing* dengan *alpha* 0,1-0,9. |
|  | 5. Memilih *alpha* terbaik dengan melihat perhitungan nilai MSE terendah. |
| ***Alternate Course*** | - | |
| ***Post-condition*** | Sistem menampilkan hasil ramalan obat dan memilih *alpha* terbaik berdasarkan perhitungan MSE. | |

Tabel 4.12 merupakan penjelasan dari *use case scenario* yang dilakukan pengguna dalam melakukan proses untuk menghitung peramalan*.*

Tabel 4.13 *Use case* *Scenario* Menghitung EOQ

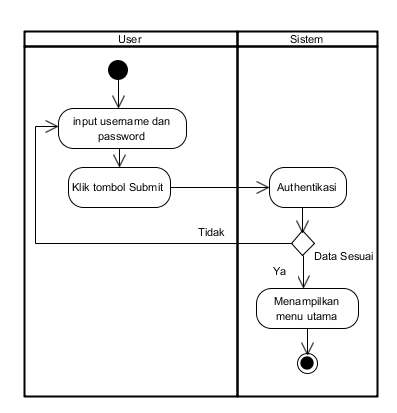
|  |  |  |
| --- | --- | --- |
| ***Use case Name*** | Menghitung EOQ | |
| ***Description*** | *Use case* ini menjelaskan bagaimana pengguna melakukan perhitungan EOQ. | |
| ***Actor*** | *User* | |
| ***Pre-condition*** | Pengguna berhasil masuk ke dalam sistem dan berhasil melakukan peramalan obat | |
| ***Typical of Events*** | ***Actor Action*** | ***System Response*** |
| 1. Memilih "Hitung EOQ" pada menu "Peramalan" | 2. Menampilkan *form* perhitungan EOQ. |
| 3. Memasukkan data yang dibutuhkan untuk melakukan perhitungan EOQ. | 4. Menjalankan operasi perhitungan EOQ. |
| ***Alternate Course*** | - | |
| ***Post-condition*** | Sistem menampilkan hasil perhitungan EOQ. | |

Tabel 4.13 merupakan penjelasan dari *use case scenario* yang dilakukan pengguna dalam melakukan proses untuk menghitung nilai EOQ*.*

### 4.2.4 *Activity diagram*

Agar dapat lebih memahami proses yang terjadi di dalam sistem informasi Apotek Nadia Farma, diperlukan sebuah penggambaran dalam bentuk alur kerja yang lebih detail untuk masing-masing proses yang terjadi di dalamnya. Berikut ini adalah *activity diagram* dari beberapa proses yang terdapat di dalam sistem.

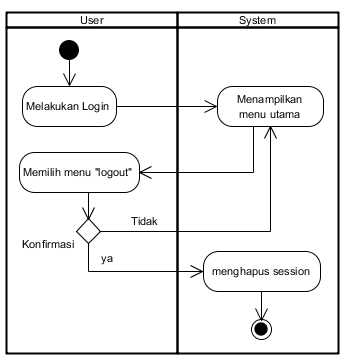
* *Login*



Gambar 4.5 *Activity diagram* *Login*

Gambar 4.5 menjelaskan bahwa hal yang pertama kali dilakukan oleh pengguna adalah dengan memasukkan *username* dan *password* pada *form* yang telah disediakan. Apabila telah selesai memasukkan keduanya, sistem akan melakukan autentikasi yang merupakan pengecekan data di dalam *database*, apakah data yang dimasukkan sesuai atau tidak dengan data yang ada di dalam *database*. Apabila data sesuai, maka sistem akan menampikan menu utama dan membuat *session* baru, apabila data masih belum sesuai, maka pengguna harus memasukkan kembali *username* dan *passwordnya.*

* *Logout*



Gambar 4.6 *Activity diagram* *Logout*

Gambar 4.6 menjelaskan bahwa pengguna sebelumnya harus melakukan proses *login* terlebih dahulu. Setelah *session* baru dimulai dan ingin melakukan *logout*, maka pengguna memilih menu "*logout*". Setelah memilih menu tersebut, maka akan dilakukan konfirmasi kepada pengguna apakah pengguna ingin menghapus *session* atau masih belum ingin menghapusnya. Apabila pengguna memilih ya maka *session* akan terhapus, tetapi apabila pengguna memilih tidak maka sistem akan kembali ke manu utama.

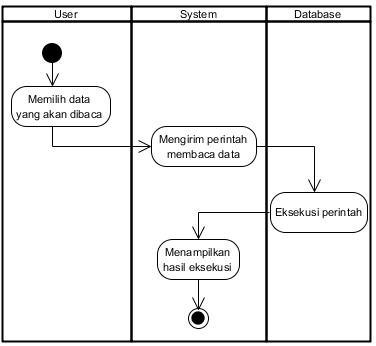
* *Activity diagram* *Create*



Gambar 4.7 *Activity diagram* *Create*

Gambar 4.7 menjelaskan bahwa pada proses ini dimulai dengan pengguna melakukan input data baru ke dalam *database*. Setelah selesai memasukkan data baru yang ingin ditambahkan, maka pengguna menekan tombol "submit" dan setelah itu sistem melakukan validasi. Apabila data tidak valid maka akan keluar informasi bahwa data tidak valid dan kembali lagi ke *form create*. Sedangkan apabila data valid, maka data tersebut akan dikirimkan ke *database* dan dieksekusi untuk melakukan penyimpanan di dalam *database*. Setelah dilakukan eksekusi pembuatan data baru, meka sistem akan menampilkan informasi bahwa data yang telah ditambahkan sukses.

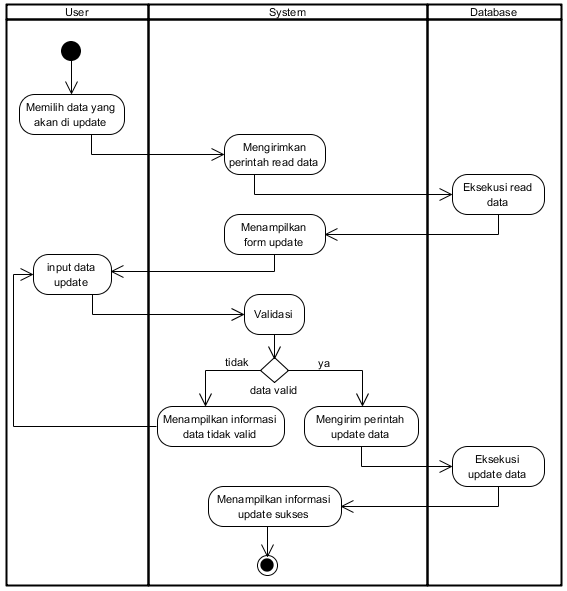
* *Activity diagram* *Read*



Gambar 4.8 *Activity diagram* *Read*

Gambar 4.8 menjelaskan bahwa pada proses *read*, pengguna melakukan pemilihan data yang akan dibaca terlebih dahulu. Lalu sistem akan mengirimkan perintah untuk membaca data yang telah dipilih. Kemudian *database* mengeksekusi perintah untuk membaca data tersebut dan pada akhirnya sistem menampilkan data hasil eksekusi dari *database*.

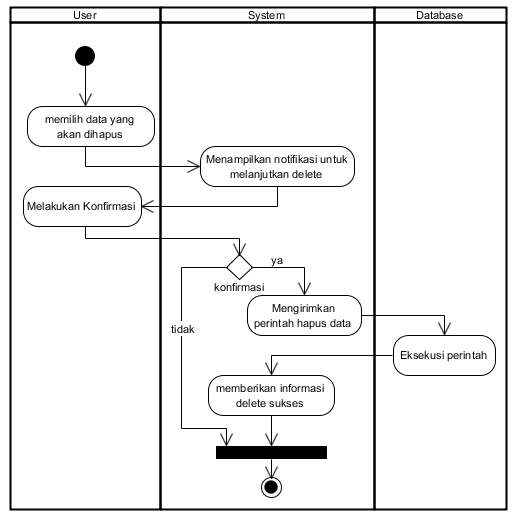
* *Activity diagram* *Update*



Gambar 4.9 *Activity diagram* *Update*

Gambar 4.9 menjelaskan bahwa pada proses ini, pertama-tama pengguna memilih data yang akan di*update*. Selanjutnya, sistem mengirimkan perintah kepada *database* untuk membaca data yang telah dipilih dan kemudian akan dieksekusi oleh *database*. Setelah dilakukan *read* pada data, maka sistem akan menampilkan *form* *update* yang berisi data-data yang telah dieksekusi. Pengguna memasukkan data yang ingin diubah pada *form* *update* tersebut. Setelah data dimasukkan, maka sistem akan melakukan validasi apakah data tersebut valid atau tidak. apabila data tidak valid, maka pengguna harus melakukan penginputan data kembali, sedangkan apabila data tersebut valid maka sistem akan mengirim perintah *update* pada *database*. Setelah dikirimkan perintah, maka *database* melakukan eksekusi data dengan meng*update* data yang telah dipilih. Setelah itu sistem akan menampilkan informasi bahwa *update* berhasil dilakukan.

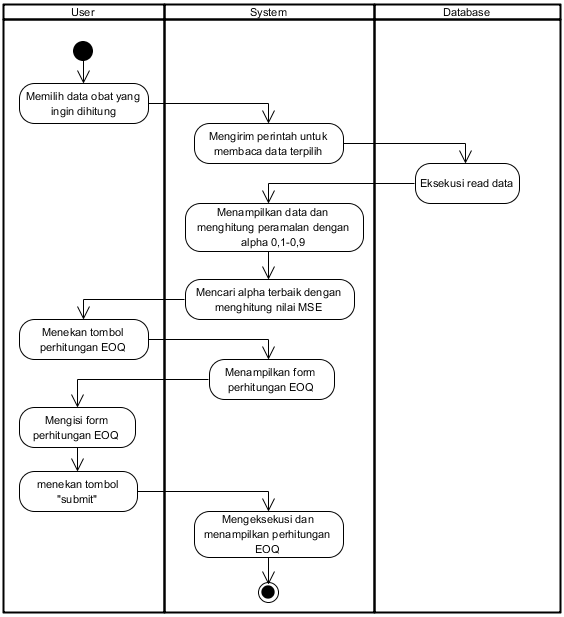
* *Activity diagram* *Delete*



Gambar 4.10 *Activity diagram* *Delete*

Gambar 4.10 menjelaskan bahwa pada proses ini, pengguna harus memilih data yang akan dihapus terlebih dahulu. Setelah data terpilih, maka sistem akan menampilkan notifikasi untuk melanjutkan proses yang kemudian akan dilakukan konfirmasi oleh pengguna. Apabila pengguna memilih tidak, maka tidak akan terjadi apa apa pada data, sedangkan apabila pengguna memilih ya, sistem akan mengirimkan perintah hapus data kepada *database* dan perintah tersebut dieksekusi oleh *database*. Setelah melakukan penghapusan data di dalam *database*, maka sistem akan menampilkan informasi bahwa data yang telah terpilih tersebut sudah dihapus.

* *Activity diagram* Perhitungan EOQ

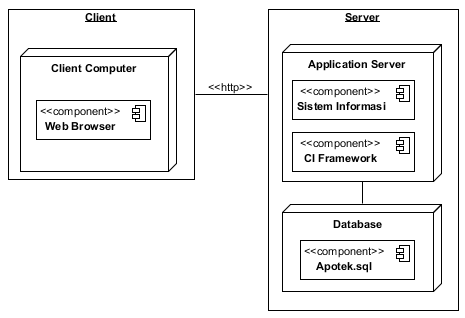


Gambar 4.11 *Activity diagram* EOQ

Gambar 4.11 menjelaskan bahwa pada proses ini, hal pertama yang harus dilakukan adalah memilih data obat yang ingin dihitung. Sistem kemudian akan mengirimkan perintah kepada *database* untuk membaca data yang telah dipilih oleh pengguna dan selanjutnya akan dieksekusi oleh *database*. Setelah itu, sistem akan menampilkan data dan menghitung peramalan menggunakan *exponential smoothing* dengan *alpha* 0,1-0,9. Setelah terlihat peramalan, maka sistem akan mencari *alpha* terbaik dengan menggunakan perhitungan *Mean Squared Error* (MSE). Untuk melakukan perhitungan EOQ, maka pengguna harus menekan tombol perhitungsan EOQ. Setelah itu, sistem akan menampilkan *form* untuk perhitungan EOQ. Pengguna selanjutnya mengisi *form* yang telah ditampilkan untuk melakukan perhitungan dan menekan tombol submit jika telah selesai melakukan pengisian. Setelah itu, sistem akan menghitung jumlah EOQ dan menampilkan perhitungan yang telah dilakukan.

### 4.2.5 Deployment Diagram

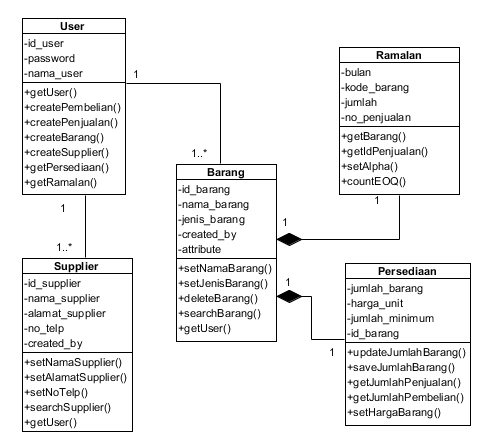
*Deployment Diagram* adalah sebuah diagram yang menggambarkan topologi komponen sistem ketika dilakukan implementasi.

**

Gambar 4.12 *Deployment Diagram*

Gambar 4.12 menampilkan topologi komponen dari sistem yang digammbarkan di dalam sebuah *deployment diagram.*

### 4.2.6 *Class diagram*

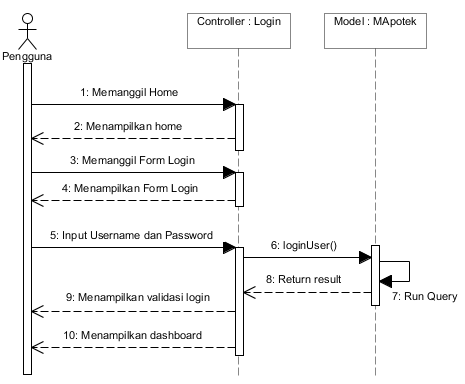


Gambar 4.13 *Class diagram* dari sistem

Gambar 4.13 menampilkan *class diagram* dari sistem yang akan dibuat. Apabila *user* melakukan pembelian, maka *class* yang terlibat adalah *class* *user*, *supplier*, barang, dan persediaan. *User* membeli barang dari *supplier*, kemudian barang tersebut akan ditambahkan pada persediaan di apotek. Apabila *user* ingin melakukan penjualan, maka *class* yang terlibat adalah *class* *user*, barang, persediaan, dan peramalan. *User* menjual barang, barang tersebut diambil dari persediaan yang ada pada apotek dan jumlah penjualannya digunakan untuk melakukan peramalan.

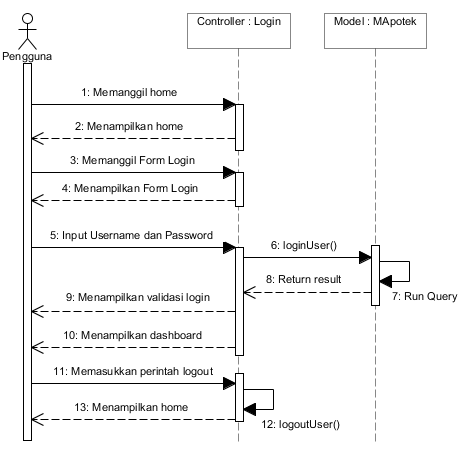
### 4.2.7 Sequence Diagram

Berikut merupakan *sequence diagram* dari sistem yang akan dibuat.



Gambar 4.14 Sequence Diagram *Login*

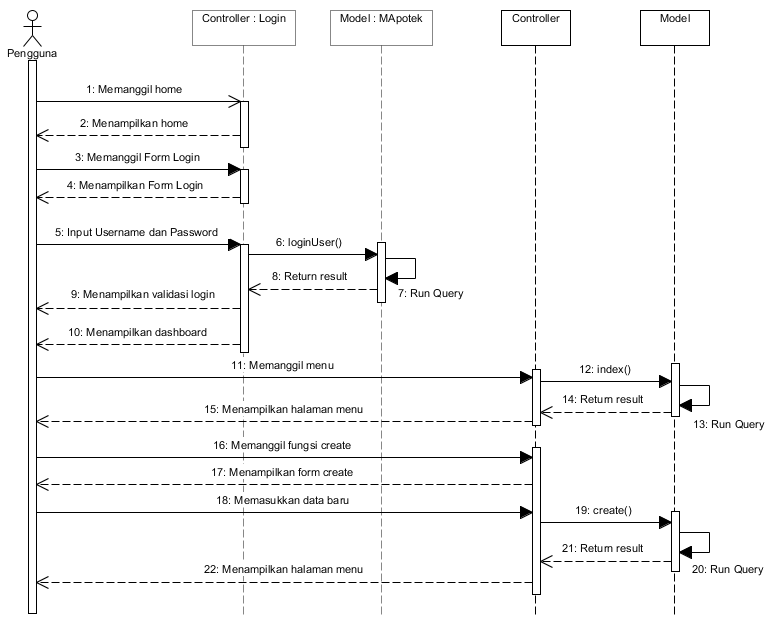
Gambar 4.14 merupakan *sequence diagram* dari *login* pengguna. Proses ini menggunakan *controller Login*.php sebagai logika kontrol dan MApotek.php untuk melakukan *query*.



