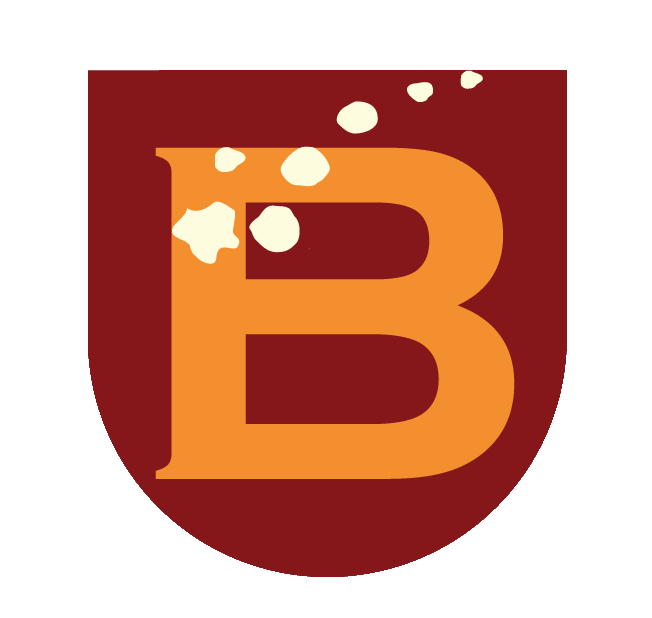
# LAPORAN TUGAS AKHIR

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

****

**Bagus Dhanist Rananta**  
**1102001013**

**PROGRAM STUDI INFORMATIKA**

**FAKULTAS TEKNIK DAN ILMU KOMPUTER**

**UNIVERSITAS BAKRIE**

**2015**

# PROGRAM STUDI INFORMATIKA

**UNIVERSITAS BAKRIE**

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



USULAN TUGAS AKHIR

Sebagai salah satu syarat untuk melaksanakan Tugas Akhir

Pada Program Studi Informatika

Universitas Bakrie

**Oleh:**

**Bagus Dhanist Rananta**

**1102001013**

**Disetujui,**

**Jakarta, ...............................**

PembimbingPembahas

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

(**Yusuf Lestanto, ST, M.Sc, MBA**) ( )

# KATA PENGANTAR

Segala puji syukur penulis panjatkan atas kehadirat Allah SWT, yang atas limpahan rahmat, taufik, dan hidayah-Nya, penulis mampu menyelesaikan Proposal Tugas Akhir ini. Terselesaikanya Proposal 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 Teknik Informatika dan Sistem Informasi Universitas Bakrie yang telah memberikan ilmu dan wawasan selama masa perkuliahan.
3. Seluruh keluarga dan rekan-rekan seperjuangan atas doa yang diberikan dan dukungan dalam berbagai hal.

Semoga proposal penelitian ini bisa bermanfaat bagi yang membaca dan dapat dikembangkan lagi lebih jauh di masa depan.

Jakarta, 27 Agustus 2015

Bagus Dhanist Rananta

# DAFTAR ISI

[HALAMAN JUDUL i](#_Toc432503263)

[LEMBAR PENGESAHAN ii](#_Toc432503264)

[KATA PENGANTAR iii](#_Toc432503265)

[DAFTAR ISI iv](#_Toc432503266)

[DAFTAR GAMBAR vi](#_Toc432503267)

[DAFTAR TABEL vii](#_Toc432503268)

[BAB I 1](#_Toc432503269)

[PENDAHULUAN 1](#_Toc432503270)

[1.1 Latar Belakang 1](#_Toc432503271)

[1.2 Perumusan Masalah 3](#_Toc432503272)

[1.3 Pembatasan Masalah 3](#_Toc432503273)

[1.4 Tujuan Penelitian 3](#_Toc432503274)

[1.5 Manfaat Penelitian 4](#_Toc432503275)

[BAB II 5](#_Toc432503276)

[TINJAUAN PUSTAKA 5](#_Toc432503277)

[3.1 Studi Literatur 5](#_Toc432503278)

[2.2 Apotek 7](#_Toc432503279)

[3.3 Persediaan 7](#_Toc432503280)

[2.4 *Economic Order Quantity (EOQ)* 9](#_Toc432503281)

[2.5Peramalan 12](#_Toc432503282)

[2.6 *Exponential Smoothing* 15](#_Toc432503283)

[2.7 *SDLC (System Development Life Cycle)* 16](#_Toc432503284)

[2.8 Konsep Dasar Sistem Informasi 19](#_Toc432503285)

[2.9 Metode Pengujian Sistem 21](#_Toc432503286)

[2.10 *Object Oriented Programming (OOP)* 22](#_Toc432503287)

[2.11 Bahasa Pemrograman 24](#_Toc432503288)

[BAB III 26](#_Toc432503289)

[METODOLOGI PENELITIAN 26](#_Toc432503290)

[3. 1 Alat Penelitian 26](#_Toc432503291)

[3. 2 Metode Pengumpulan Data 26](#_Toc432503292)

[3. 3 Pengolahan Data 27](#_Toc432503293)

[3. 4 Metode Pengembangan Sistem 29](#_Toc432503294)

[3.4.1 Analisis dan Definisi Persyaratan 29](#_Toc432503295)

[3.4.2 Perancangan Sistem dan Perangkat Lunak 30](#_Toc432503296)

[3.4.3 Implementasi dan Pengujian Unit 30](#_Toc432503297)

[3.4.4 Integrasi dan Pengujian sistem 31](#_Toc432503298)

[3.4.5 Operasi 31](#_Toc432503299)

[3. 5 Jadwal Penelitian 31](#_Toc432503300)

[DAFTAR PUSTAKA 32](#_Toc432503301)

# DAFTAR GAMBAR

[Gambar 2. 1 Klasifikasi Metode Peramalan (Sahara, 2013) 13](#_Toc432503236)

[Gambar 2. 2 Software Development Life Cycle (SDLC) (Maheswari & Jain, 2012) 17](#_Toc432503237)

[Gambar 2. 3 Original Waterfall Model (Maheswari & Jain, 2012) 17](#_Toc432503238)

[Gambar 2. 4 Iterative Waterfall model (Maheswari & Jain, 2012) 18](#_Toc432503239)

[Gambar 2. 5 Prototype model (Maheswari & Jain, 2012) 18](#_Toc432503240)

[Gambar 2. 6 Spiral Model (Maheswari & Jain, 2012) 18](#_Toc432503241)

[Gambar 2. 7 Model Pemrograman OOP (Danuri, 2009) 23](#_Toc432503242)

[Gambar 2. 8 Statistik Penggunaan Tools untuk Pengembangan Web (Nagilla, 2013) 24](#_Toc432503243)

# DAFTAR TABEL

[Tabel 2. 1 Perbandingan dari empat metode SDLC (Maheswari & Jain, 2012) 19](#_Toc432503246)

[Tabel 2. 2 Perbandingan tiga teknik pengujian (Khan & Khan, 2012) 22](#_Toc432503247)

[Tabel 2. 3 Perbedaan Antara PHP dan ASP.NET (Nagilla, 2013) 25](#_Toc432503248)

[Tabel 3. 1 Jadwal Penelitian 20](#_Toc428480206)

# 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).

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.

