

BAB II

Landasan Teori

2.1 Tinjauan Pustaka

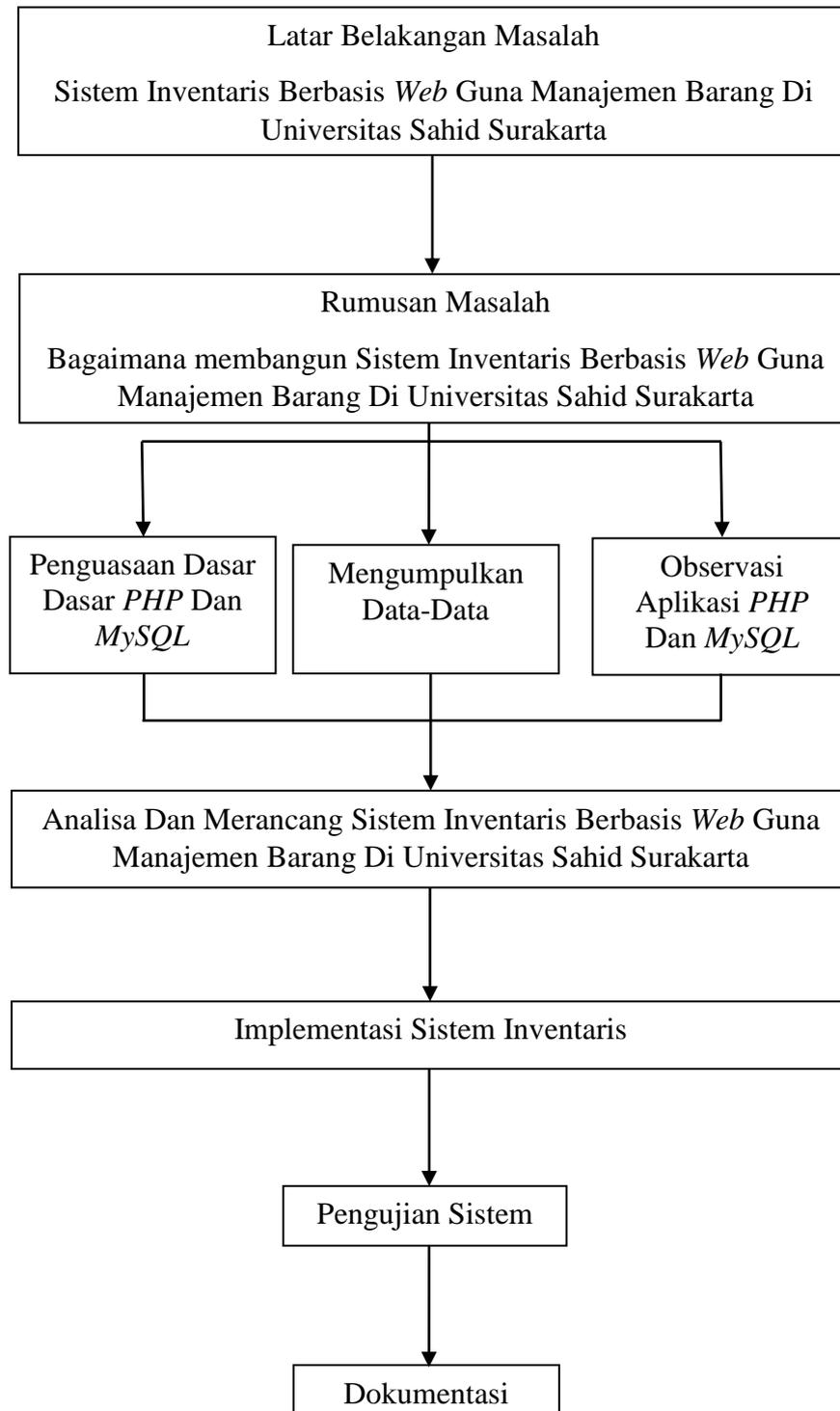
Rahmadi, dkk (2014). membuat sebuah perancangan sistem informasi inventori pada AMIK Lembah Dempo Pagaralam yang nantinya hasil dari perancangan ini dapat dipakai ataupun dikembangkan untuk pembangunan sistem informasi inventaris di AMIK Lembah Dempo. Dengan menggunakan sistem informasi inventaris dapat membantu dan mempermudah proses pengelolaan data inventaris pada AMIK Lembah Dempo. Beban kerja hanya dilakukan satu kali tanpa perlu melakukan proses pembukuan terlebih dahulu sebelum meng-*input*-kan data. Pengambilan data dan informasi jadi lebih cepat dibandingkan dengan sistem manual yang lama. Data ataupun informasi tersimpan rapi di dalam *database* yang terjamin keamanannya, tidak bisa diakses oleh siapa saja dan dapat dipanggil kapan saja.