Gambar 4.15 Sequence Diagram Logut

Gambar 4.15 merupakan *sequence diagram* dari *logout* pengguna. Proses ini menggunakan *controller Login*.php sebagai logika kontrol dan MApotek.php untuk melakukan *query*.

Pada dasarnya, pengelolaan persediaan yang dilakukan pada sistem yang akan dibuat merupakan proses CRUD (*Create, Read, Update,Delete*). Maka, untuk *sequence diagram* pengelolaan data oleh pengguna hanya akan ditampilkan secara general dengan detail *controller* dan *model* yang digunakan oleh masing-masing kategori.



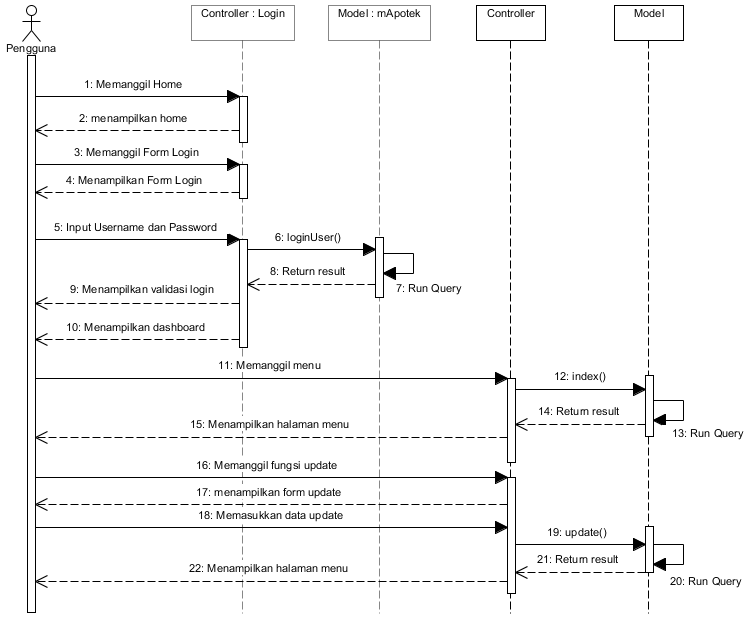
Gambar 4.16 Sequence Diagram *Create*

Gambar 4.16 merupakan *sequence diagram* dari *create* yang dilakukan pengguna. Proses ini menggunakan *controller* controllers/*User*.php sebagai logika kontrol dan model/*User*\_model.php untuk melakukan *query*. *Sequence diagram* ini akan memanggil laman pembuatan data baru sebagai hasil dari proses. Berikut adalah tabel model dan action untuk kategori yang berbeda. *Index()* merupakan fungsi untuk memanggil *home*, sedangkan *create()* adalah fungsi untuk menambah data baru*.*

Tabel 4.14 Tabel Model, Controller, dan Fungsi *Create*

|  |  |  |
| --- | --- | --- |
| **Model** | **Controller** | **Function** |
| Models/Barang\_model.php | Controllers/Barang.php | *Create*() |
| Models/*Supplier*\_model.php | Controllers/*Supplier*s.php | *Create*() |
| Models/Penjualan\_model.php | Controllers/Penjualan.php | *Create*() |
| Models/Pembelian\_model.php | Controllers/Pembelian.php | *Create*() |

Tabel 4.14 merupakan tabel yang menunjukkan *model* dan *controller* dari sistem yang menggunakan fungsi *create.*



Gambar 4.17 Sequence diagram *update*

Gambar 4.17 merupakan *sequence diagram* dari *update* yang dilakukan pengguna. Proses ini menggunakan *controller* controllers/*User*.php sebagai logika kontrol dan model/*User*\_model.php untuk melakukan *query*. *Sequence diagram* ini akan memanggil laman peng-*update-*an data sebagai hasil dari proses. Berikut adalah tabel model dan action untuk kategori yang berbeda. *Index()* merupakan fungsi untuk memanggil *home*, sedangkan *update()* adalah fungsi untuk merubah data yang dipilih*.*

Tabel 4.15 Model, Controller, dan Fungsi dari *Update*

|  |  |  |
| --- | --- | --- |
| **Model** | **Controller** | **Function** |
| Models/Barang\_model.php | Controllers/Barang.php | *Update*() |
| Models/*Supplier*\_model.php | Controllers/*Supplier*s.php | *Update*() |
| **Model** | **Controller** | **Function** |
| Models/Penjualan\_model.php | Controllers/Penjualan.php | *Update*() |
| Models/Pembelian\_model.php | Controllers/Pembelian.php | *Update*() |

Tabel 4.15 merupakan tabel yang menunjukkan *model* dan *controller* dari sistem yang menggunakan fungsi *update.*



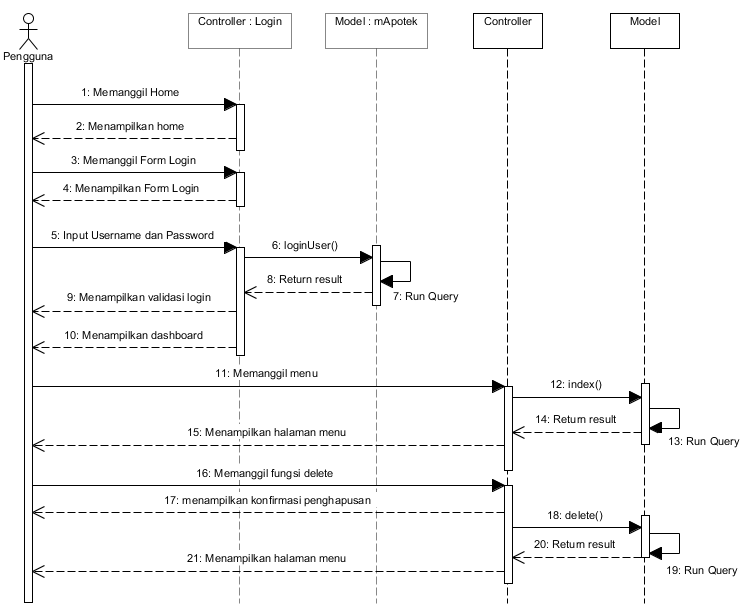
Gambar 4.18 Sequence Diagram *read*

Gambar 4.18 merupakan *sequence diagram* dari *read* yang dilakukan pengguna. Proses ini menggunakan *controller* controllers/*User*.php sebagai logika kontrol dan model/*User*\_model.php untuk melakukan *query*. *Sequence diagram* ini akan memanggil laman untuk melihat rincian data sebagai hasil dari proses. Berikut adalah tabel model dan action untuk kategori yang berbeda. *Index()* merupakan fungsi untuk memanggil *home*, sedangkan *read()* adalah fungsi untuk melihat detail dari data yang telah dipilih*.*

Tabel 4.16 Model, Controller, dan Fungsi *Read*

|  |  |  |
| --- | --- | --- |
| **Model** | **Controller** | **Functio n** |
| Models/Barang\_model.php | Controllers/Barang.php | *Read*() |
| Models/*Supplier*\_model.php | Controllers/*Supplier*s.php | *Read*() |
| Models/Penjualan\_model.php | Controllers/Penjualan.php | *Read*() |
| Models/Pembelian\_model.php | Controllers/Pembelian.php | *Read*() |

Tabel 4.16 merupakan tabel yang menunjukkan *model* dan *controller* dari sistem yang menggunakan fungsi *read.*



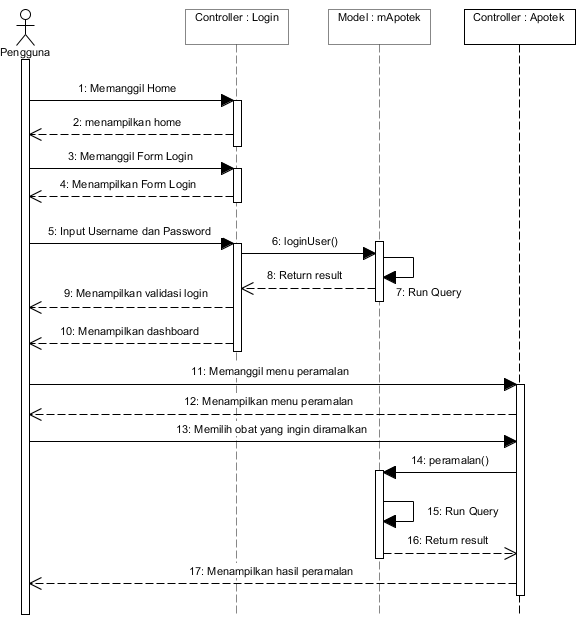
Gambar 4.19 Sequence Diagram *Delete*

Gambar 4.19 merupakan *sequence diagram* dari *delete* yang dilakukan pengguna. Proses ini menggunakan *controller* controllers/*User*.php sebagai logika kontrol dan model/*User*\_model.php untuk melakukan *query*. *Sequence diagram* ini akan memanggil laman untuk menghapus data sebagai hasil dari proses. Berikut adalah tabel model dan action untuk kategori yang berbeda. *Index()* merupakan fungsi untuk memanggil *home*, sedangkan *delete()* adalah fungsi untuk menghapus data yang telah dipilih*.*

Tabel 4.17 Model, Controller, dan Fungsi *Delete*

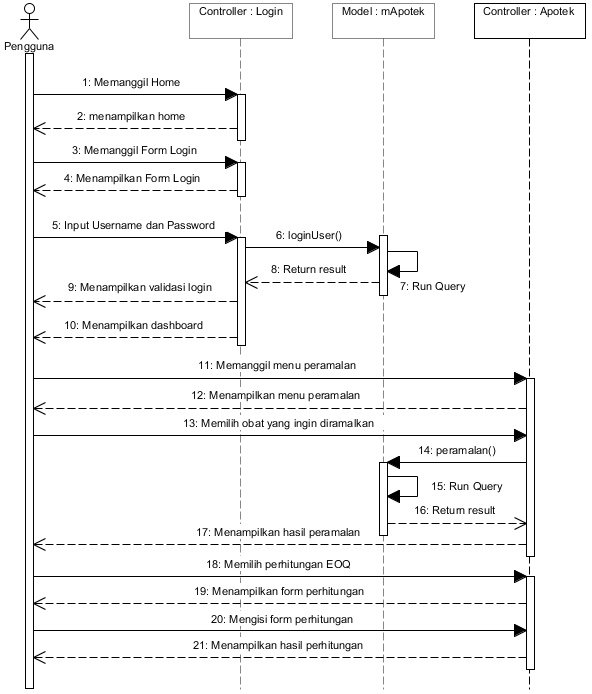
|  |  |  |
| --- | --- | --- |
| **Model** | **Controller** | **Function** |
| Models/Barang\_model.php | Controllers/Barang.php | *Delete*() |
| Models/*Supplier*\_model.php | Controllers/*Supplier*s.php | *Delete*() |
| Models/Penjualan\_model.php | Controllers/Penjualan.php | *Delete*() |
| Models/Pembelian\_model.php | Controllers/Pembelian.php | *Delete*() |

Tabel 4.17 merupakan tabel yang menunjukkan *model* dan *controller* dari sistem yang menggunakan fungsi *create.*



Gambar 4.20 Sequence Diagram Peramalan

Gambar 4.20 merupakan *sequence diagram* dari peramalan yang dilakukan sistem. Proses ini menggunakan *controller* controllers/Apotek.php sebagai logika kontrol dan model/MApotek.php untuk melakukan *query*. *Sequence diagram* ini akan memanggil laman untuk melakukan peramalan data sebagai hasil dari proses. Peramalan() adalah fungsi yang digunakan untuk menghitung hasil ramalan dari obat yang telah dipilih.

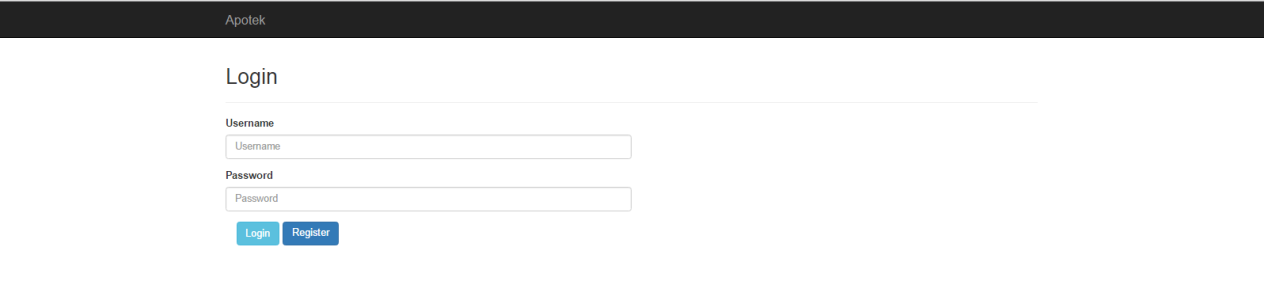


Gambar 4.21 Sequence Diagram EOQ

Gambar 4.21 merupakan *sequence diagram* dari perhitungan EOQ yang dilakukan sistem. Proses ini menggunakan *controller* controllers/Apotek.php sebagai logika kontrol dan model/MApotek.php untuk melakukan *query*. *Sequence diagram* ini akan memanggil laman untuk melakukan perhitungan EOQ sebagai hasil dari proses. Peramalan() adalah fungsi yang digunakan untuk menghitung hasil ramalan dari obat yang telah dipilih.

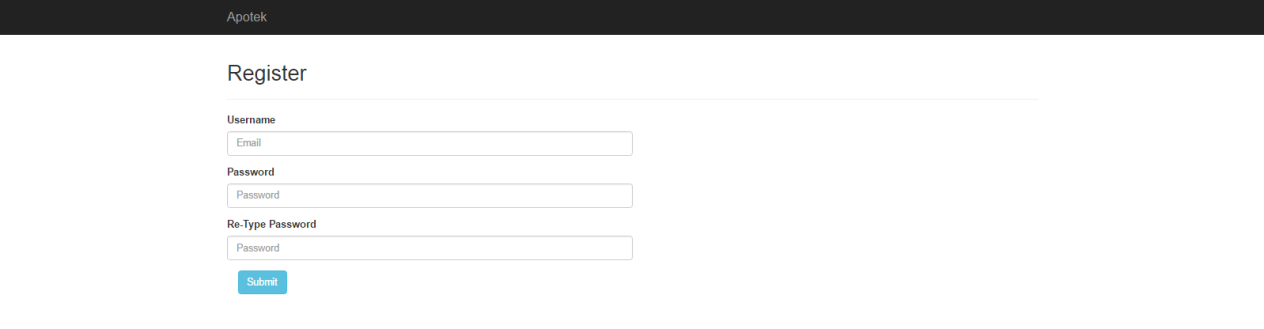
## 4.3 *User* Interface Sistem

Pada tahapan ini penulis akan menjelaskan tampilan *user interface* dari sistem dengan menggunakan bahasa pemrograman berbasis web yaitu PHP yang menggunakan *framework CodeIgniter*. Berikut ini merupakan tampilan *user interface* dari sistem:



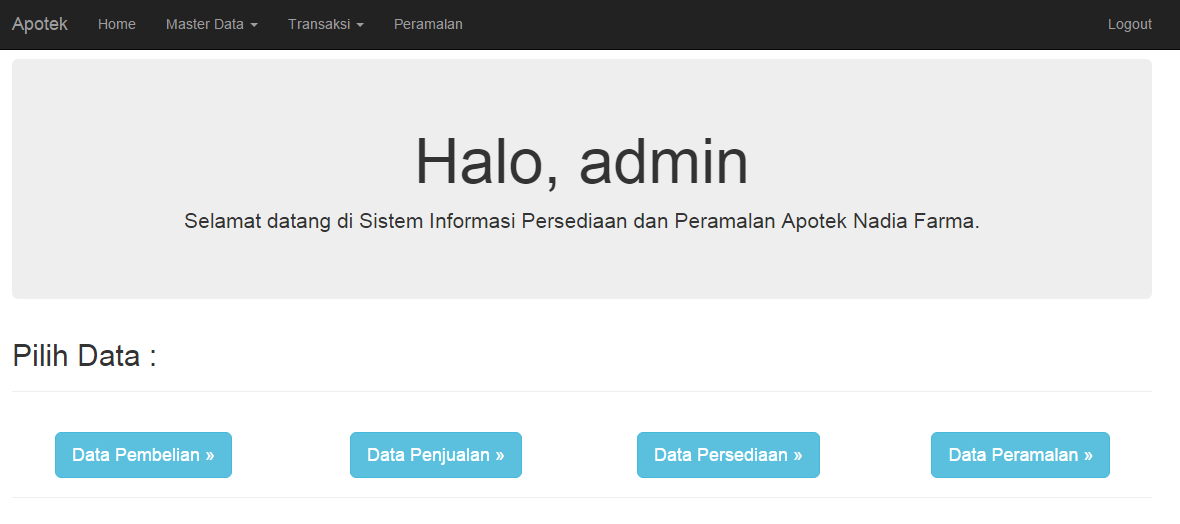
Gambar 4.22 Halaman *Login*

Gambar 4.22 merupakan halaman login yang akan muncul saat sistem dijalankan oleh pengguna. Pengguna harus menginputkan *username* dan *password* yang valid dan sudah terdaftar di dalam *database* agar dapat masuk ke dalam sistem. Jika tidak sesuai maka sistem akan menampilkan notifikasi bahwa nama *username* dan *password* salah atau belum diisi dan sistem akan menampilkan halaman *login* dan meminta pengguna untuk memasukkan data kembali. Pada halaman ini terdapat tombol *register* yang digunakan untuk mengakses menu *registrasi user* apabila ada pengguna baru yang ingin mendaftar masuk ke dalam sistem.