Penelitian ini dilakukan di apotek xyz. Wawancara dan observasi dilakukan pada tanggal 23 September 2015. Pada saat ini, apotek ini 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 ada 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 xyz. 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 bulan 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 di ketahui dari pembelian barang sampai pada titik konsumsi (Kyei, et al., 2008). Oleh karena itu, dibutuhkan sistem manajemen dan peramalan terhadap persediaan yang ada 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 xyz dalam pengelolaan persediaan obat yang ada 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 persediaan obat pada apotek?
2. Bagaimana cara meramalkan jumlah penjualan obat untuk bulan 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 mencakup sistem penjualan obat pada apotek.
3. Sistem tidak menyimpan data seluruh pegawai.
4. Pengujian aplikasi dilakukan oleh staff apotek dengan menggunakan metode *black box testing.*
5. Sistem hanya mencatat persediaan obat yang terdapat pada apotek, apabila di apotek terdapat barang selain itu maka barang tersebut tidak tercatat.
6. Peramalan jumlah obat hanya dapat diketahui untuk bulan berikutnya, tidak bisa dilakukan untuk meramalkan dua bulan 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.*
3. Merancang dan membangun sistem informasi persediaan obat pada apotek dengan metode EOQ (*Economic Order Quantity*).

## Manfaat Penelitian

Manfaat dari penelitian ini adalah:

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

# BAB II

# TINJAUAN PUSTAKA

## Studi Literatur

Sebelum melakukan penelitian ini, dilakukan 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 yang ada sebelumnya akan dijadikan 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 mengenai 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.

## 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).

## Persediaan

Persediaan merupakan suatu istilah yang menunjukkan segala sesuatu dari sumber daya yang ada 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 ada 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.

## 2.4 *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 ada 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.

Jumlah pesanan pada setiap pembelian (Q) yang optimal akan memperoleh total biaya persediaan (TC) yang minimal. Secara matematis, jumlah pesanan yang optimal (Q\*) dapat dihitung sebagai berikut (Royyan, 2015):



Atau



Persamaan untuk kuantitas pembelian optimal (Royyan, 2015):

EOQ = Q\* = √2 CR / H (1)

H = P × f (2)

B = RL / N (3)

Keterangan:

R : Permintaan perbulan (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 ada pesanan ulang (*back order*) karena kehabisan persediaan (*storage*).
6. Harga pembelian atau biaya pembuatan tidak berubah-ubah.
7. Tidak ada potongan harga (*quantity discount*).
8. Variabel biaya hanya biaya pesan (*ordering cost*) dan biaya simpan (*holding cost).*

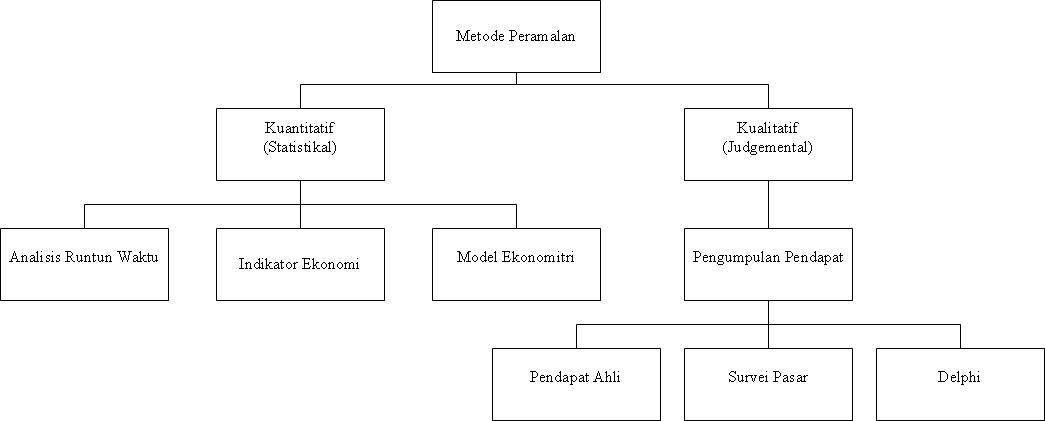
Menurut kartika hendra (2009), keunggulan dari metode EOQ adalah (Valerie & Sinuraya, 2011):

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 penganamann (*safety stock).*
3. Mudah diaplikasikan pada proses produksi secara massal.
4. Lazim digunakan pada rumah sakit, khususnya pada persediaan obat.

## 2.5Peramalan

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.

Gambar 2. 1 Klasifikasi Metode Peramalan (Sahara, 2013)

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 ada 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 bisa 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 ada 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 ada 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).

## 2.6 *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).

Metode ini adalah pengembangan dari metode *moving average* (MA) yang menggunakan rumus sebagai berikut (Sahara, 2013):

Ft+1 (4)

Keterangan:

Ft+1 = Ramalan untuk periode ke t + 1

XT = Nilai riil periode ke t

T = Jangka waktu rata-rata bergerak

Metode *moving average* memang mudah untuk dihitung akan tetapi metode ini memberikan bobot yang sama pada setiap data.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 (5)

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 bisa digunakan ukuran statistik untuk menghitung besarnya penyimpangan yaitu dengan menggunakan persamaan berikut (Nahar, 2013):

*Mean Square Error* (MSE) =  (6)

*Mean Absolute Error* (MAPE) =  (7)

Dengan nilai n adalah jumlah pengamatan.