Nugrahanti,(2015). membuat suatu aplikasi program yang dapat melakukan kontrol persediaan *sparepart* mesin *fotocopy* dan memberikan laporan *update* stok dan juga *inventory sparepart* mesin *fotocopy* ini memudahkan karyawan bagian administrasi dalam mengolah data sehingga dapat meminimalisir semua kemungkinan dalam manipulasi data dan kesalahan pencatatan.

Rahmad, dkk (2014). membuat perangkat lunak sistem informasi inventori *sparepart* elektronik berbasis *Web PHP* dengan kemampuan dapat menyimpan data penjualan dan pembelian barang, mengontrol stok barang masuk dan keluar gudang, dapat mencatat penggunaan biaya inventaris yang ada.

2.2 Kerangka Pemikiran

Berikut ini adalah tahapan kerangka pemikiran yang akan dijalankan oleh penulis dalam membangun sistem *inventaris* barang di Universitas Sahid Surakarta berbasis *web* seperti pada Gambar 2.1.



Gambar 2.1 Kerangka Pemikiran

Uraian dari kerangka berfikir sebagai berikut :

1. Latar Belakang Masalah

Latar belakang masalah pada tugas akhir ini adalah belum adanya sistem inventaris barang di Universitas Sahid Surakarta menyebabkan kurangnya efisiensi waktu dalam pendataan barang yang lebih cepat dan tepat.

2. Rumusan Masalah

Rumusan masalah pada tugas akhir ini adalah bagaimana cara membuat sistem inventaris barang di Universitas Sahid Surakarta berbasis *website* yang dapat memberikan informasi dan pelayanan di bagian umum.

3. Pengumpulan Data

Pengumpulan data dalam penelitian ini ada dua yaitu pengumpulan data tertulis dan pengumpulan data tidak tertulis. Penelitian ini mengumpulkan semua data yang dibutuhkan dengan melakukan *interview* dengan bagian umum di Universitas Sahid Surakarta.

4. Penguasaan Dasar

Penguasaan dasar dalam penelitian ini adalah penguasaan yang telah diketahui atau dikuasai penulis mengenai *PHP* dan *MySQL*.

5. Observasi Aplikasi

Observasi alat dalam penelitian ini adalah penguasaan terhadap *software-software* yang akan digunakan untuk membuat sistem informasi pengajuan anggaran kegiatan di Universitas Sahid Surakarta.

6. Analisis dan Perancangan Sistem

Analisis dan perancangan sistem dalam penelitian ini adalah memahami dengan detail, apa yang harus dilakukan untuk sistem yang akan dibuat serta merancang sistem yang baru.

7. Implementasi Sistem

Implementasi sistem dalam penelitian ini adalah proses pelaksanaan penelitian, setelah dilakukan analisis dan perancangan kemudian dilakukan pembuatan sistem inventaris.

8. Pengujian Sistem

Pengujian sistem dalam penelitian ini adalah dilakukan untuk mengetahui jika ternyata masih ada kesalahan atau kekurangan pada sistem yang dibuat.