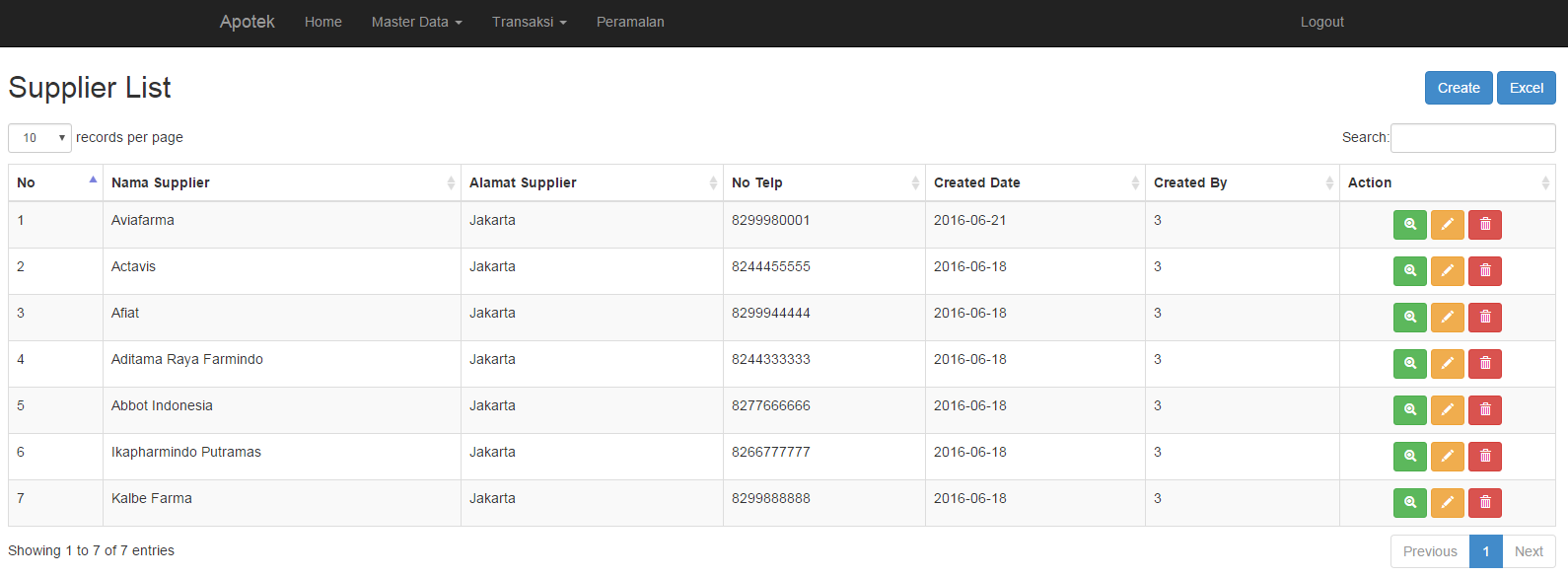
Gambar 4.23 Halaman Registrasi *User*

Gambar 4.23 merupakan halaman registrasi *user* yang digunakan untuk menambahkan pengguna ke dalam sistem. Pengguna harus memasukkan *Username* dan *Password* yang baru agar dapat membuat data baru. Setelah menginputkan data, pengguna menekan tombol *submit* dan data akan tersimpan di dalam *database.* Setelah itu, sistem akan kembali ke halaman *login* kembali.



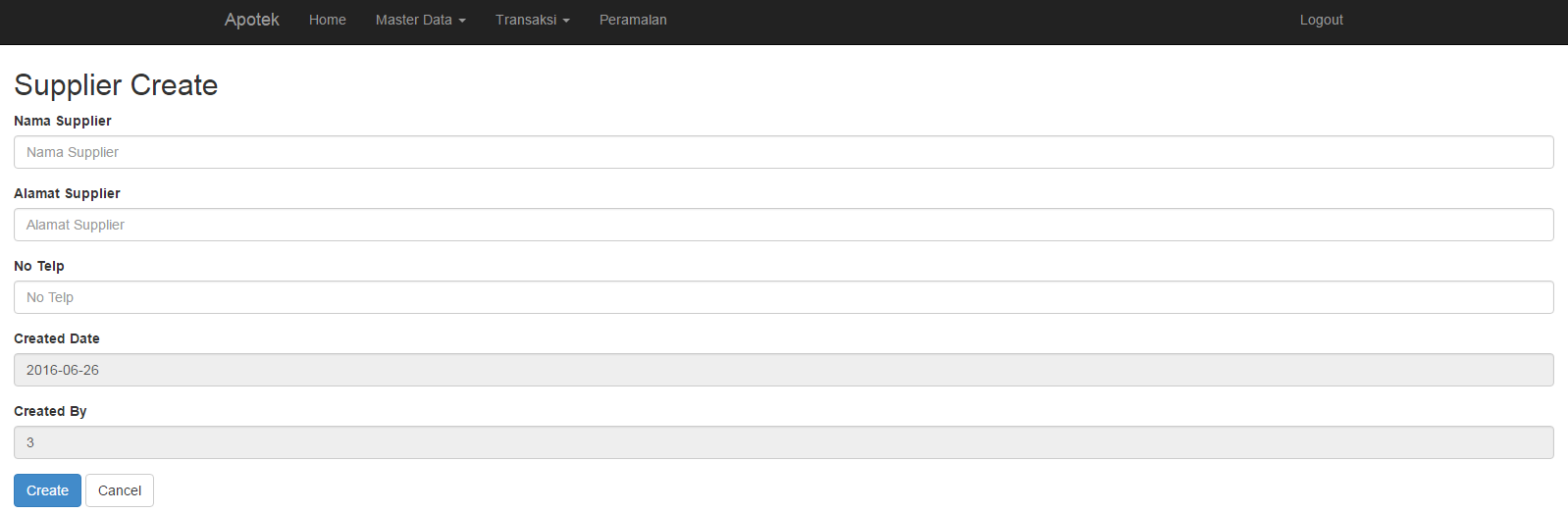
Gambar 4.24 Halaman Home

Gambar 4.24 merupakan halaman home dari sistem. Setelah pengguna berhasil masuk ke dalam sistem maka sistem akan menampilkan halaman *home* dan *menu* yang dapat diakses oleh pengguna. Pada halaman ini terdapat bar navigasi yang terletak pada bagian atas laman dan berisi menu yang digunakan untuk mengakses laman lain yang berisi informasi dan fungsi untuk pengguna. Bar navigasi menu ini memiliki lima menu utama yaitu home, master data, transaksi, permalan, dan *logout*. terdapat dua buah *dropdown menu,* yaitu menu master data yang di dalamnya terdapat menu data obat dan data *supplier*, dan menu transaksi yang di dalamnya terdapat menu pembelian, penjualan, dan persediaan.



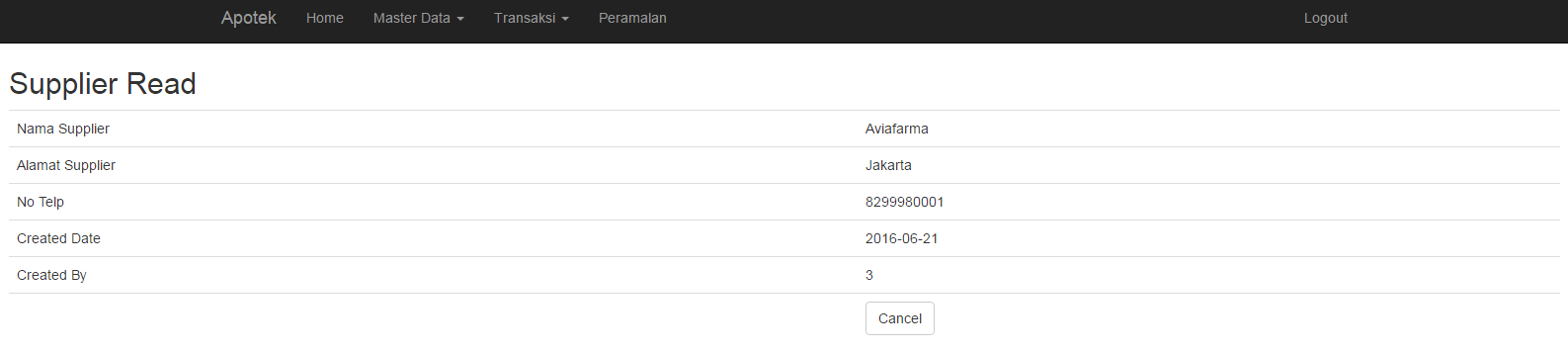
Gambar 4.25 Halaman Data *Supplier*

Gambar 4.25 merupakan halaman data *supplier* yang menampilkan tentang seluruh informasi yang berhubungan dengan *supplier*. Pengguna dapat melakukan pencarian, melihat detil tentang barang, melakukan perubahan, membuat data baru, mengekspor data *supplier* ke *Microsoft® Office™ Excel*, dan menghapus data *supplier*.



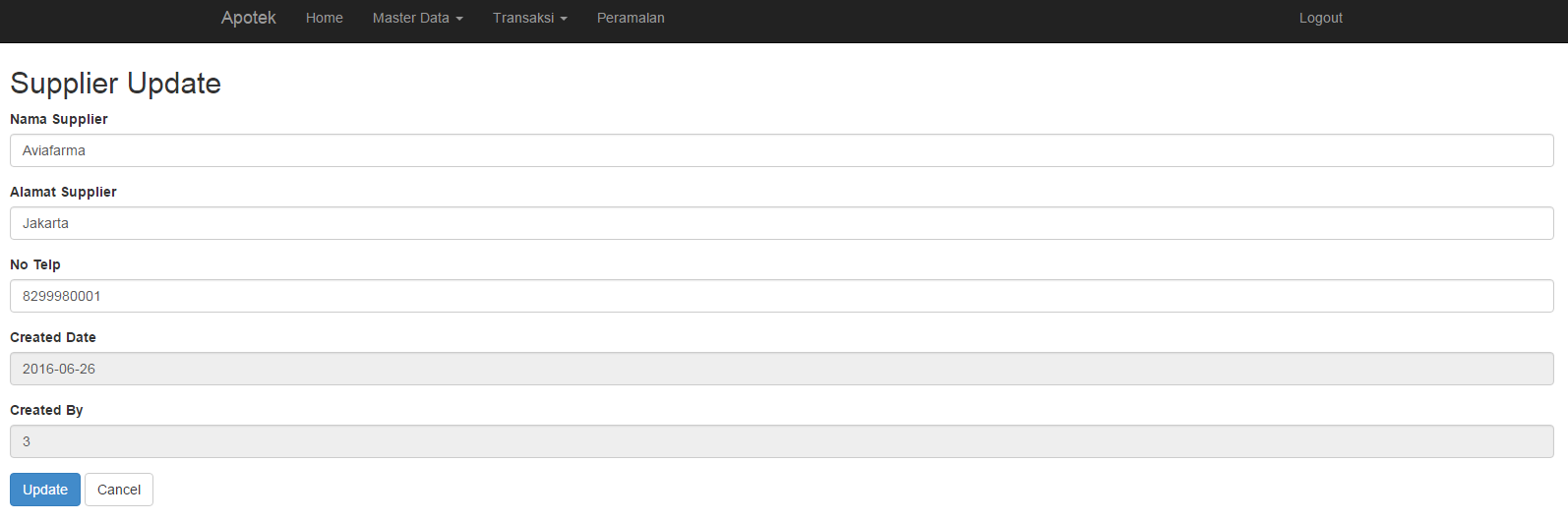
Gambar 4.26 Halaman Buat *Supplier* Baru

Gambar 4.26 merupakan halaman yang digunakan untuk menampilkan *form* yang digunakan untuk menginput data *supplier* yang ingin dimasukkan ke dalam *database.* Terdapat tombol *create* yang digunakan untuk mengirim dan melakukan validasi data yang telah dimasukkan. Apabila data tersebut telah lolos validasi, maka data akan disimpan ke dalam *database*. Sedangkan tombol cancel digunakan apabila pengguna batal membuat data baru.



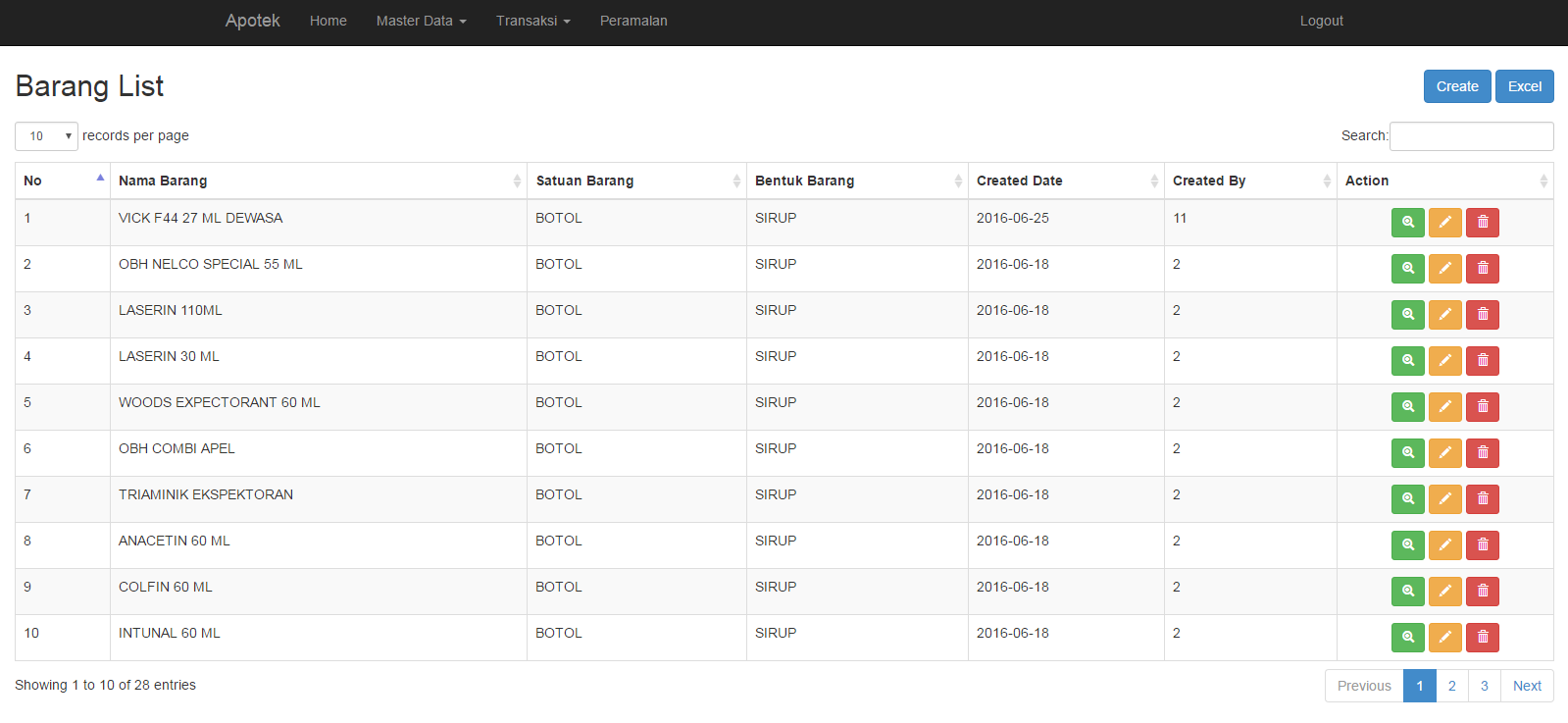
Gambar 4.27 Halaman Detail Data *Supplier*

Gambar 4.27 merupakan halaman yang digunakan untuk membaca detail data *supplier* yang telah dipilih. Data yang telah dipilih pada halaman data *supplier* kemudian ditampilkan kembali pada halaman baru dan ditampilkan detailnya pada halaman ini. Terdapat tombol cancel yang digunakan apabila pengguna ingin kembali ke halaman data *supplier*.