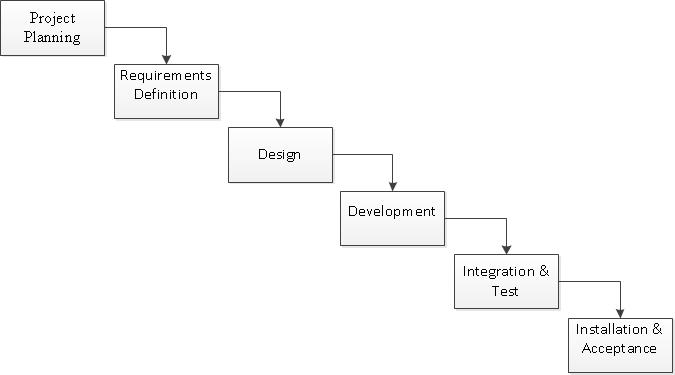
## 2.7 *SDLC (System Development Life Cycle)*

*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).

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

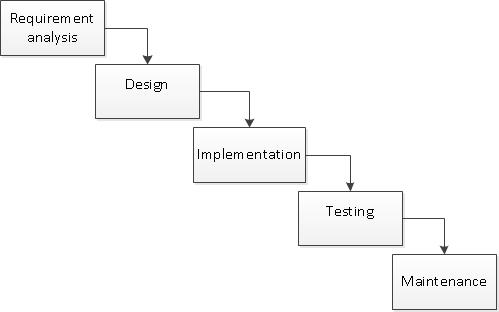
Tahapan tersebut dapat dijabarkan seperti gambar dibawah ini:



Gambar 2. 2 Software Development Life Cycle (SDLC) (Maheswari & Jain, 2012)

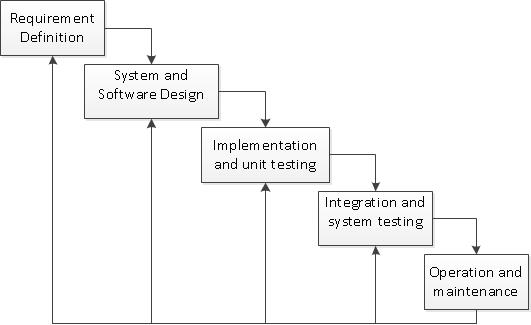
Menurut Shikha dan Dinesh (2012), ada empat tipe model dari SDLC, yaitu:

1. *Original Waterfall Model*

**

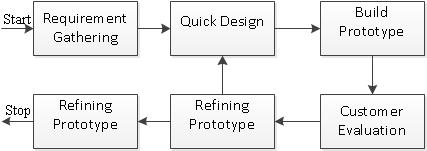
Gambar 2. 3 Original Waterfall Model (Maheswari & Jain, 2012)

1. *Iterative Waterfall*

**

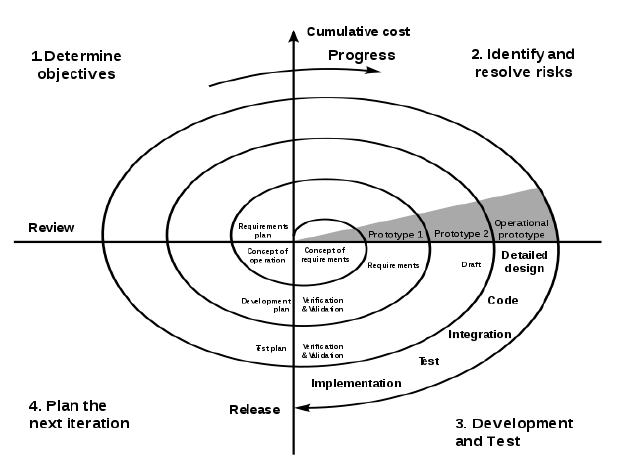
Gambar 2. 4 Iterative Waterfall model (Maheswari & Jain, 2012)

1. *Prototype*

**

Gambar 2. 5 Prototype model (Maheswari & Jain, 2012)

1. *Spiral*

**

Gambar 2. 6 Spiral Model (Maheswari & Jain, 2012)

Tabel 2. 1 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 |
| 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 |

## 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 ada 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*)

## 2.9 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. Di dalam teknik ini penguji harus memiliki pengetahuan tentang keseluruhan kode yang ada di dalamnya.
2. Teknik *Black Box Testing,* merupakan teknik dari pengujian yang tidak memiliki pengetahuan sama sekali terhadap pekerjaan yang ada di 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.

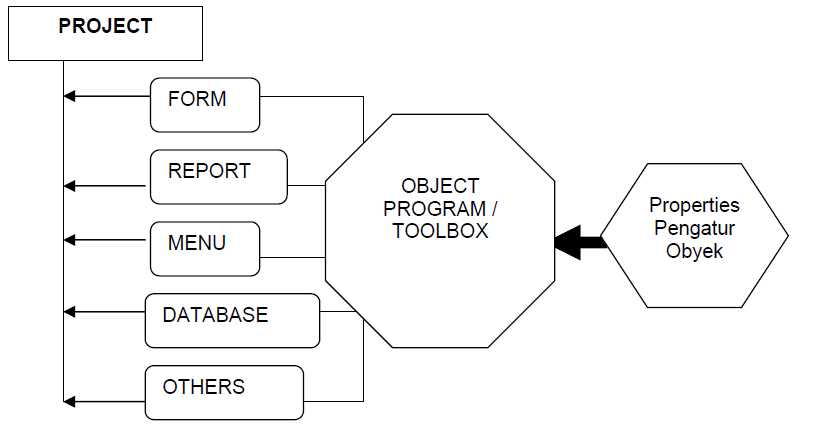
Tabel 2. 2 Perbandingan tiga teknik pengujian (Khan & Khan, 2012)

|  |  |  |
| --- | --- | --- |
| **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 |
| 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) |

## 2.10 *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).



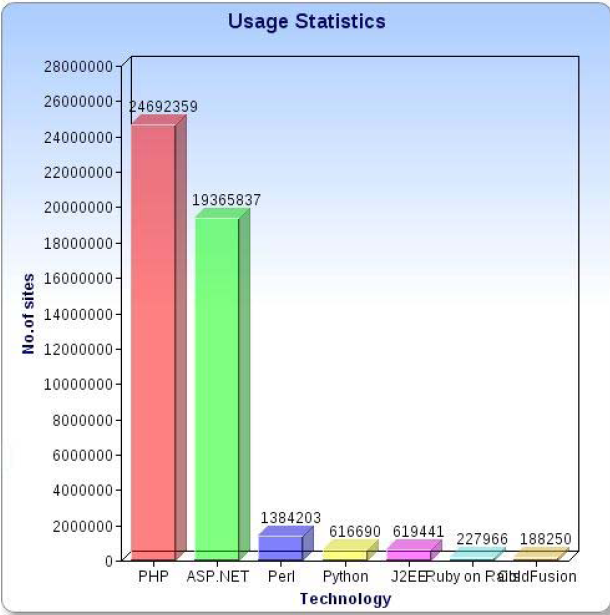
Gambar 2. 7 Model Pemrograman OOP (Danuri, 2009)

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.