9. Penerapan Sistem dan Dokumentasi

Penerapan sistem dan dokumentasi dalam penelitian ini adalah proses menerapkan aplikasi di Universitas Sahid Surakarta dan proses pengambilan dokumentasi setelah sistem selesai dibuat dari keseluruhan kegiatan Tugas Akhir.

2.3 Dasar Teori

2.3.1 Sistem

Sistem adalah sebuah tatanan (keterpaduan), sekelompok unsur atau elemen yang berhubungan satu dengan yang lain untuk mencapai suatu tujuan (Rahmawati, 2017).

2.3.2 Website

Website merupakan kumpulan halaman-halaman yang berisi informasi yang disimpan di internet yang bisa diakses atau dilihat melalui jaringan internet pada perangkat-perangkat yang bisa mengakses internet itu sendiri seperti komputer. Definisi kata *web* adalah *web* sebenarnya penyederhanaan dari sebuah istilah dalam dunia komputer yaitu *World Wide Web* yang merupakan bagian dari teknologi internet (Hastanti, 2015).

World Wide Web atau disingkat dengan nama *www*, merupakan sebuah sistem jaringan berbasis *Client-Server* yang mempergunakan protokol *HTTP* (*Hypertext Transfer Protocol*) dan *TCP/IP* (*Transmission Control Protocol / Internet Protocol*) sebagai medianya. Karena kedua sistem ini mempunyai hubungan yang sangat erat, maka untuk saat ini sulit untuk membedakan antara *HTTP* dengan *WWW*.

Internet dapat diartikan sebagai jaringan komputer yang luas dan besar yang mendunia, yaitu menghubungkan pemakai komputer dari negara ke negara di seluruh dunia. Pada awalnya internet atau *web* hanya dipergunakan untuk kepentingan militer yaitu suatu teknologi yang dipergunakan untuk mengirimkan pesan melalui satelit. Akan tetapi lama kelamaan teknologi tersebut akhirnya

meluas, dan bahkan internet pada saat ini sudah sama populernya dengan *telephone*. Informasi yang dikirimkan lewat internet dapat diakses ke seluruh dunia hanya dalam hitungan menit bahkan detik.

Teknologi yang digunakan menjadi sangat populer dan cepat sekali perkembangannya. Saat ini internet sudah tidak menjadi istilah yang asing lagi. Suatu informasi yang dikirimkan lewat internet dapat berupa teks, gambar maupun multimedia sehingga internet juga dimanfaatkan oleh perusahaan-perusahaan untuk mempromosikan produk-produknya dengan cepat dan mudah.

2.3.3 PHP (*Personal Home Page*)

PHP merupakan bahasa pemrograman pelengkap *HTML* (*Hypertext Markup Language*) yang memungkinkan aplikasi *web* dinamis untuk pengolahan data, pemrosesan data dari *user* via *form*, membuat buku tamu, toko *online*, dan lain sebagainya, dengan mudah *PHP* dapat melakukan koneksi ke *database* karena *PHP* memang dilengkapi fitur yang memungkinkan koneksi ke *PHP* dilakukan dengan mudah, tanpa harus melakukan pemrograman yang memusingkan. *PHP* juga merupakan bahasa pemrograman berbasis *server side* yang dapat melakukan parsing *script PHP* menjadi *script web* sehingga dari sisi *client* menghasilkan suatu tampilan yang menarik.

Jadi dapat disimpulkan bahwa pengertian *Personal Home Page* (*PHP*) adalah bahasa pemrograman pelengkap *HTML* berbasis *server side* yang memungkinkan aplikasi *web* dinamis, dapat melakukan koneksi ke *database* dan menghasilkan suatu tampilan yang menarik (Rahmawati, 2017).

2.3.4 HTML (*Hypertext Markup Language*)

HTML adalah sebuah bahasa markah yang digunakan untuk membuat sebuah halaman *web*, menampilkan berbagai informasi di dalam sebuah penjelajah *web* internet dan pemformatan *hypertext* sederhana yang ditulis dalam berkas format *ASCII* agar dapat menghasilkan tampilan wujud yang terintegrasi.