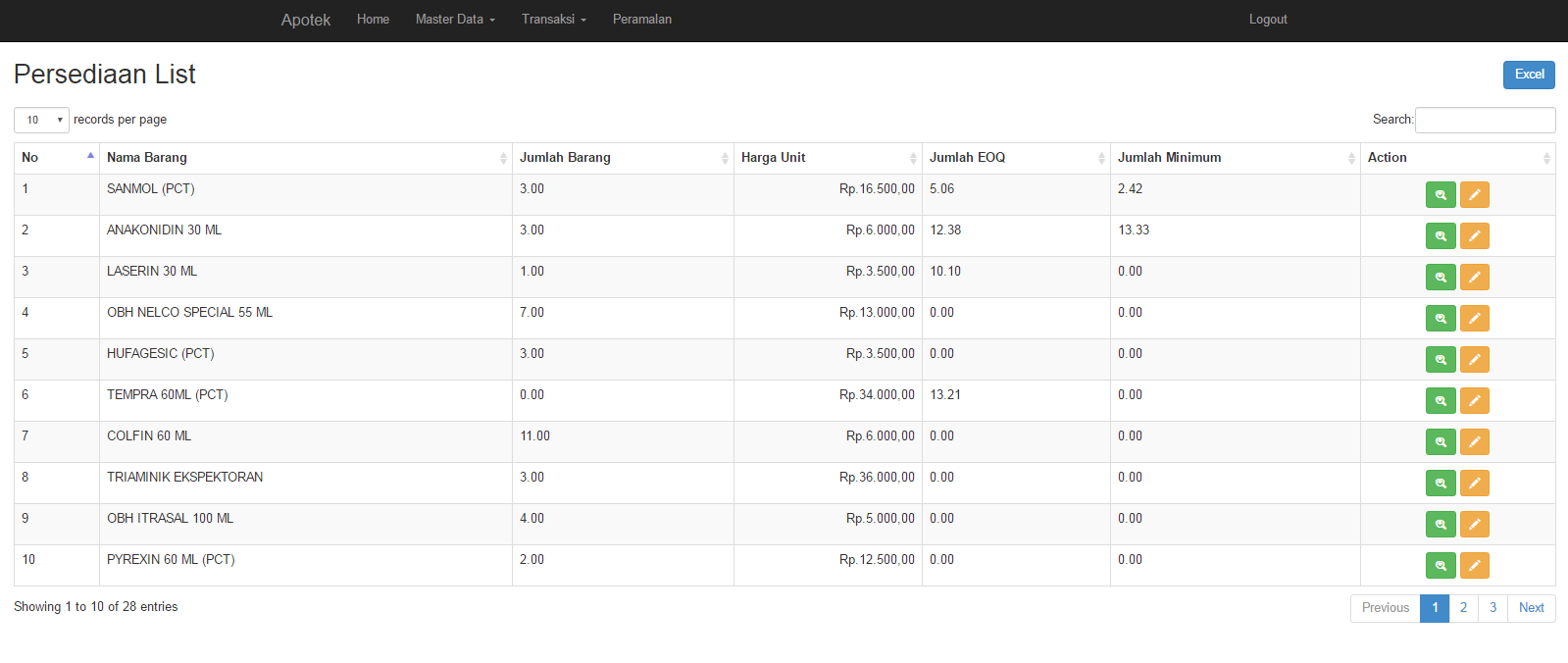
Gambar 4.28 Halaman Ubah *Supplier*

Gambar 4.28 merupakan halaman yang digunakan untuk mengubah data *supplier* yang sudah tersimpan di dalam *database*. Sebelum dmasuk ke halaman ini, pengguna memilih data *supplier* yang akan diubah pada halaman data *supplier* dan memilih tombol ubah data yang berwarna kuning. Setelah itu akan keluar halaman *supplier* dan terdapat *form* untuk melakukan perubahan data. Setelah melakukan perubahan, maka pengguna menekan tombol *update*, dan apabila pengguna batal melakukan perubahan, pengguna menekan tombol *update*.



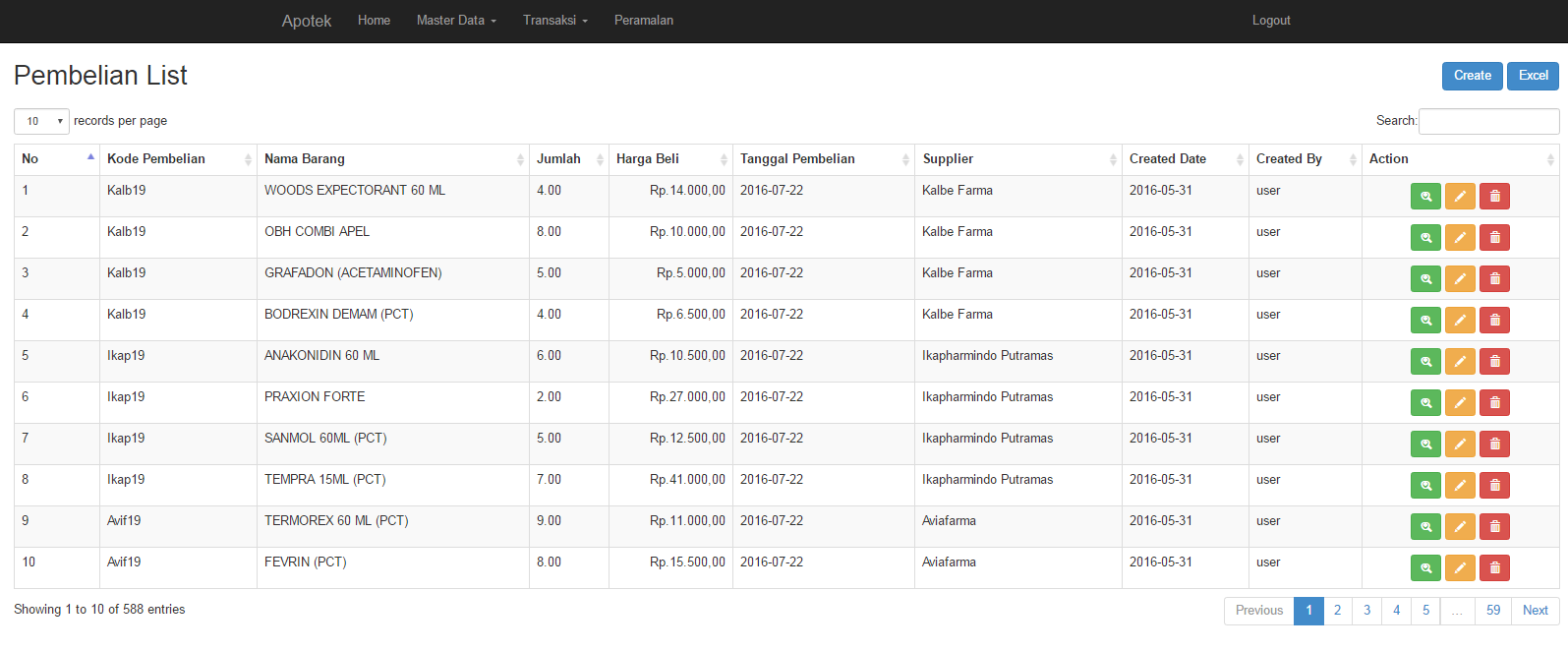
Gambar 4.29 Halaman Data Barang

Gambar 4.29 merupakan halaman data barang yang menampilkan tentang seluruh informasi obat yang tersimpan di dalam *database*. Pengguna dapat melakukan pencarian, melihat detail tentang barang, melakukan perubahan, membuat data baru, mengekspor data barang ke *Microsoft® Office™ Excel*, dan menghapus data barang. Halaman detail barang, perubahan barang, dan pembuatan data baru memiliki tampilan yang serupa dengan yang ada pada halaman data *supplier*, hanya berbeda pada isi datanya.



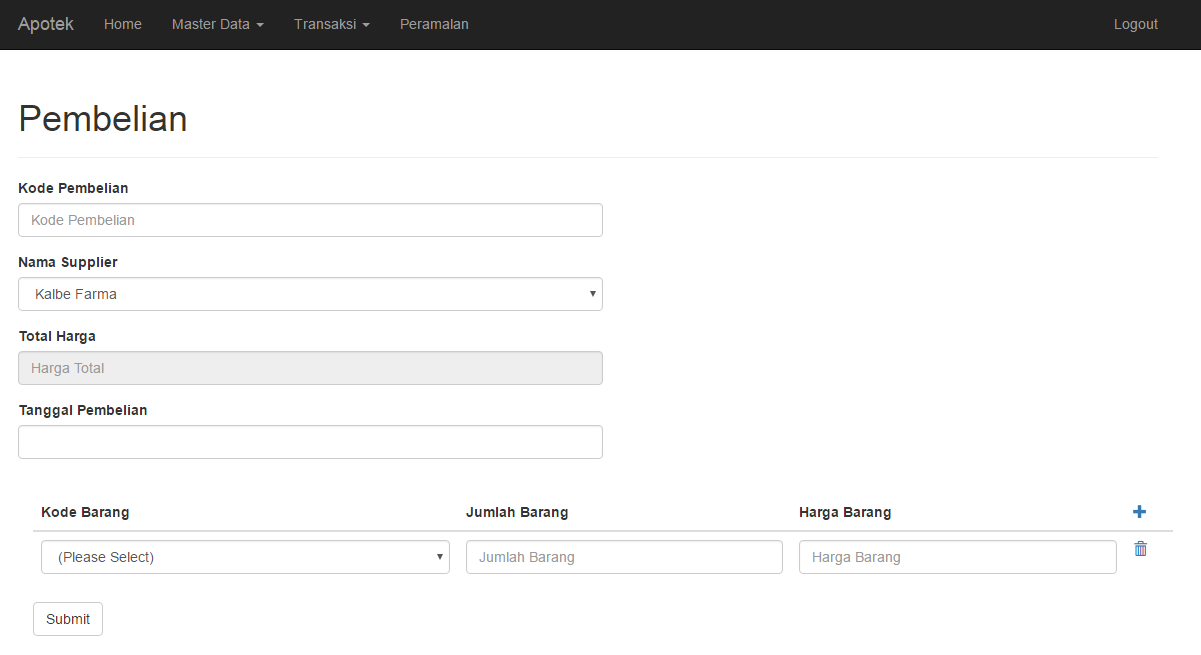
Gambar 4.30 Halaman Data Persediaan

Gambar 4.30 merupakan halaman data persediaan yang menampilkan tentang seluruh informasi persediaan yang tersimpan di dalam *database*. Pengguna dapat melakukan pencarian, melihat detail tentang persediaan, mengekspor data persediaan ke excel, dan menghapus data persediaan. Halaman detail persediaan, dan mengubah data persediaan memiliki tampilan yang serupa dengan yang ada pada halaman data *supplier*, hanya berbeda pada isi datanya. Pada halaman data persediaan tidak ada tombol untuk membuat data persediaan, karena data persediaan akan ada yang baru jika ada data obat yang baru dibuat.



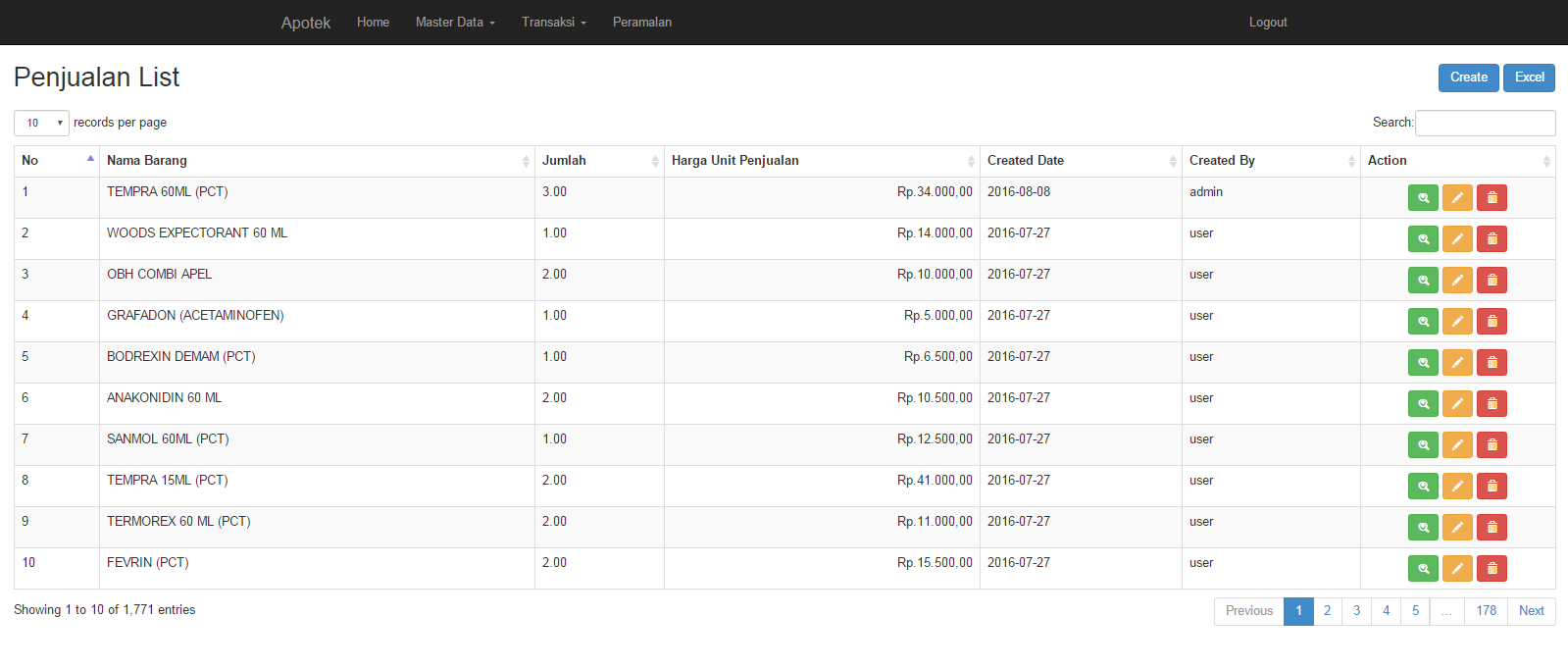
Gambar 4.31 Halaman Data Pembelian

Gambar 4.31 merupakan halaman data pembelian yang menampilkan tentang seluruh informasi pembelian yang tersimpan di dalam *database*. Pengguna dapat melakukan pencarian, melihat detail tentang informasi pembelian, mengekspor data pembelian ke excel, dan menghapus data pembelian. Halaman detail pembelian, dan mengubah data pembelian memiliki tampilan yang serupa dengan yang ada pada halaman data *supplier*, hanya berbeda pada isi datanya.



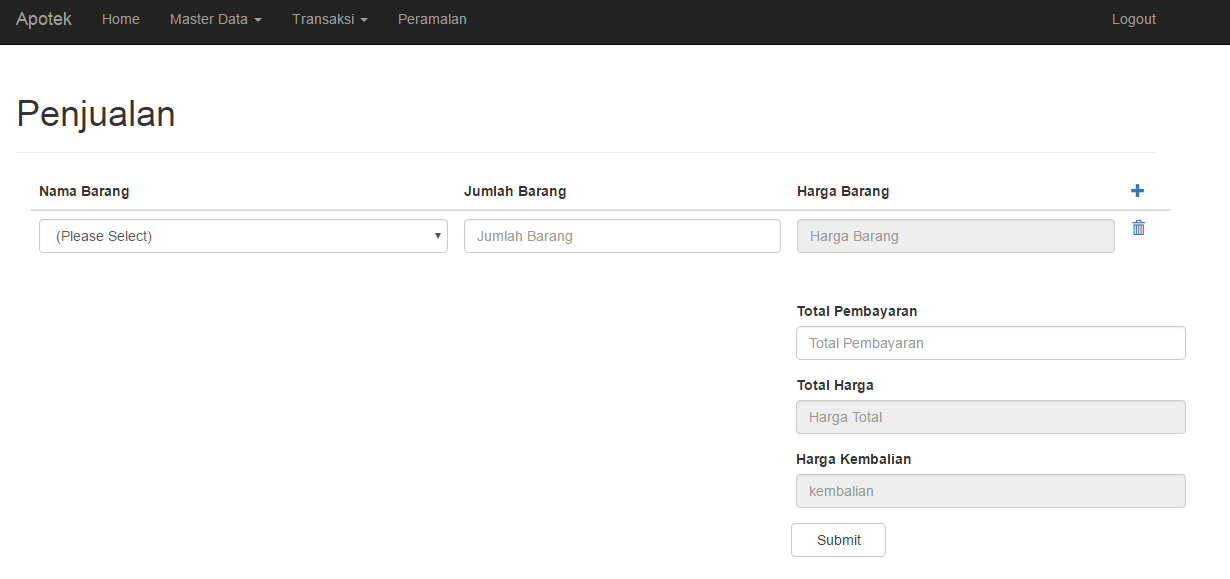
Gambar 4.32 Halaman Tambah Pembelian

Gambar 4.32 merupakan halaman yang digunakan untuk menambah data pembelian apabila dilakukan transaksi pembelian. Halaman ini berisi *form* yang digunakan untuk memasukkan data pembelian ke dalam *database*. Terdapat tombol plus (+) yang dapat digunakan untuk menambah *item* apabila dalam satu pembelian terdapat lebih dari satu barang. Setelah semua data telah diisi, kemudian tekan tombol submit yang digunakan untuk mengirim data ke dalam *database*.