## 2.11 Bahasa Pemrograman

Dalam pembuatan suatu sistem, diperlukan beberapa *tools* yang digunakan dalam merancang dan membangun sistem informasi berbasis web. Berikut ini akan di berikan beberapa *tools* yang dapat digunakan untuk pembuatan sistem informasi reservasi, check-in, check-out, dan inventaris data. 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. 8 Statistik Penggunaan Tools untuk Pengembangan Web (Nagilla, 2013)

Di dalam statistik terlihat bahwa PHP dan ASP.NET mendominasi penggunaan tools dalam pengembangan web. Berikut penjelasan secara singkat mengenai PHP dan ASP.NET:

* 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. 3 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 |

# 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 HP Probook 4421s.* 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.
* *Microsoft Visio 2010,* 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.
* *Adobe Dreamweaver CS3,* 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. Penentuan Target Penjualan.

Target penjualan ditentukan berdasarkan hasil peramalan pada tahun tersebut. Peramalan dilakukan dengan metode *exponential smoothing* dengan mengacu pada data penjualan pada tahun-tahun 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 berdasarkan metode yang telah diterapkan di apotek sebelumnya.

Tahapan ini dilakukan dengan menghitung semua biaya persediaan dan berdasarkan pada sistem yang sedang berjalan pada apotek sebelumnya. Perhitungan dilakukan dengan melihat jumlah pembelian barang setiap kali pembelian, dan frekuensi pembelian barang dalam jangka waktu setahun. Setelah itu dilakukan perhitungan biaya total persediaan pada tahun tersebut.

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 dan *safety stock.*

1. Perbandingan biaya total persediaan pada metode yang telah diterapkan pada apotek dengan metode baru yang menggunakan EOQ.

Pada tahapan ini dilakukan perbandingan biaya total persediaan yang didapat pada metode yang sedang diterapkan dengan biaya total persediaan yang didapat dengan menggunakan metode EOQ. Dari hasil perbandingan ini dapat ditarik kesimpulan tentang metode mana yang lebih efektif dan dapat menghemat biaya persediaan.

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:

### 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).

### 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.

### 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.

### 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.*

### 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. 5 Jadwal Penelitian

Adapun jadwal kegiatan dalam penelitian ini adalah seperti yang tertera pada tabel 3.1 berikut ini:

Tabel 3. 1 Jadwal Penelitian

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| No | Jenis Kegiatan | September | | | | Oktober | | | | November | | | | Desember | | | |
| 1 | Analisis dan definisi persyaratan |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2 | Perancangan sistem |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3 | Perancangan perangkat lunak |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4 | Implementasi unit |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 5 | Pengujian unit |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 6 | integrasi sistem |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 7 | pengujian sistem |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 8 | penulisan tugas akhir |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

# 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 thapan awal dalam penelitian ini yang termasuk ke dalam tahap analisis dan definisi sistem di dalam metode *Iterative Waterfall.* Pada tahap ini dilakukan obeservasi dan wawancara kepada pemilik Apotek Nadia Farma, yang berlokasi di Jl Ratu Dibalau, Tajung Seneng, 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 pemilknya. 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 ada 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.

## 4.3.2 Desain Database

Perancangan database dilakukan berdaasarkan analisis kebutuhan yang telah dilakukan pada proses analisis dan definisi sistem. Perancangan database msauk 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 membanngun sistem. Semua *entity* ysang diperoleh meweakili 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\_iitem | *Entity* yang berisi tentang data pembelian barang. Pada *entity* ini merinci pembelian obat yang terjadi di dalam apotek. |
| 5. | Penjualan\_item | *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. |

Entity tersebut merupakan 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 |

Berikut ini merupakan *conceptual database design* dari database yang dibangun berdasarkan hubungan yang telah ditentukan sebelumnya:

C:\Users\ThinkPad\Downloads\Untitled Diagram (1).png

Gambar 4 1 *Conceptual database design* dari sistem

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*

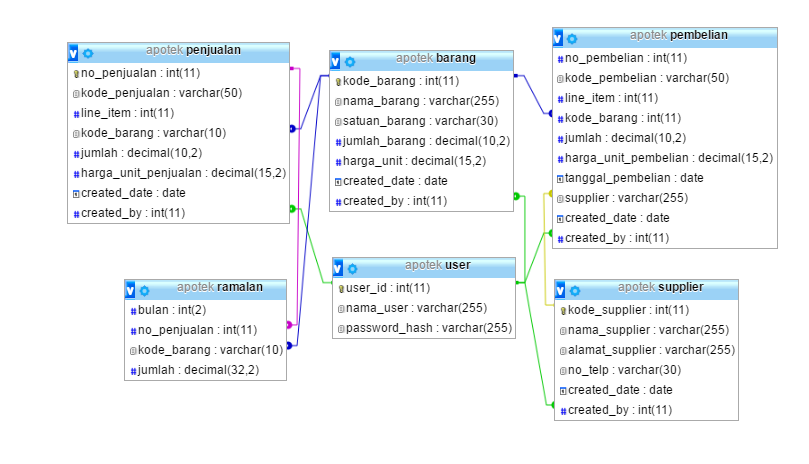
|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Nama *Entity*** | **Attribute** | **Keterangan** |
| 1 | *User* | user\_id | *Primary key* pada *entity user.* |
| nama\_user | Nama yang digunakan untuk autentikasi user. |
| password\_hash | Kata sandi yang digunakan untuk autentikasi pada sistem. |
| 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.* |
| created\_date | Informasi mengenai tanggal pembuatan data *supplier*  di dalam sistem. |
| created\_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. |
| jumlah\_barang | Informasi mengenai jumlah dari obat yang terdapat di gudang. |
| harga\_unit | Informasi mengenai harga satuan dari obat. |
| created\_date | Informasi mengenai tanggal pembuatan data obat di dalam sistem. |
| created\_by | informasi tentang siapa yang membuat data obat di dalam sistem. |
| 4 | Pembellian | no\_pembelian | *Primary key* pada *entity* pembelian*.* |
| kode\_pembelian | Informasi tentang kode pembelian obat. |
| line\_item | *Attribute* ini digunakan agar bisa 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. |
| created\_date | Informasi mengenai tanggal pembuatan data pembelian obat di dalam sistem. |
| created\_by | informasi tentang siapa yang membuat data pembelian obat di dalam sistem. |
| 5 | Penjualan | no\_penjualan | *Primary key* pada *entity* penjualan*.* |
| kode\_penjualan | Informasi tentang kode penjualan obat. |
| kode\_barang | Informasi mengenai kode barang, merupakan foreign key dari tabel barang. |
| line\_item | *Attribute* ini digunakan agar bisa 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. |
| created\_date | Informasi mengenai tanggal pembuatan data penjualan obat di dalam sistem. |
| created\_by | informasi tentang siapa yang membuat data penjualan obat di dalam sistem. |
| 6 | Ramalan | bulan | Berisi Informasi tentang bulan , karena jumlah penjualan nantinya akan dikelompokkan dan dijumlahkan setiap bulannya. |
| kode\_barang | Informasi mengenai kode barang, merupakan foreign key dari tabel barang. |
| no\_penjualan | Berisi informasi mengenai no penjualan dari masing-masing obat yang telah terjual. Merupakan *foreign key* dari tabel penjualan. |
| jumlah | Infromasi tentang jumlah penjualan obat pada bulan yang telah ditentukan. |