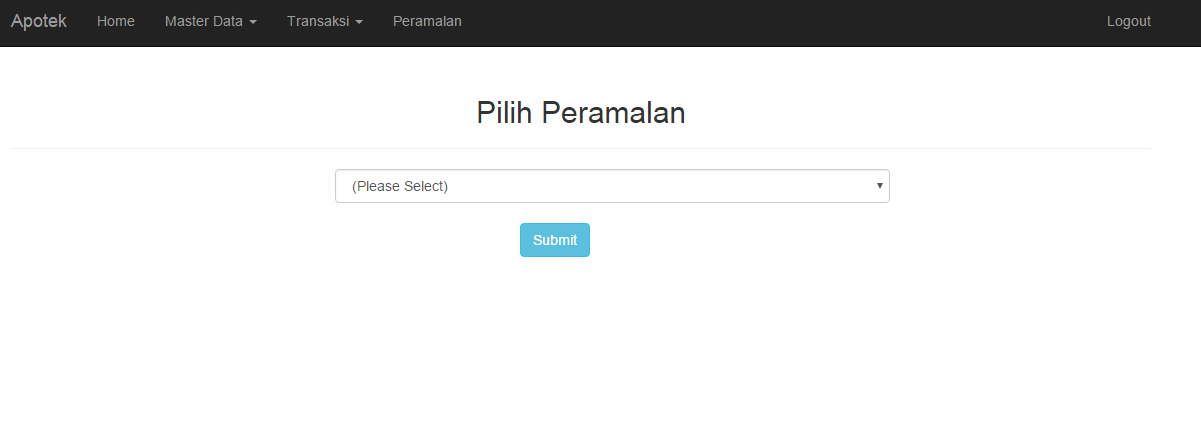
Gambar 4.33 Halaman Data Penjualan

Gambar 4.33 merupakan halaman data penjualan yang menampilkan tentang seluruh informasi penjualan yang tersimpan di dalam *database*. Pengguna dapat melakukan pencarian, melihat detail tentang informasi penjualan, mengekspor data penjualan ke excel, dan menghapus data penjualan. Halaman detail penjualan, dan mengubah data penjualan memiliki tampilan yang serupa dengan yang ada pada halaman data *supplier*, hanya berbeda pada isi datanya.



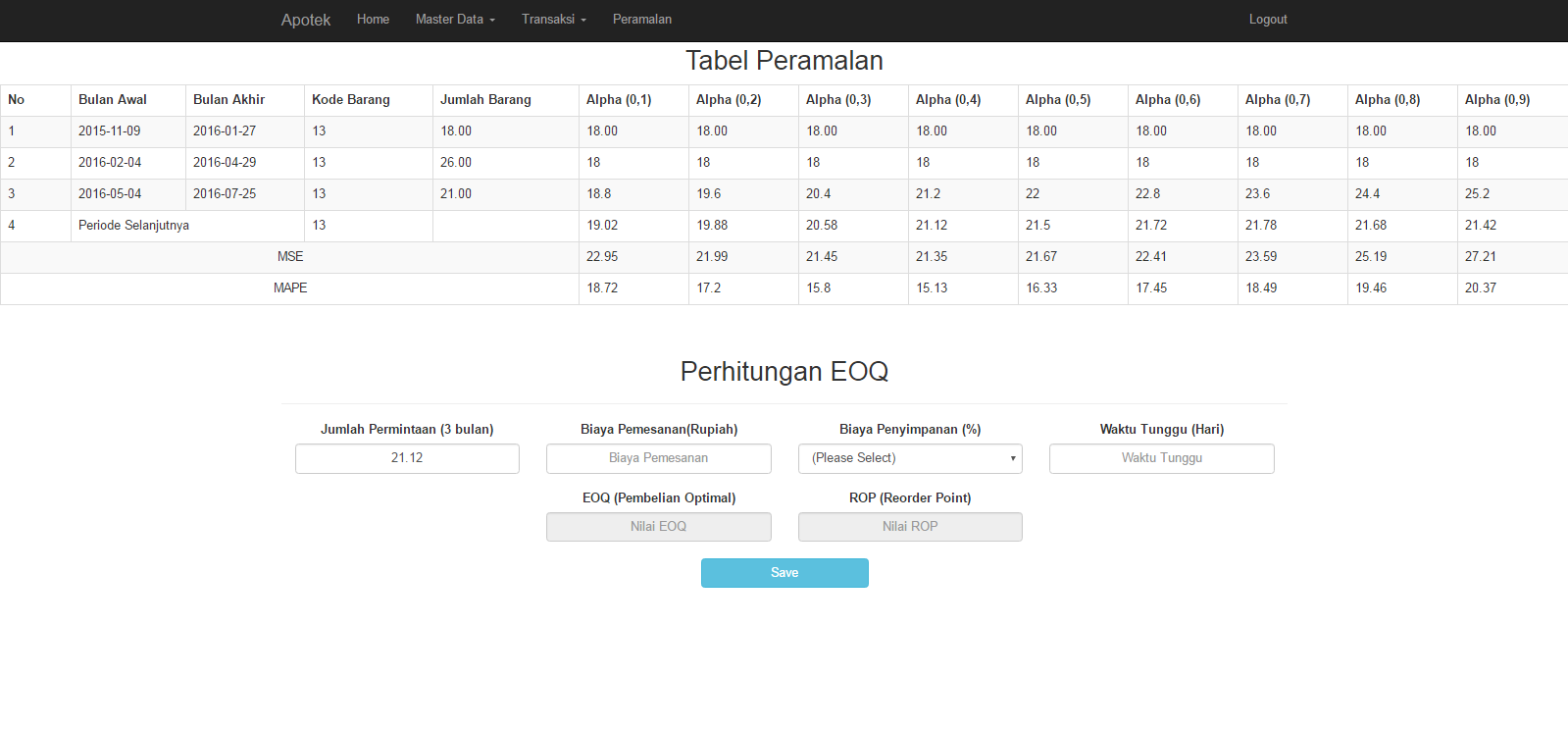
Gambar 4.34 Halaman Tambah Penjualan

Gambar 4.34 menapilkan halaman yang digunakan untuk menambah data penjualan apabila dilakukan transaksi penjualan yang baru. Halaman ini berisi *form* yang digunakan untuk memasukkan data penjualan ke dalam *database*. Terdapat tombol plus (+) yang dapat digunakan untuk menambah *item* apabila dalam satu penjualan terdapat lebih dari satu barang. Setelah semua data telah diisi, kemudian tekan tombol submit yang digunakan untuk mengirim data ke dalam *database*.



Gambar 4.35 Halaman Pilih Peramalan

Gambar 4.35 menampilkan halaman yang merupakan halaman awal sebelum melakukan peramalan. Pada halaman ini pengguna memilih obat yang ingin dilakukan permalan. Setelah obat dipilih, pengguna menekan tombol submit untuk melanjutkan ke halaman permalan.



Gambar 4.36 Halaman Peramalan dan EOQ

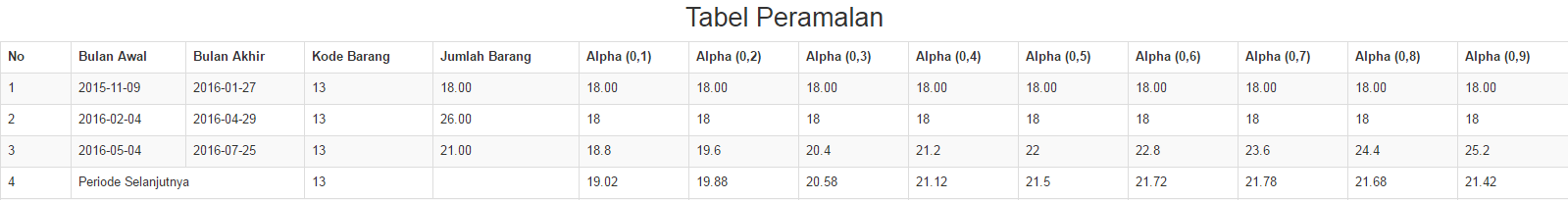
Setelah memilih obat yang ingin dilakukan peramalan, maka akan keluar halaman peramalan dan juga perhitungan EOQ, seperti yang terlihat pada gambar 4.36. Terdapat tabel peramalan pada halaman ini yang berisikan periode, kode barang, jumlah barang dalam satu periode, dan perhitungan *exponential smoothing* dari *alpha* 0,1-0,9. Setelah mengetahui hasil permalan dari masing-masing *alpha*, kemudian dipilih *alpha* terbaik dengan menggunakan MAPE dan MSE. Pada halaman ini juga terdapat perhitungan EOQ, yang berisikan *form* yang digunakan untuk melakukan perhitungan EOQ. Setelah memasukkan data di dalam *form*, maka nilai EOQ dan nilai ROP akan diketahui. Pengguna menekan tombol save apabila ingin menyimpan hasil perhitungan EOQ dan ROP ke dalam *database*.

## 4.4 Pengujian Sistem

Pengujian sistem dilakukan dengan menggunakan metode *blackbox testing* sebagaimana yang telah dijelaskan pada BAB II mengenai pengujian aplikasi. pengujian ini hanya berdasarkan pada kebutuhan fungsional dari sistem yang telah dijabarkan pada tahapan perencanaan kebutuhan. Pada tahap ini, pengguna diminta untuk menjalankan aplikasi ini sesuai dengan skenario yang tertera di dalam *form* pengujian. Hasil dari pengujian ini akan dilampirkan di dalam lampiran.

### 4.4.1 Metode Exponential Smoothing dan EOQ

Pengujian sistem dilakukan dengan cara menghitung permalan penjualan dengan metode *exponential smoothing* yang telah dilakukan oleh sistem. Data transaksi obat yang ada merupakan data *dummy* bukan data sebenarnya karena pihak apotek tidak memberikan izin untuk memberikan data transaksi tersebut. Dalam sistem ini, peramalan dilakukan dengan menggunakan data penjualan dalam satu periode (satu periode = tiga bulan). Oleh karena itu, dibutuhkan setidaknya data penjualan selama tiga periode (sembilan bulan) di dalam menjalankan sistem ini.



Gambar 4.37 Hasil Peramalan

Pada gambar 4.37, terdapat bulan awal dan bulan akhir, yang merupakan periode atau waktu selama tiga bulan yang digunakan dalam melakukan permalan. Obat yang digunakan di dalam peramalan tersebut adalah Sanmol (PCT) dengan kode barang 13. Diketahui jumlah penjualan Sanmol (PCT) di dalam tiga bulan seperti yang tertera pada kolom jumlah. Setelah mengetahui data tersebut, sistem melakukan perhitungan peramalan dengan metode *exponential smoothing* dengan *alpha* 0,1 sampai 0,9. Peramalan pada periode pertama mengambil jumlah penjualan pada periode pertama, sehingga memiliki nilai yang sama.



Gambar 4.38 Peramalan, MSE dan MAPE

Setelah mengetahui jumlah peramalan dari masing masing *alpha*, sistem menghitung jumlah MSE (*Mean Squared Error*) dan MAPE (*Mean Absolute Persentage Error*), seperti yang terlihat pada gambar 4.38. Dipilih MSE dan MAPE terkecil untuk mendapatkan *alpha* terbaik dari permalan yang sedang dilakukan. Diketahui nilai *alpha* terbaik adalah *alpha* 0,4 dengan nilai MSE 21,35 dan MAPE 15,13. Dengan begitu peramalan penjualan jumlah barang pada periode selanjutnya adalah 21,12.

Yang dilakukan untuk melakukan perhitungan EOQ adalah menentukan data yang diperlukan dalam perhitungan. Adapun data yang dibutuhkan antara lain biaya penyimpanan, biaya pemesanan, waktu tunggu (*lead time*), dan jumlah permintaan pada satu periode. Jumlah permintaan pada satu periode di tentukan dari nilai peramalan pada periode selanjutnya yang telah dihitung sbelumnya, yaitu 21,12. Data lainnya diinput manual oleh pengguna, karena setiap barang memiliki biaya tersendiri. Sebagai contoh biaya penyimpanan menggunakan 10% dari harga barang 16500 menjadi 1650. Biaya pemesanan adalah 5000 yang berupa biaya transportasi, biaya telepon, dan biaya lainnya. Waktu tunggu yang diperlukan agar barang sampai ke apotek adalah 2 hari.

* Biaya penyimpanan : 1650
* Biaya pemesanan : 5000
* Waktu tunggu : 2 hari

Dari data yang telah ditentukan kemudian dimasukkan ke dalam sistem dan sistem melakukan perhitungan terhadap nilai EOQ dan nilai *reorder point* pada obat Sanmol (PCT).



Gambar 4.39 Perhitungan EOQ

Jadi, dari hasil dari perhitungan seperti gambar 4.39, dapat terlihat bahwa persediaan optimum yang harus dimiliki apotek terhadap Sanmol (PCT) adalah 11,31 dengan jumlah *reorder point* 12,11.

# BAB V

# PENUTUP

## 5.1 Kesimpulan

Berdasarkan perancangan, implementasi dan hasil pengujian dari sistem informasi persediaan dan peramalan data obat pada Apotek Nadia Farma dengan menggunakan metode *Exponential Smoothing* dan EOQ (*economic order quantity*) dengan menggunakan metode penelitian *Iterative Waterfall Mode* dapat diperoleh kesimpulan sebagai berikut:

1. Analisis sistem informasi persediaan dan peramalan data obat pada Apotek Nadia Farma dilakukan dengan melakukan observasi dan wawancara terhadap calon pengguna sistem untuk mendapatkan data kebutuhan sistem. Sistem informasi persediaan dan peramalan telah dibuat sesuai dengan perancangan dengan menggunakan bahasa pemrograman PHP dan *framework CodeIgniter*. Sedangkan implementasi *database* menggunakan *Relational Database Management System* (RDBMS), yaitu MySQL. Sistem ini telah diuji oleh pihak apotek dengan metode pengujian *blackbox* dengan menguji sistem secara fungsional. Hasil dari pengujian adalah semua fungsi di dalam sistem berjalan sesuai dengan yang diharapkan.
2. Dari pengujian peramalan dengan menggunakan *exponential smoothing,* sistem ini dapat meramalkan jumlah penjualan obat pada periode selanjutnya dengan baik.
3. Berdasarkan pengujian perhitungan pemesanan bahan baku dengan metode EOQ, pengguna dapat melihat jumlah pembelian bahan baku dan titik pemesanan kembali pada obat.

## 5.2 Saran

Adapun saran yang dapat diberikan untuk pengembangan penelitian selanjutnya adalah sebagai berikut:

1. Sistem informasi penyimpanan dan peramalan obat dapat dikembangkan dengan menambah beberapa fitur baru, seperti memberikan notifikasi jika terdapat obat yang akan habis, penjualan obat menggunakan *barcode scan*, dan beberapa fitur yang sesuai dengan kebutuhan pengguna.
2. Memberikan informasi mengenai obat mana yang lebih laku dan diberikan grafik penjualan obatnya agar lebih jelas dalam membandingkan tingkat penjualan obat.
3. Pembuatan laporan transaksi yang lebih *detail* dan mudah agar memudahkan pengguna dalam membuat laporan. Selain itu, hal ini dapat memudahkan pengguna dalam mengambil keputusan.

# DAFTAR PUSTAKA

Aprilian, Y., 2014. Sistem Informasi Penyewaan Kamar Pada Hotel Candra Kirana. *Indonesian Jurnal on Computer Science,Vol.* 11 No.1, pp. 20-25.

Damayanti, R., 2014 Pembuatan Sistem Informasi Pemesanan Dan Pembayaran Kamar Pada Hotel Remaja Pacitan. *Indonesian Journal on Computer Science, Vol. 3 No. 4 ,*pp. 1-8.

Danuri, M., 2009. Object oriented Programming (OOP) Pembangun Program Aplikasi Berbasis Windows. *INFOKAM, Vol.* 5 No.1, pp. 40-47.

Facrurazzi, S., 2015. Peramalan Penjualan Obat Menggunakan Metode Single Exponential Smoothing Pada Toko Obat Bintang Geurogok. *TECHSI, Vol.*6 No.1, pp. 20-30.

Fatona, E., 2013. Pengembangan Aplikasi Sistem Pendukung Keputusan Pengontrol Persediaan dan Jumlah Pembelian Barang Menggunakan Metode EOQ (Economic Order Quantity). *Karmapati, Vol.*2 No.1, pp. 317-331.

Gessong, Y., 2001. Analisis Persediaan Pada Furniture Sangasanga. *Eksis, Vol.*8 No.1, pp. 2032-2035.

Hansen, M., 2001. *Manajemen Biaya.* 1 penyunt. Jakarta: Salemba Empat.

Himawan, H., 2014. Efektifitas Penggunaan Metode Exponential Smoothing Pada Peramalan Produk. Skripsi Program Studi Informatika, UPN "Veteran" Yogyakarta.

Jovanovic, I., 2009. *Software Testing Methods and Techniques,* s.l.: s.n.

*Kamus Besar Bahasa Indonesia (KBBI).*( n.d.). 29 June 2016. http://kbbi.web.id/obat.

Khan, E. & Khan, F., 2012. Comparative Study of White Box, Black Box, and Grey Box Testing Techniques. *(IJACSA) International Journal of Advanced Computer Science and Applications, Vol.*3 No.6, pp. 12-15.

Kyei, J., Boateng, S., Gyimah, P. & Annan, J. B., 2008. *Principles and Techniques of Managing Inventory.* s.l.:s.n.

Lestari, G. & Setyorini, R., 2014. Analisis Pengendalian Persediaan Bahan Baku Daging Dengan Menggunakan Metode Economic Order Quantity (EOQ) Pada Restoran Steak Ranjang Bandung. Skripsi Program Studi Ilmu Administrasi Bisnis Fakultas Komunikasi dan Bisnis, Universitas Telkom.

Liliek, Y. P., 2012. Sistem Informasi Manajemen Agenda Pada Badan Pelayanan Perijinan Terpadu Kabupaten Karanganyar. *Indonesian Journal on Computer Science Speed, Vol.*9 No.3, pp. 1-6.

Mayora, H., 2015. *Perbandingan Sistem Economic Order Quantity dan Just In Time Pada Pengendalian Persediaan Bahan Baku,* Semarang: Skripsi Fakultas Ekonomi dan Bisnis Universitas Dipenogoro .

Maheswari, S. & Jain, D., 2012. A Comparative Analysis of Different types of Models in Software Development Life Cycle. *International Jurnal of Advanced Research in Computer Science and Software Engineering, Vol.*2 No.5, pp. 285-290.

Meilani, D. & Eka, R. S., 2013. Pengendalian Persediaan Bahan Baku Vulkanisisr Ban(Studi Kasus: PT. Gunung Pulo Sari. *Optimasi Sistem Industri, Vol.*12 No.1, pp. 326-334.

Mulyani, I., 2011. *www.informasi-obat.com.* 01 June 2016. http://www.informasi-obat.com/pengertian-obat-yang-harus-anda-ketahui.html.

Munassar, N. & Govardhan, 2010. A Comparison Between Five Models Of Software Engineering. *IJCSI, Vol.*7, No.5, pp. 94-101.

Nagilla, R., 2013. *Comparison of Web Development Technologies- ASP.NET & PHP,* s.l.: s.n.

Nahar, J., 2013. Menentukan Persediaaan Beras Dengan Menggunakan Model Economic Order Quantitiy (EOQ) Berdasarkan Ramalan Permintaan Pada Tahun 2012. *PTNBR-BATAN,* pp. 619-623.

Priyadna, A. & Yulianto, L., 2012. Pembuatan SIstem Registrasi Kamar Berbasis Website Pada Hotel Graha Prima Pacitan. *Indonesian Journal on Networking and Security,* Volume 2, pp. 8-14.

Putu, S., 2012. Sistem Just In Time (JIT) Penting Bagi Perusahaan Industri. *Dharma Ekonomi,* Volume 36.

Radack, S., 2009. *The System Development Life Cycle (SDLC),* s.l.: s.n.

Rolliawati, D., 2013. *Pemrograman Berorientasi Objek.* s.l.:s.n.

Royyan, M., 2015. Rancang Bangun Sistem Informasi Pembelian dan Perencanaan Persediaan Barang Pada CV. Jaya Tama, Skripsi Program Studi Sistem Informasi, Sekolah Tinggi Manajemen Informatika & Teknik Surabaya.

Sahara, A., 2013. Sistem Peramalan Persediaan Unit Mobil Mitsubishi Pada PT. Sardana Indah Berlian Motor Dengan Menggunakan Metode Exponential Smoothing. *Informasi dan Teknologi Ilmiah (INTI), Vol.*1 No.1, pp. 1-7.

Soewarno, 2005. Just In Time (JIT) sebagai Upaya untuk Meningkatkan Competitive Advantage. *Majalah Ekonomi,* Volume 3A, pp. 425-440.

Sungkawa, I. & Tri, R. M., 2011. Penerapan Ukuran Ketepatan Waktu Nilai Ramalan Data Deret Waktu Dalam Seleksi Model Peramalan Volume Penjualan PT Satriamandiri Citramulia. *ComTech, Vol.*2, No.2, pp. 636-645.

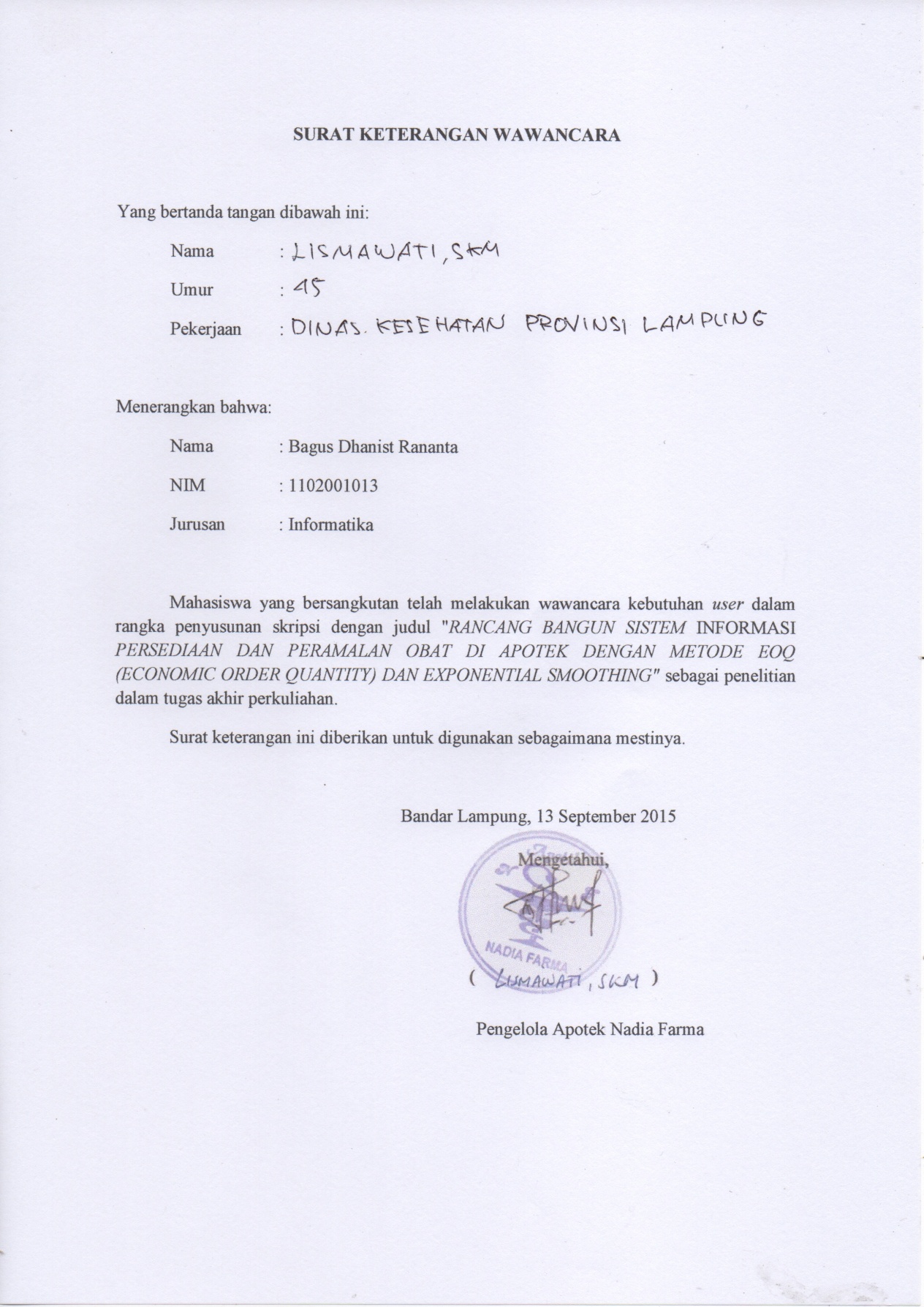
Suprianto, D., 2008. *Buku Pintar Pemrograman PHP.* Bandung: Oase Media.

Valerie, C. & Sinuraya, C., 2011. Perbandingan Metode EOQ (Economic Order Quantity) dan JIT (Just In Time) Terhadap Efisiensi Biaya Persediaan dan Kinerja Non-Keuangan. *Akurat Jurnal Ilmiah Akuntansi, Vol.*5 No.2.

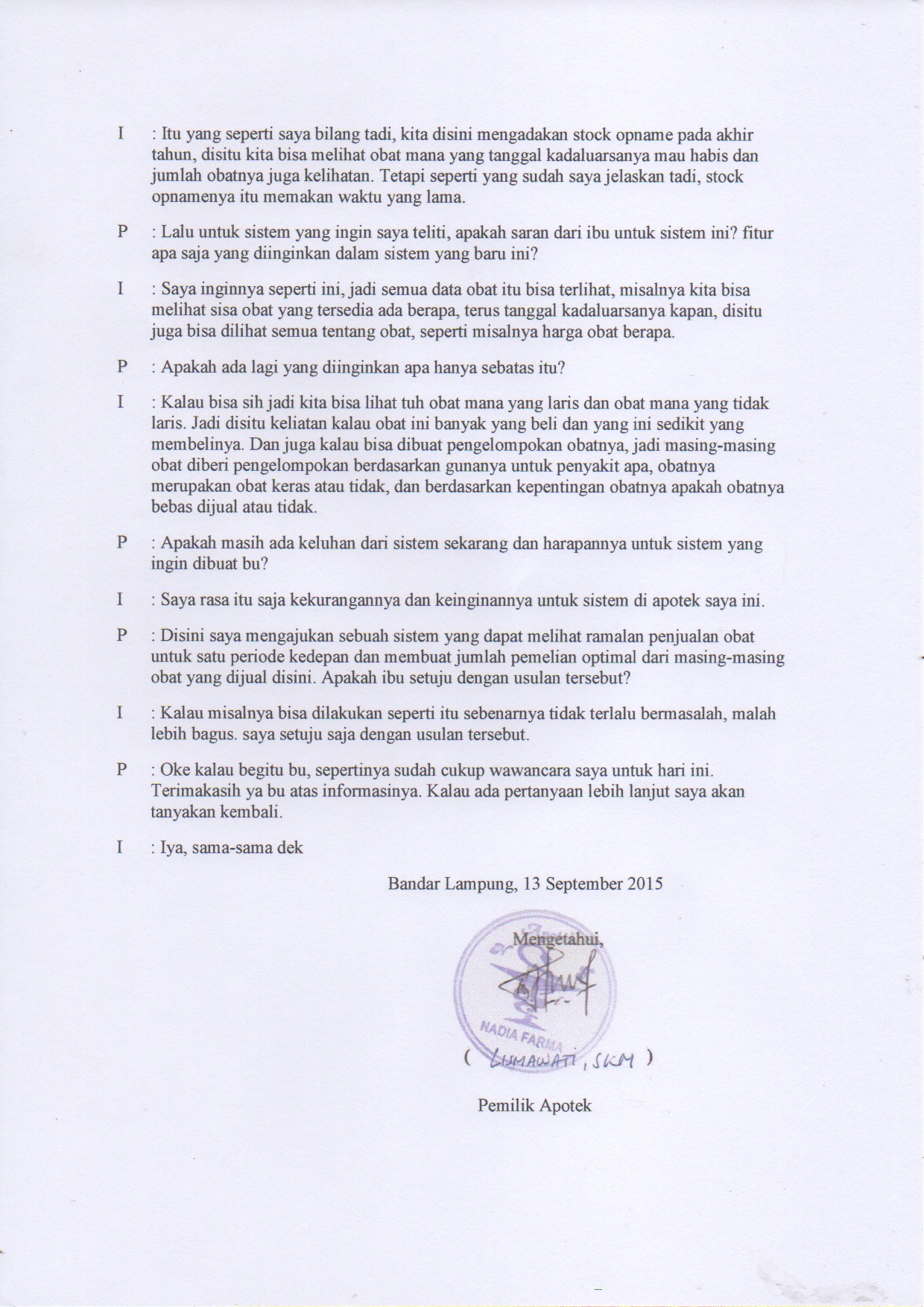
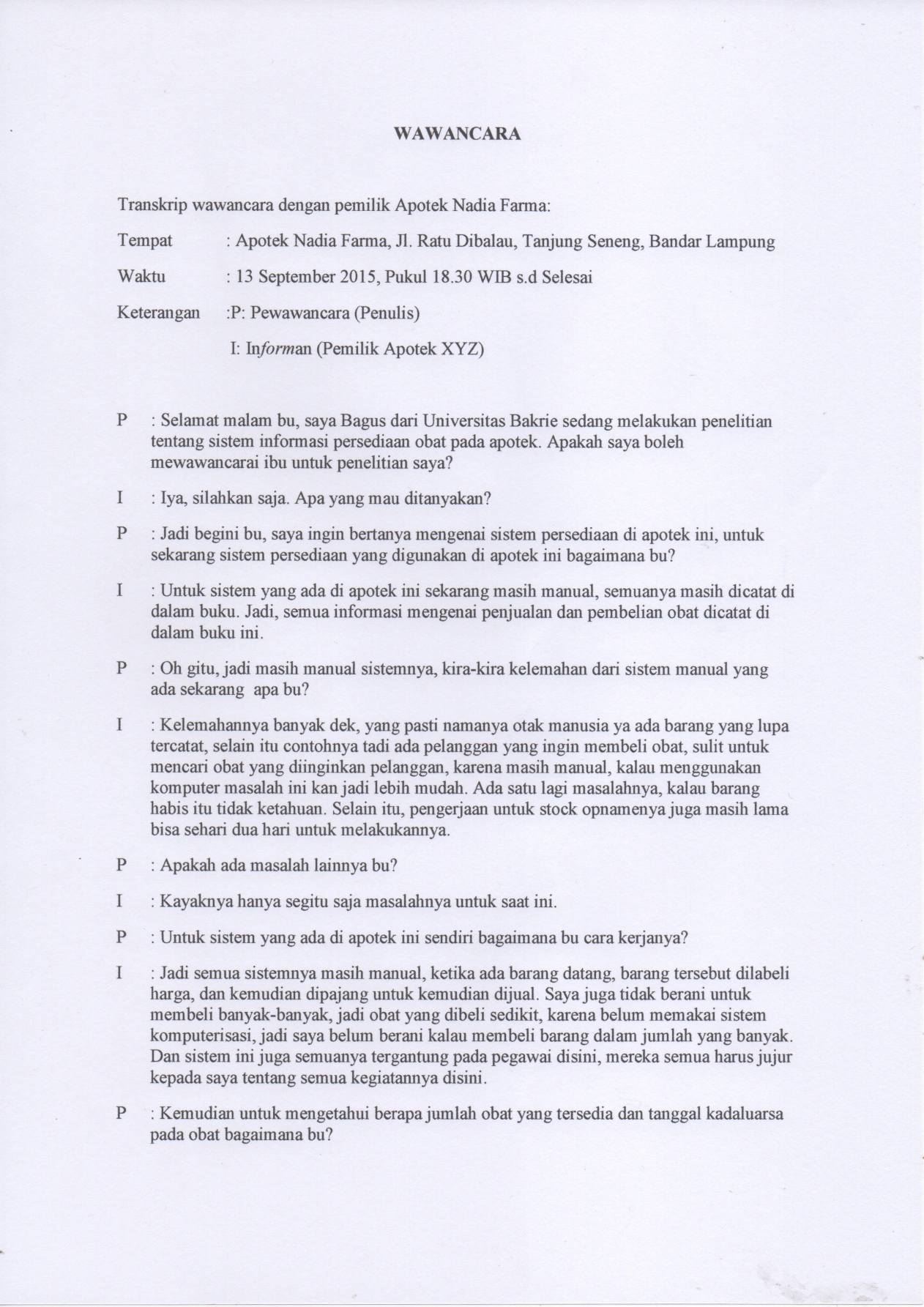
Wibisetiadi, Y., 2011. Rancang Bangun Sistem Informasi Apotek Sinar, Skripsi Program Studi Sistem Informasi, Sekolah Tinggi Manajemen Informatika & Teknik Surabaya, pp. 1-12.

Wijaya, A., 2014. Analisis Pengendalian Internal Persediaan Obat Pada Apotek Kencana Semarang, Skripsi Fakultas Ekonomi Dan Bisnis Program Studi Akuntansi, Universitas Dian Nuswantoro Semarang.