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.*

C:\Users\ThinkPad\Downloads\Untitled Diagram.png

Gambar 4 2 *Logical database design* dari sistem

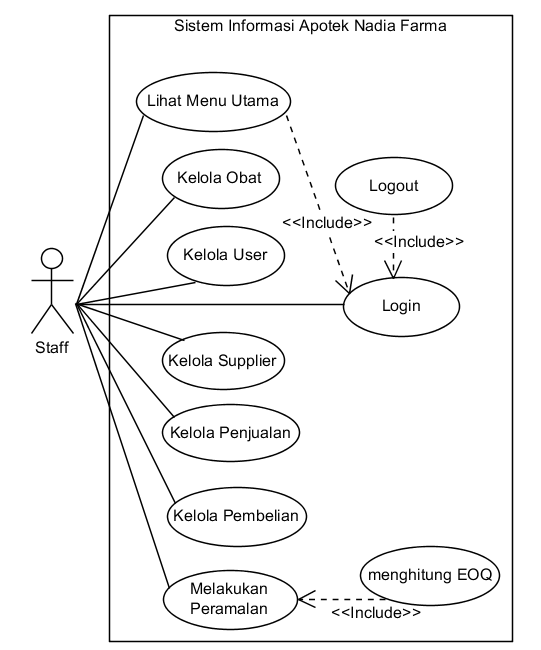
Berikut ini merupakan gambaran dari physical database design yang telah dilengkapi dengan tipe data yang digunakan pada masing-masing attribut. Skema ini merupakan desain yang sebenarnya digunakan dalam pembuatan sistem.



Gambar 4 3 Diagram *entity relationship diagram* pada sistem

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 empat buah tabel, yaitu tabel pembelian, penjualan, user, 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.

## 4.3.3 Use Case Diagram



Gambar 4 4 *Use case diagram* dari sistem

Penjelasan dari *use case diagram* diatas 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 passwordnya | |
| ***Post-condition*** | Sistem menampilkan menu utama | |

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 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 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 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 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 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 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 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. | |

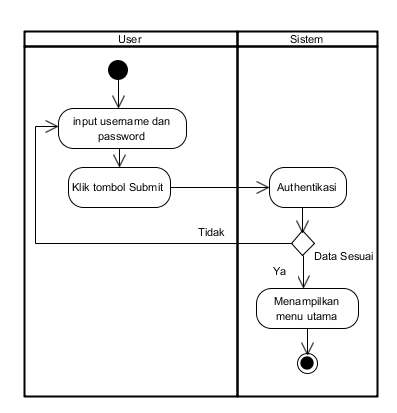
Tabel 4 13 Use Case Scenario Menghitung EOQ

|  |  |  |
| --- | --- | --- |
| ***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. | |

## 4.3.4 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

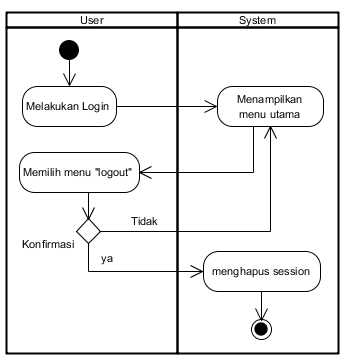
* Login



Gambar 4 5 Activity Diagram Login

Pada proses ini, 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.*

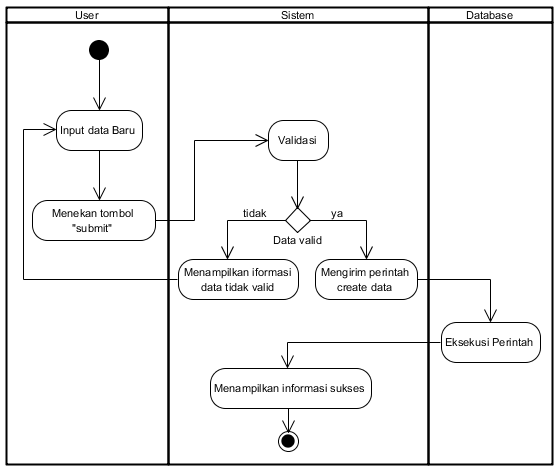
* Logout



Gambar 4 6 Activity Diagram Logout

Pada proses ini, 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.

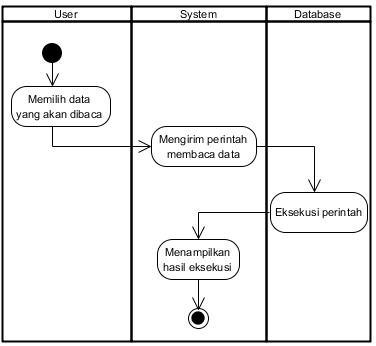
* Activity Diagram Create



Gambar 4 7 Activity Diagram Create

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 di eksekusi untuk melakukan penyimpanan di dalam database. Setelah dilakukan eksekusi pembuatan data baru, meka sistem akan menampilkan informasi bahwa data yang telah ditambahkan sukses.