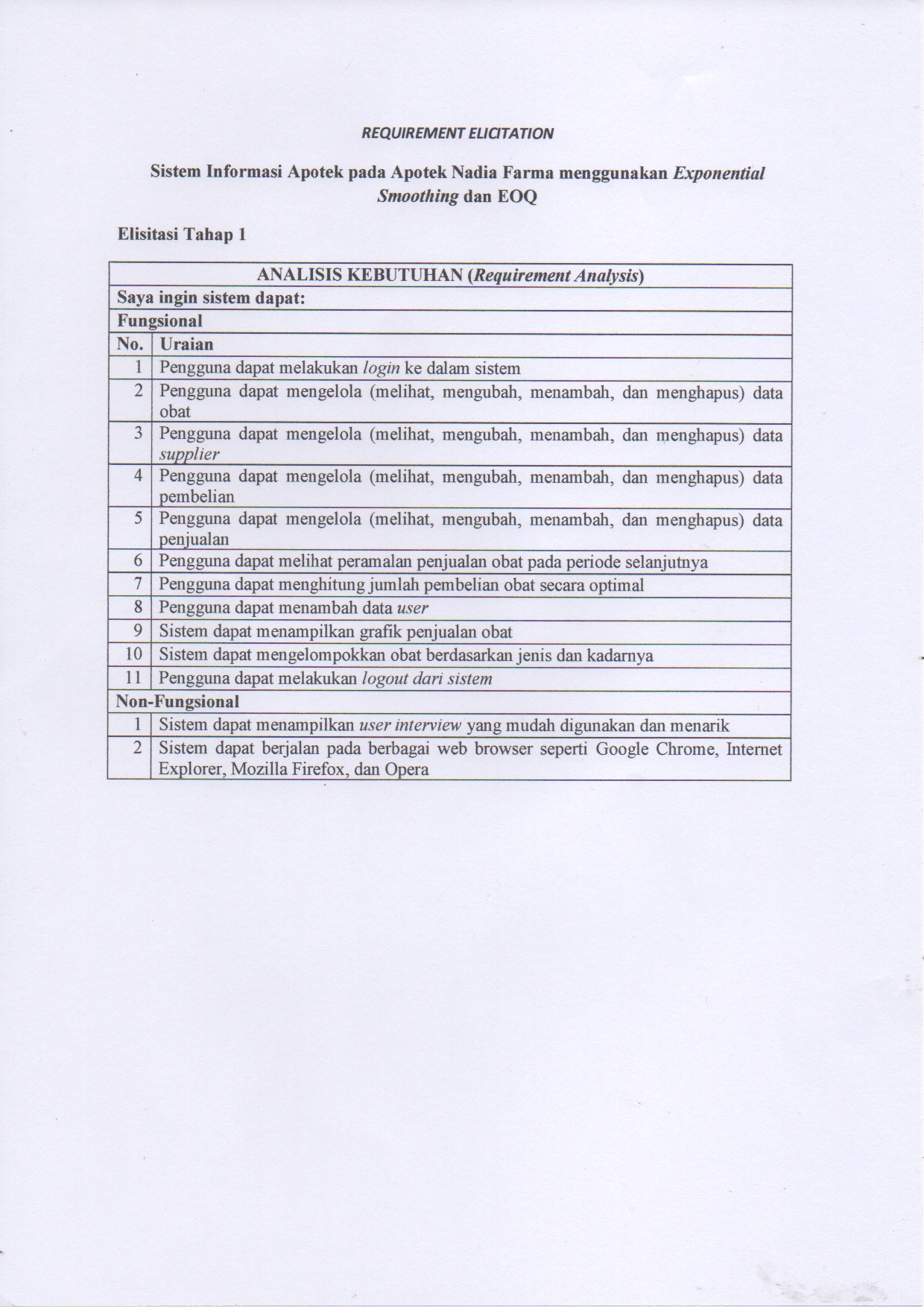
Yogeswara, I. W. K., Wisnubhadra, I. & Mudjihartono, P., 2013. Analisis Rancang Bangun Sistem Informasi Hotel Terintegrasi yang Selaras Dengan Rencana Strategis Teknologi Informasi (Studi Kasus: Hotel Dalu). Tesis Program Magister Teknik Informatika, Universitas Atma Jaya Yogyakarta.

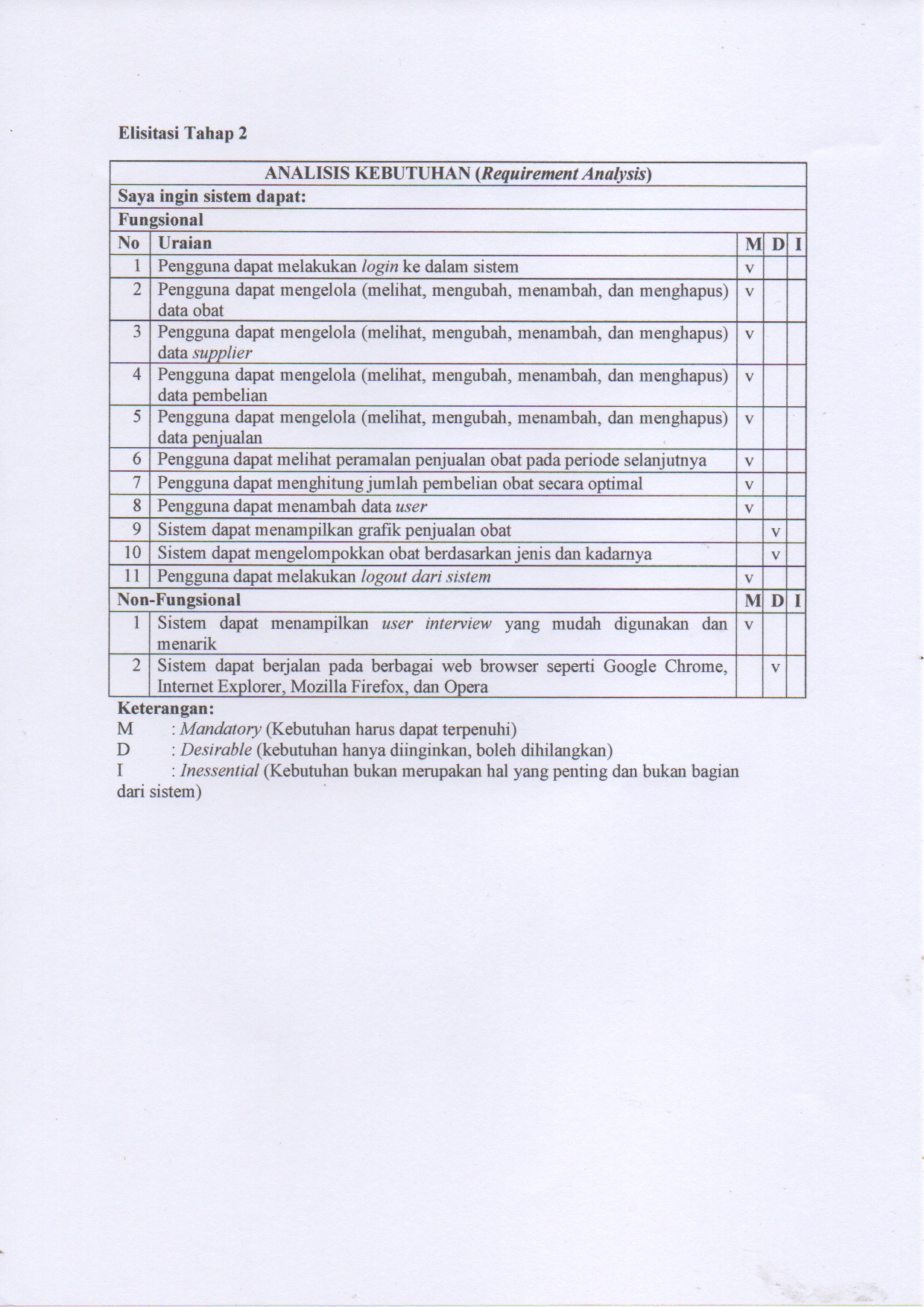
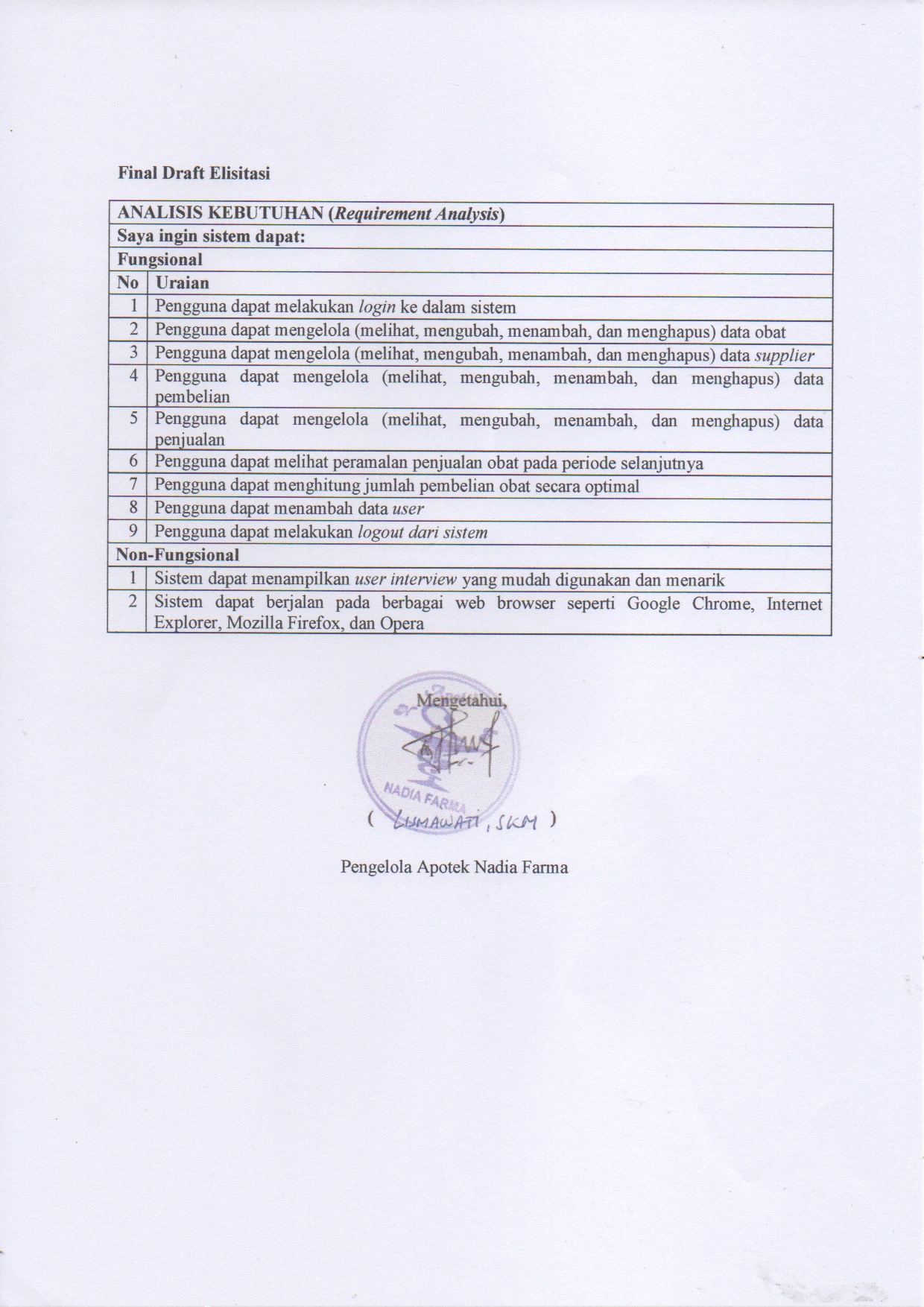
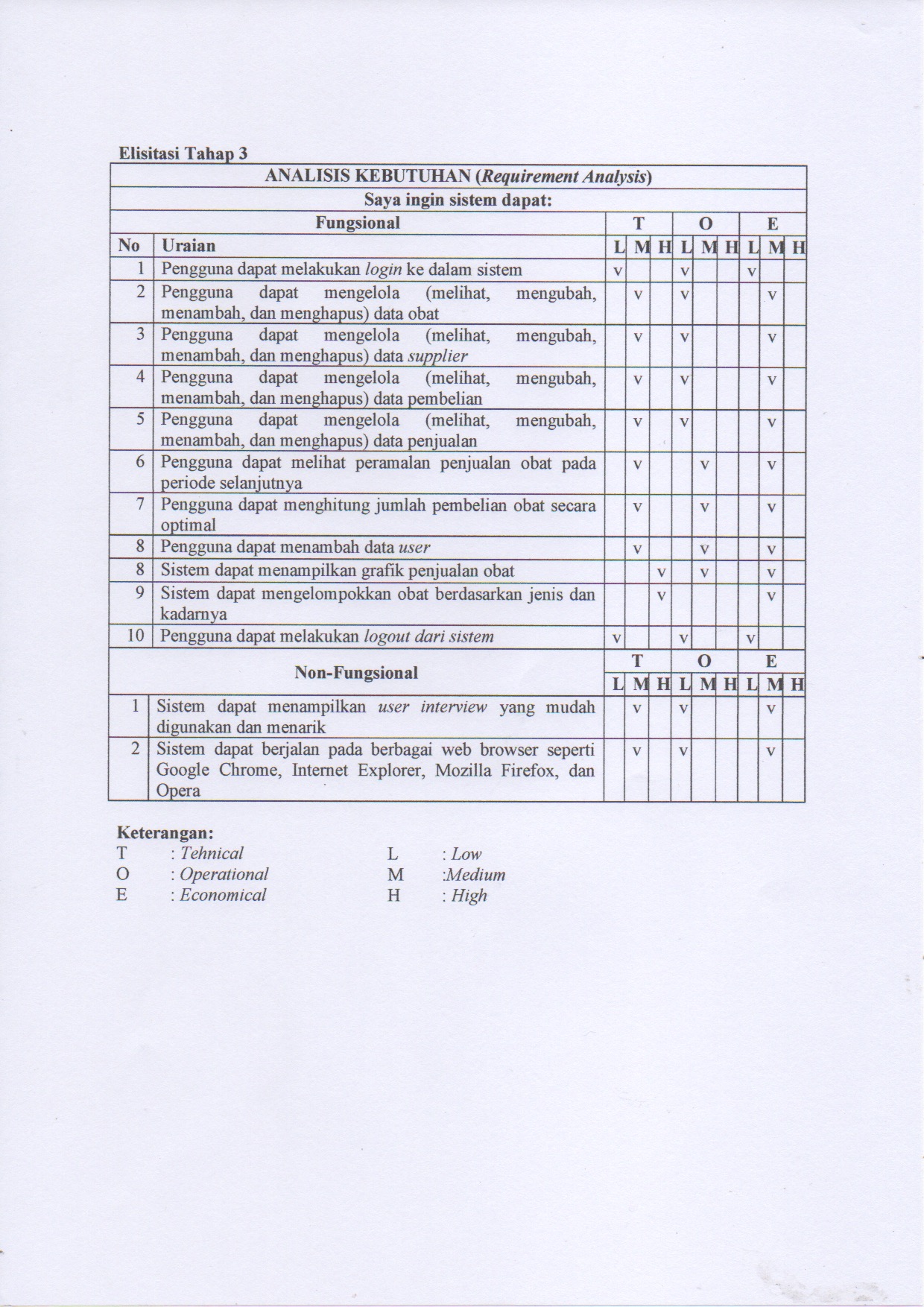
**Lampiran 1: Surat Keterangan Wawancara**

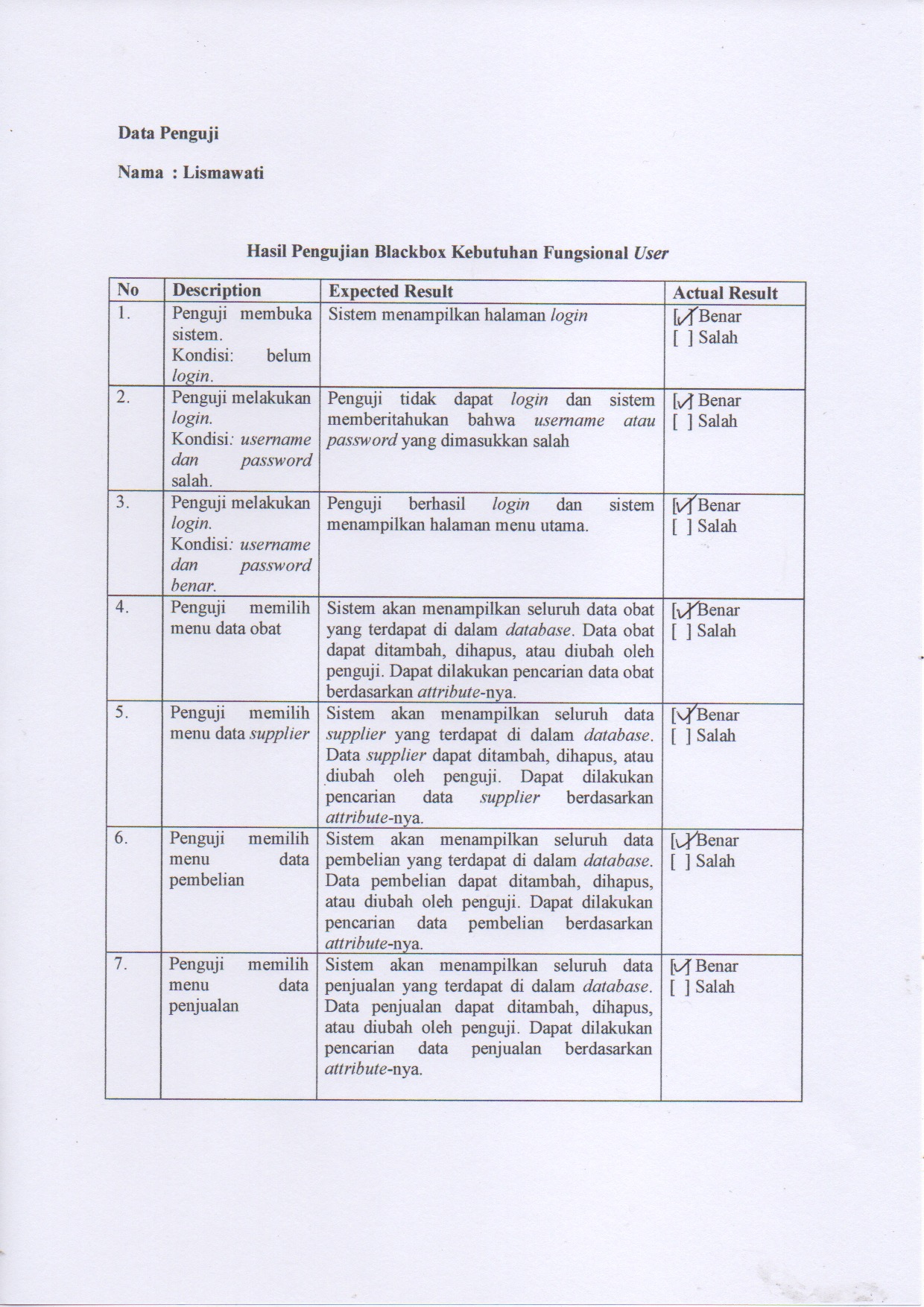
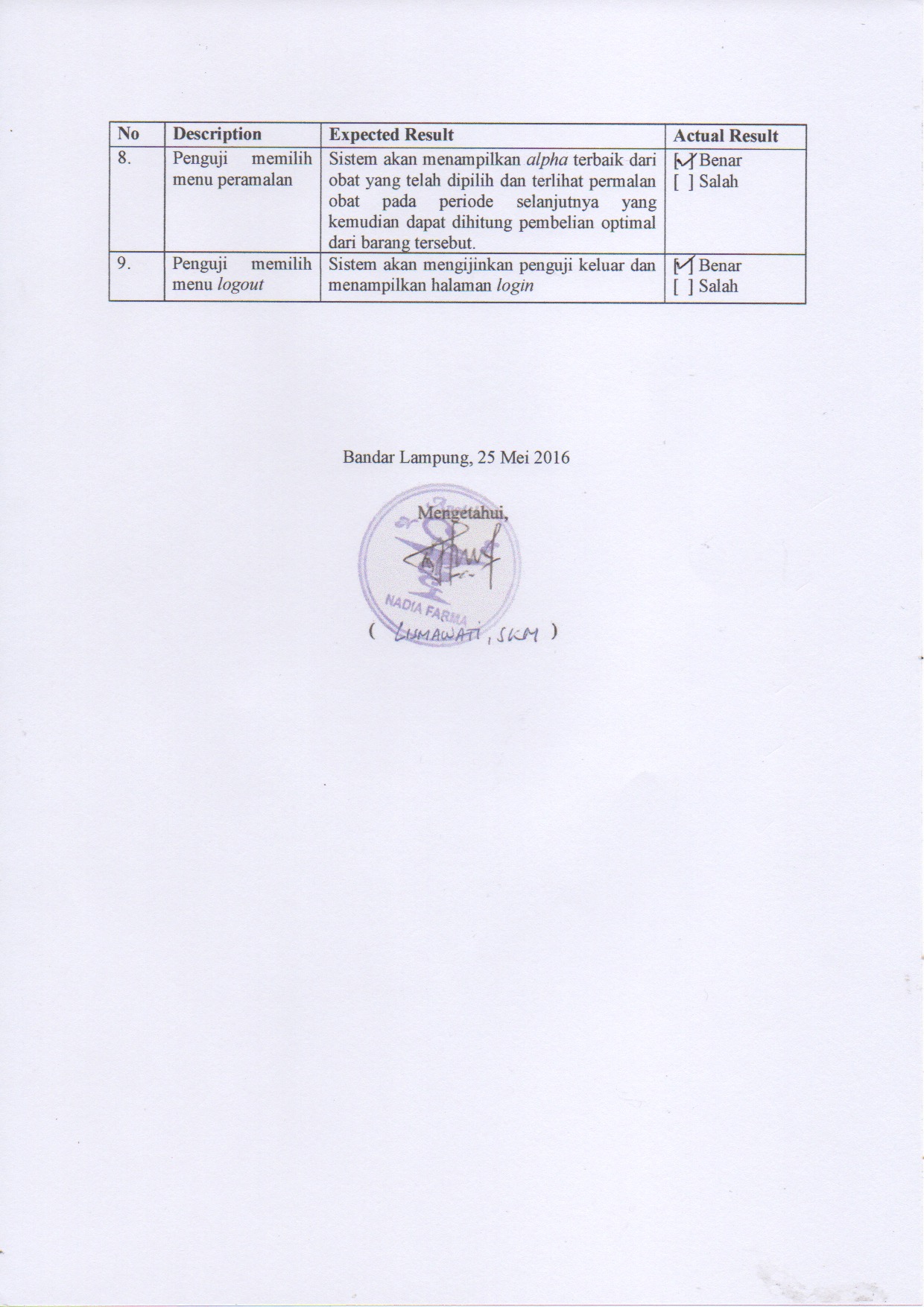
**Lampiran 2: Lembar Wawancara**

****

**Lampiran 3: Elisitasi**



** **

**Lampiran 4: Pengujian Aplikasi****** ****

**Lampiran 4 :*Software Requirements Specification***

***Software Requirements Specification***

**Sistem Informasi**

**Apotek Nadia Farma**

**Bandar Lampung**

**Dibuat oleh:**

**Bagus Dhanist Rananta**

**Teknik Informatika Universitas Bakrie**

**2015**

DAFTAR ISI

[1. Pendahuluan 12](#_Toc453970640)

[1.1 Tujuan Penelitian Dokumen 12](#_Toc453970641)

[1.2 Ruang Lingkup Masalah 12](#_Toc453970642)

[1.3 Definisi, Akronim, dan Singkatan 12](#_Toc453970643)

[1.4 Deskripsi Umum Dokumen 13](#_Toc453970644)

[2. Deskripsi Global Perangkat Lunak 13](#_Toc453970645)

[2.1 Fungsi Produk 13](#_Toc453970646)

[2.2 Karakteristik Pengguna 14](#_Toc453970647)

[2.3 Batasan 14](#_Toc453970648)

[2.4 Asumsi dan Ketergantungan 15](#_Toc453970649)

[2.5 Metode Pengembangan Sistem 15](#_Toc453970650)

[1. Analisis dan Definisi Persyaratan 15](#_Toc453970651)

[3. Rancangan Sistem 17](#_Toc453970652)

[3.1 Kebutuhan Fungsional Sistem 17](#_Toc453970653)

[3.2 Kebutuhan Non-Fungsional Sistem 17](#_Toc453970654)

[3.3 Kebutuhan Antarmuka Eksternal 18](#_Toc453970655)

[3.3.1 *User Interface* 18](#_Toc453970656)

[3.3.2 *Hardware Interface* 18](#_Toc453970657)

[3.3.3 *Software Interface* 18](#_Toc453970658)

[3.3.4 *Communication Interface* 18](#_Toc453970659)

[4. *Applicable Standards* 19](#_Toc453970660)

[5. Kebutuhan Lisensi 19](#_Toc453970661)

[6. *Legal, Copyright, and Other Notices* 19](#_Toc453970662)

[7.Informasi Tambahan 19](#_Toc453970663)

# 1. Pendahuluan

Dokumen ini berisi tentang *Sofwtware Requirement Specification* (SRS) yang didalamnya berisi penjelasan mengenai kebutuhan penggunaan sistem berbasis web, yang terdiri kebutuhan fungsional dan non-fungsional dari sistem yang akan dibuat. Kebutuhan *interface* dijelaskan juga dlam dokumentasi ini seperti kebutuhan *client,* spesifikasi minimum hardware yang harus terpenuhi untuk dapat menjalankan sistem ini.

Dokumen ini juga menetapkan batasan masalah agar sistem yang akan dibuat tidak menyimpang dari tujuan awal serta sesuai dengan kebutuhan pengguna. SRS ini juga dapat menjadi acuan dalam pembuatan aplikasi agar dapat berjalan dengan baik sesuai dengan yang diinginkan.

## 1.1 Tujuan Penelitian Dokumen

Tujuan dari penulisan dokumen spesifikasi kebutuhan perangkat lunak ini adalah untuk memberi gambaran yang mendetail tentang kebutuhan yang dibutuhkan dalam pembuatan dan pengembangan sistem yang akan dibuat.

## 1.2 Ruang Lingkup Masalah

Sistem yang dikembangkan adalah berbasis *web*. Sistem ini merupakan sistem informasi yang akan digunakan oleh pengguna pada Apotek Nadia Farma sebagai sistem yang dilakakan untuk pengelolaan data obat, *supplier*, pembelian, penjualan, peramalan obat, dan perhitungan jumlah pembelian optimal pada obat. Pengembangan sistem ini dilakukan untuk menggantikan sistem manual yang selama ini dipakai oleh Apotek Nadia Farma.

## 1.3 Definisi, Akronim, dan Singkatan

SRS : *Software Requirement Specifications*

*Hardware* : Perangkat keras

*Software* : Perangkat lunak

*Interface* : Antar muka

## 1.4 Deskripsi Umum Dokumen

Dokumen SRS ini merupakan acuan atau standar dalam menyelesaikan pengembangan sistem informasi berbasis web pada Apotek Nadia Farma agar sesuai dengan spesifikasi kebutuhan yang telah ditentukan sehingga dapat menghasilkan aplikasi yang sesuai dengan kebutuhan klien.

# 2. Deskripsi Global Perangkat Lunak

Sistem informasi ini merupakan sebuah sistem berbasis *web* yang dapat digunakan untuk melakukan beberapa aktivitas yang terdapat pada Apotek Nadia Farma. Aktivitas yang dimaksud adalah mengelola data obat, *supplier*, pembelian, penjualan, peramalan, menentukan jumlah pembelian untuk masing-masing obat dan elemen terkait dalam menjalankan pekerjaan di apotek. Sistem ini akan menggantikan sistem manual yang selama ini digunakan.

## 2.1 Fungsi Produk

Fungsi utama dasi sistem ini adalah:

1. Melakukan validasi dalam proses *login.*
2. Memasukkan data barang, *supplier*, *user,* penjualan, dan pembelian ke dalam *database.*
3. Menampilkan data barang, *supplier*, *user,* penjualan, dan pembelian di dalam *database.*
4. Mengubah data barang, *supplier*, *user,* penjualan, dan pembelian di dalam *database.*
5. Menghapus data barang, *supplier*, *user,* penjualan, dan pembelian di dalam *database.*
6. Melakukan validasi dalam proses memasukkan data melalui sistem.
7. Melakukan validasi dalam proses perubahan isi data melalui sistem.

## 2.2 Karakteristik Pengguna

Berikut merupakan karakteristik pengguna pada sistem:

|  |  |
| --- | --- |
| **Pengguna** | **Aksi** |
| *User* | 1. *User* dapat melakukan *login* ke dalam sistem. 2. *User* dapat membuat, melihat, mengubah, dan menghapus data *user.* 3. *User* dapat membuat, melihat, mengubah, dan menghapus data obat*.* 4. *User* dapat membuat, melihat, mengubah, dan menghapus data *supplier.* 5. *User* dapat membuat, melihat, mengubah, dan menghapus data penjualan*.* 6. *User* dapat membuat, melihat, mengubah, dan menghapus data pembelian*.* 7. *User* dapat melakukan *logout* ke dalam sistem. |

## 2.3 Batasan

Batasan-batasan dalam pengembangan sistem ini adalah:

1. Sistem ini akan dikembangkan menggunakan bahasa pemograman PHP bersama *Codeigniter framework.*
2. *Database* yang digunakan adalah MySQL.
3. Aplikasi ini digunakan untuk pengelolaan data di dalam Apotek Nadia Farma dan peramalan penjualan serta perhitungan jumlah pembelian optimal.
4. Pengembangan sistem belum dilakukan implementasi.

## 2.4 Asumsi dan Ketergantungan

Asumsi dan ketergantungan yang akan dikembangkan adalah sebagai berikut:

1. Pengguna harus memahami penggunaan komputer, dan *browser.*
2. Sistem dapat diakses tanpa menggunakan akses internet.
3. Spesifikasi minimum komputer pengguna agar dapat mengakses aplikasi ini adalah sebagai berikut:

* *Device* : Komputer atau *laptop*
* *Operating system* : *Windows XP* atau setara
* *Processor* : 1 GHz
* *RAM (Memory)* : 1.00 GB
* *Browser* : *Google Chrome, Mozilla firefox* atau setara

## 2.5 Metode Pengembangan Sistem

Pengembangan sistem informasi Apotek Nadia Farma akan menggunakan metode *Iterative Waterfall* dengan tahapan sebagai berikut:

### Analisis dan Definisi Persyaratan

Tahap ini merupakan tahapan awal dari penelitian. Tahap ini merupakan tahap pencarian topik dan ide penelitian. Pada definisi persyaratan, dilakukan pendefinisian dari seluruh kebutuhan yang dapat dilayani oleh sistem yang akan dibangun. Pada tahap analisis difokuskan untuk mengumpulkan kebutuhan sistem dengan lebih spesifik.

1. Perancangan Sistem dan Perangkat Lunak

Pada tahap ini dilakukan perancangan dan penggambaran tentang sistem yang akan dibuat. Perancangan ini melibatkan identifikasi dan deskripsi dari abstraksi sistem yang mendasar dan hubungan-hubungannya. Perancangan sistem dan perangkat lunak terdiri dari perancangan data, perancangan arsitektur, dan perancangan antarmuka.

1. Implementasi dan Pengujian Unit

Pada tahap ini, perancangan sistem yang telah dibuat diterapkan sebagai serangkaian program atau unit program. Sistem diterapkan dengan melakukan penulisan kode yang merupakan terjemahan dari hasil perancangan ke dalam perangkat keras. Pengujian dilakukan dengan menguji setiap unit dari sistem. Pengujian dari unit sistem ini digunakan untuk memverifikasi bahwa semua bagian unit dari sistem telah berjalan dengan baik dan telah memenuhi spesifikasinya.

1. Integrasi dan Pengujian sistem

Pada tahap ini, setiap unit sistem yang telah diuji diintegrasikan satu sama lain. Setelah diintegrasikan, maka akan dilakukan pengujian pada sistem secara keseluruhan. Pengujian ini dilakukan untuk menjamin bahwa semua persyaratan telah dipenuhi dan tiap-tiap fungsi dari sistem sudah terhubung dengan baik.

1. Operasi

Tahapan ini tidak dilakukan dikarenakan pada tahap ini memerlukan waktu yang lama sedangkan penelitian ini bertujuan untuk mengerjakan tugas akhir. Oleh karena itu, tahapan ini tidak dilakukan dalam penelitian ini.

# 3. Rancangan Sistem

## 3.1 Kebutuhan Fungsional Sistem

Secara Fungsional, sistem diharapkan dapat mendukung pengguna untuk melakukan kegiatan berikut:

1. Pengguna dapat melakukan *login* ke dalam sistem.
2. Pengguna dapat mengelola (melihat, menambah, dan mengubah) data *user*.
3. Pengguna dapat mengelola (melihat, menambah, dan mengubah) data barang.
4. Pengguna dapat mengelola (melihat, menambah, dan mengubah) data *supplier*.
5. Pengguna dapat mengelola (melihat, menambah, dan mengubah) data pembelian.
6. Pengguna dapat mengelola (melihat, menambah, dan mengubah) data penjualan.
7. Pengguna dapat melihat peramalan penjualan.
8. Pengguna dapat menghitung jumlah pembelian optimal (EOQ).
9. Pengguna dapat melakukan *logout* dari sistem.

## 3.2 Kebutuhan Non-Fungsional Sistem

Secara non-fungsional, sistem diharapkan dapat memenuhi kriteria sebagai berikut:

1. Sistem dapat menyajikan *user interface* yang mudah digunakan dan menarik.
2. Sistem dapat berjalan melalui *web browser.*
3. Sistem mampu menjaga privasi data pengguna dengan melakukan enkripsi *password.*

## 3.3 Kebutuhan Antarmuka Eksternal

### 3.3.1 *User Interface*

*User interface* dari sistem administrasi ini merupakan bagian dari perangkat lunak dan memiliki peran untuk mempermudah interaksi antara pengguna dengan sistem.

### 3.3.2 *Hardware Interface*

Antar muka *hardware (*perangkat keras) yang dibutuhkan untuk membantu kelengkapan dari pembangunan aplikasi yang sedang dirancang meliputi:

* *Device* : *Laptop* Lenovo W520
* *Operating System* : Windows 7 64-bit
* *Processor* : Intel core i7-2860QM
* RAM (*Memory)* : 6 GB

### 3.3.3 *Software Interface*

Antar muka *software (*perangkat lunak) yang dibutuhkan untuk membantu kelengkapan dari pembangunan aplikasi yang sedang dirancang meliputi:

* *SublimeText* Versi 3 sebagai *Code Editor.*
* XAMPP.
* *Google Chrome* dan *Mozilla Firefox* sebagai *web browser.*

### 3.3.4 *Communication Interface*

Desain *communication interface* sistem yang dikembangkan menggunakan *apache webserver* sebagai penghubung antara server dengan komputer pengguna.

# 4. *Applicable Standards*

Standar aplikasi yang dapat diterapkan adalah:

* XAMPP 5.6.11
* Codeigniter 3.0.6

# 5. Kebutuhan Lisensi

Implementasi dan instalasi aplikasi ini akan terdistribusi sesuai dengan ketentuan operasional yang beelaku dari pihak pengguna, dalam hal ini adalah Apotek Nadia Farma.

# 6. *Legal, Copyright, and Other Notices*

Hak cipta sistem informasi ini menjadi milik pihak Apotek Nadia Farma. Segala sesuatu yang menyangkut sistem ini menjadi hak Apotek Nadia Farma.

# 7.Informasi Tambahan

Dokumen-dokumen yang terkait dalam pembuatan Spesifikasi Kebutuhan perangkat lunak ini meliputi:

* Dokumen Elisitasi Kebutuhan.
* Diagram UML untuk perancangan sistem pada BAB IV penelitian.
* Dokumen hasil wawancara dengan pihak Apotek Nadia Farma.