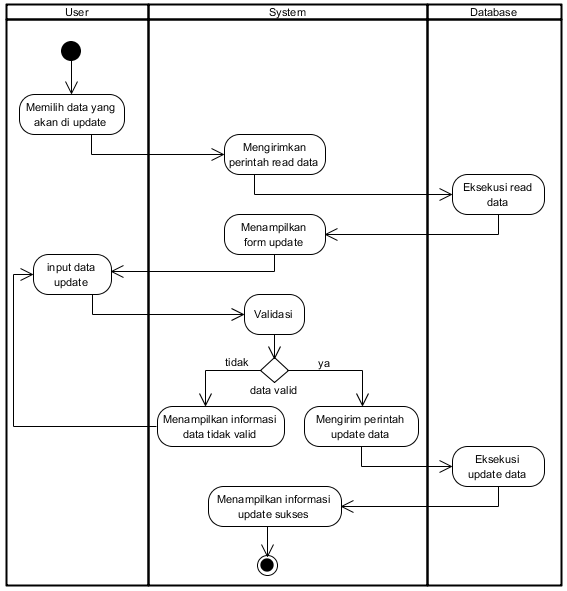
* Activity Diagram Read



Gambar 4 8 Activity Diagram Read

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.

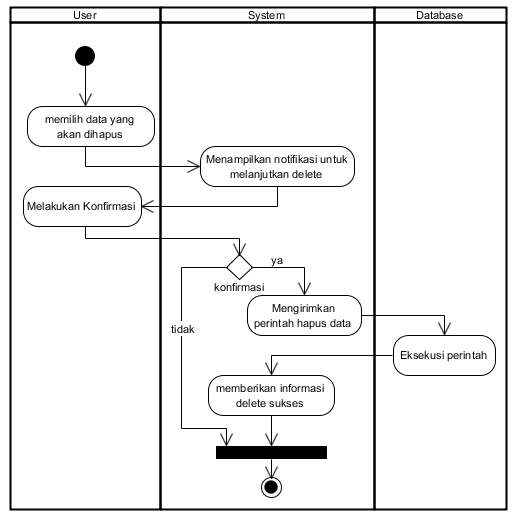
* Activity Diagram Update



Gambar 4 9 Activity Diagram Update

Pada proses ini, pertama-tama pengguna memilih data yang akan diupdate. Selanjutnya, sistem mengirimkan perintah kepada database untuk membaca data yang telah dipilih dan kemudian akan di eksekusi oleh database. Setelah dilakukan read pada data, maka sistem akan menampilkan form update yang berisi data-data yang telah di eksekusi. 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 mengupdate data yang telah dipilih. Setelah itu sistem akan menampilkan informasi bahwa update berhasil dilakukan.

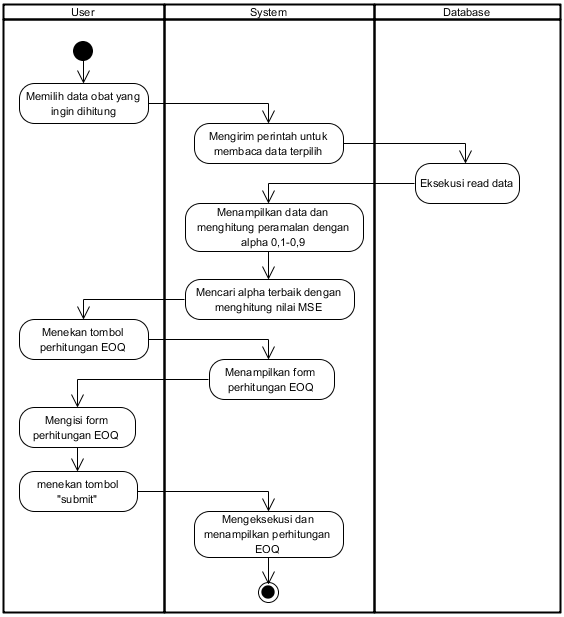
* Activity Diagram Delete



Gambar 4 10 Activity Diagram Delete

Pada proses ini, pengguna harus memilih data yang akan di hapus 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

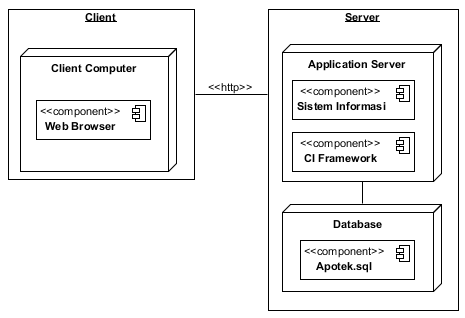


Gambar 4 11 Activity Diagram EOQ

Pada proses diatas, 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 peramaln, 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.

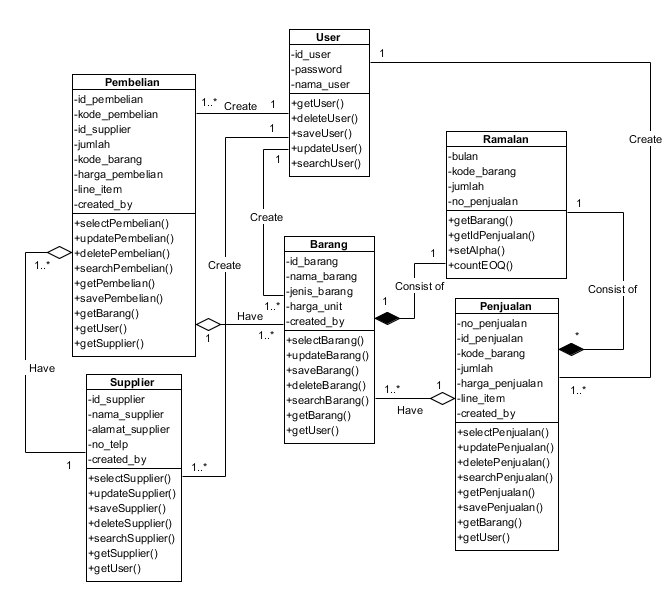
## 4.3.5 Deployment Diagram

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

**

Gambar 4 12 Deployment Diagram

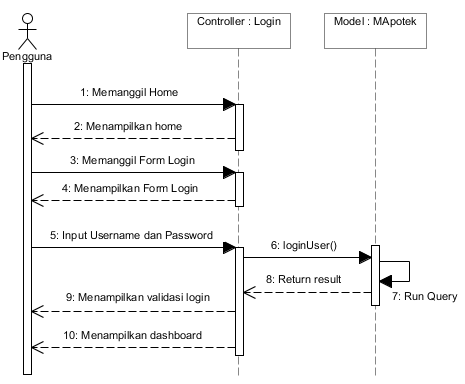
## 4.5.6 Class Diagram



Gambar 4 13 Class Diagram dari sistem

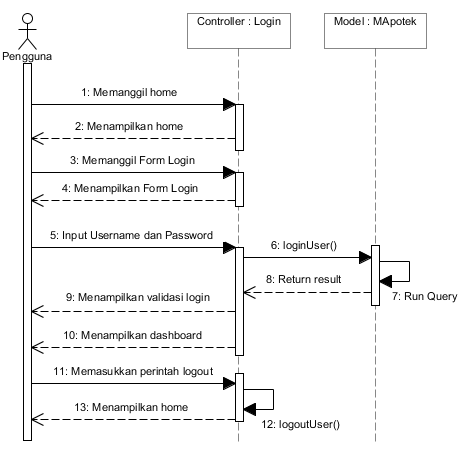
## 4.5.7 Sequence Diagram

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



Gambar 4 14 Sequence Diagram Login

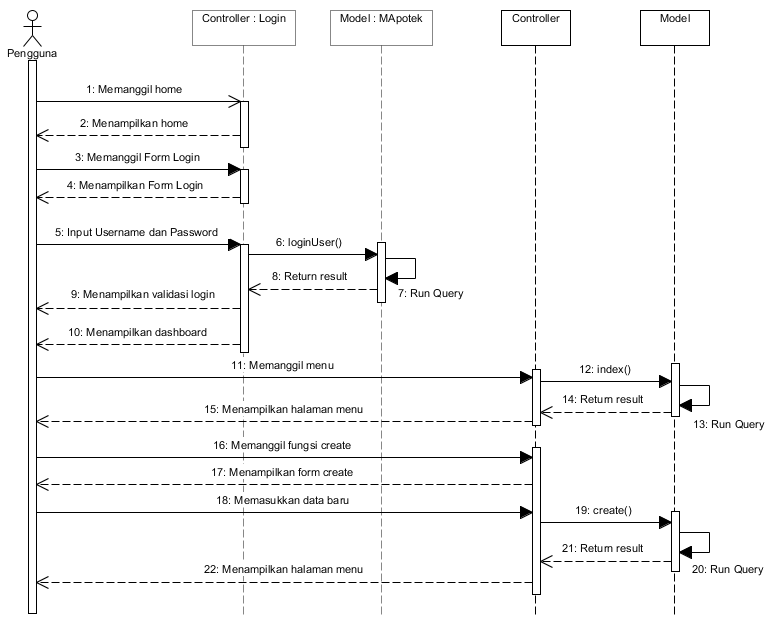
Gambar di atas merupakan *sequence diagram* dari *login* pengguna. Proses ini menggunakan *controller* Login.php sebagai logika kontrol dan MApotek.php untuk melakukan *query*.



Gambar 4 15 Sequence Diagram Logut

Gambar di atas 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.

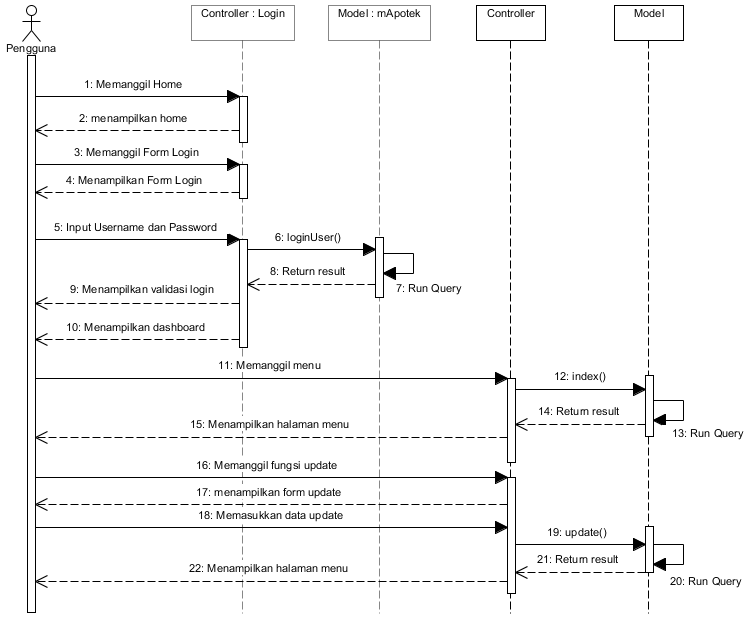


Gambar 4 16 Sequence Diagram Create

Gambar diatas 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.

Gambar 4 17 Tabel model, controller, dan fungsi create

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

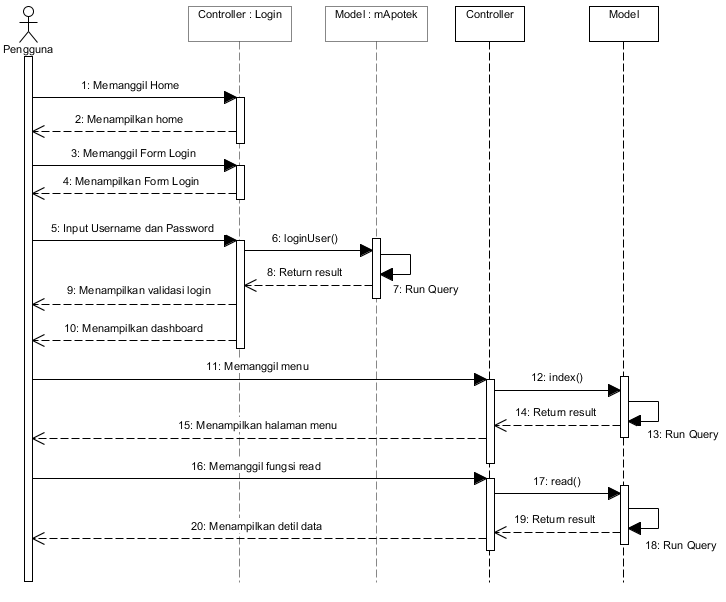


Gambar 4 18 Sequence diagram update

Gambar diatas 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.

Gambar 4 19 model, controllers, dan fungsi dari update

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

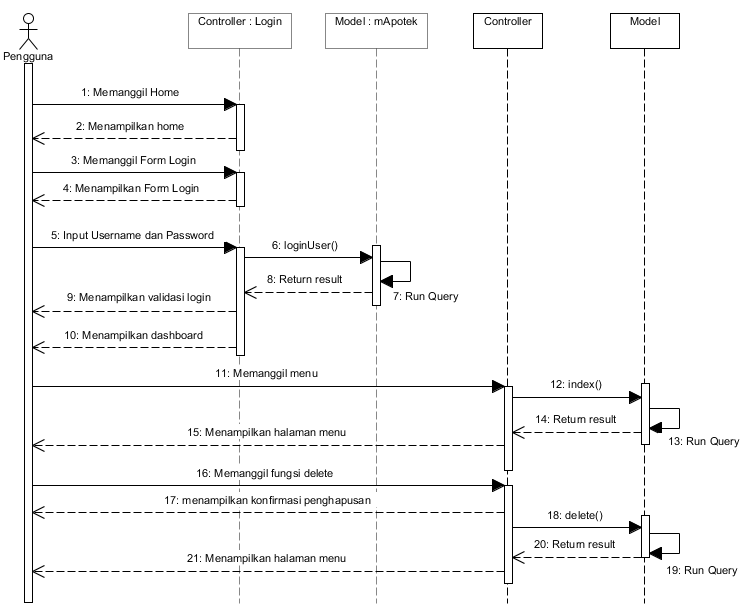


Gambar 4 20 Sequence Diagram read

Gambar diatas 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.

Gambar 4 21 model, controller, dan fungsi read

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

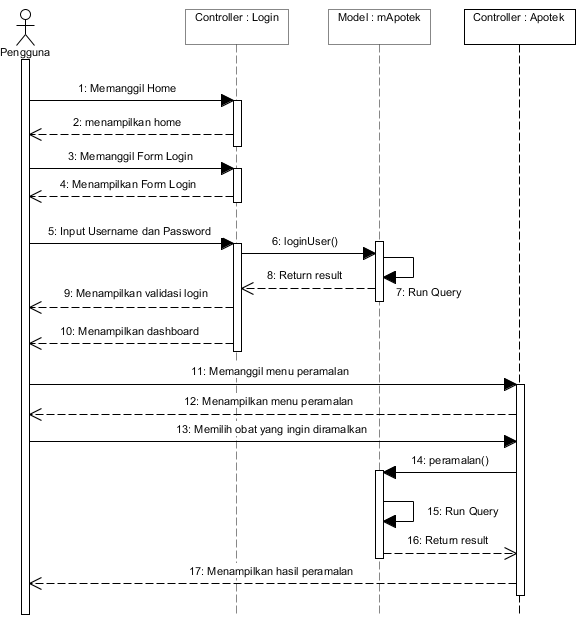


Gambar 4 22 Sequence Diagram Delete

Gambar diatas 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.

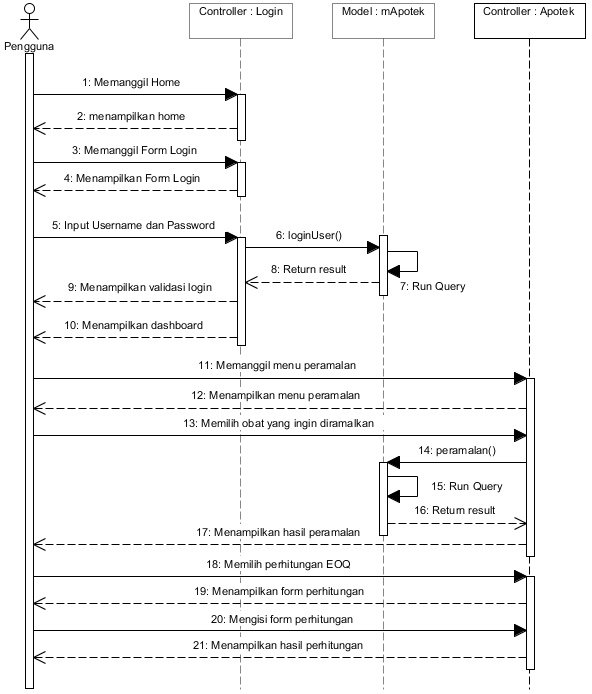
Gambar 4 23 model, controller, dan fungsi delete

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



Gambar 4 24 Sequence Diagram Peramalan

Gambar diatas 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.



Gambar 4 25 Sequence Diagram EOQ

Gambar diatas 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.

# DAFTAR PUSTAKA

Aprilian, Y., 2014. Sistem Informasi Penyewaan Kamar Pada Hotel Candra Kirana. *Indonesian Jurnal on Computer Science,* 11(1), pp. 20-25.

Ardhi, T., Tyromi, Y. & Sapto, B., 2013. Sistem Pelaporan dan Peramalan Penjualan Gula Dengan Mengimplementasikan Metode Exponential Smoothing Pada Teknologi ROLAP.

Damayanti, R., Eka, B. P. & Uly, I. W., n.d. Pembuatan Sistem Informasi Pemesanan Dan Pembayaran Kamar Pada Hotel Remaja Pacitan. *Indonesian Journal on Computer Science,* pp. 1-8.

Danuri, M., 2009. Object oriented Programming (OOP) Pembangun Program Aplikasi Berbasis Windows. *INFOKAM,* 5(1), pp. 40-47.

Facrurazzi, S., 2015. Peramalan Penjualan Obat Menggunakan Metode Single Exponential Smoothing Pada Toko Obat Bintang Geurogok. *TECHSI,* 6(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,* 2(1), pp. 317-331.

Gessong, Y., 2001. Analisis Persediaan Pada Furniture Sangasanga. *Eksis,* 8(1), pp. 2032-2035.

Himawan, H., 2014. Efektifitas Penggunaan Metode Exponential Smoothing Pada Peramalan Produk.

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

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,* 3(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.

Liliek, Y. P., 2012. Sistem Informasi Manajemen Agenda Pada Badan Pelayanan Perijinan Terpadu Kabupaten Karanganyar. *Indonesian Journal on Computer Science Speed,* 9(3), pp. 1-6.

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,* 2(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,* 12(1), pp. 326-334.

Mulyani, S., 2013. Analisis Pengendalian Intern Atas Persediaan Barang Dagangan Pada PT Grokindo.

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.

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.

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),* 1(1), pp. 1-7.

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,* 5(2).

Wibisetiadi, Y., 2011. Rancang Bangun Sistem Informasi Apotek Sinar. pp. 1-12.

Wijaya, A., 2014. Analisis Pengendalian Internal Persediaan Obat Pada Apotek Kencana 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). *SENTIKA,* p. 80.