HTML adalah bahasa *markup* untuk menstrukturkan dan menampilkan isi dari *World Wide Web*, sebuah teknologi inti dari internet.

Jadi dapat disimpulkan bahwa pengertian *Hypertext Markup Language* (*HTML*) adalah sebuah bahasa *markup* untuk membuat sebuah halaman *web* yang

menampilkan berbagai informasi untuk menstrukturkan dan menampilkan isi dari *World Wide Web* pada *browser* yang ditulis dalam berkas format *ASCII* (Rahmawati, 2017).

2.3.5 MySQL

Berbagai definisi tentang *MySQL* yang dikutip dari beberapa sumber yang dapat dilihat pada Tabel 2.1 untuk memberikan penjelasan lebih lanjut tentang *MySQL* (Rahmawati, 2017).

Tabel 2.1 Definisi *MySQL*

Sumber	Definisi
Faizal & Irnawati (2015:4)	<i>MySQL</i> adalah perangkat lunak sistem manajemen basis data <i>SQL</i> (bahasa Inggris : <i>database management system</i>) atau <i>DBMS</i> yang <i>multithread</i> , <i>multi-user</i> . <i>MySQL</i> menggunakan <i>SQL</i> (<i>Structure Query Language</i>) sebagai bahasa dasar untuk mengakses <i>database</i> .
Raharjo (2015:16)	<i>MySQL</i> merupakan <i>software RDBMS</i> (atau <i>server database</i>) yang dapat mengelola <i>database</i> dengan sangat cepat, dapat menampung data dalam jumlah sangat besar, dapat diakses oleh banyak user (<i>multi-user</i>) dan dapat melakukan suatu proses secara sinkron atau berbarengan (<i>multi-threaded</i>).
Nugroho (2014: 31)	<i>MySQL</i> adalah <i>software</i> atau program aplikasi <i>database</i> , yaitu <i>software</i> yang dapat dipakai untuk menyimpan data berupa informasi, teks dan juga angka.
Ardhana (2014:46)	<i>MySQL</i> adalah sebuah perangkat lunak sistem manajemen basis data <i>SQL</i> (<i>database management system</i>) atau <i>DBMS</i> yang <i>multithread</i> , dan <i>multi-user</i> .

Dari berbagai uraian pada Tabel 2.1 dapat disimpulkan bahwa pengertian *MySQL* adalah sebuah perangkat lunak sistem manajemen basis data *SQL* atau *DBMS software* yang dapat dipakai untuk menyimpan data berupa informasi, teks dan juga angka.

2.3.6 XAMPP

Xampp merupakan paket PHP yang berbasis *Open Source* yang dikembangkan oleh sebuah komunitas *Open Source*. Penggunaan perangkat lunak *XAMPP* diawali dengan *install* paket *XAMPP* pada halaman resmi. Tersedia beberapa *update* yang dapat di-*download* sesuai dengan *platform* komputer pengguna. Setelah penginstalan selesai maka pengguna dapat memulai pemrograman dengan membuka *XAMPP Control Panel* terlebih dahulu untuk mengaktifkan *service* yang disediakan seperti : Apache, *MySQL*, *FileZilla*, *Mercury* dan *Tomcat* dengan mengklik *Action : Star* (Rahmawati, 2017).

2.3.7 *Bootstrap*

Bootstrap sebuah alat bantu untuk membuat sebuah tampilan halaman *website* yang dapat mempercepat pekerjaan seorang pengembang *website* ataupun pendesain halaman *website*. Sesuai namanya, *website* yang dibuat oleh alat bantu ini memiliki tampilan halaman yang sama atau mirip dengan tampilan halaman *twitter* atau desainer juga dapat mengubah tampilan halaman *website* sesuai dengan kebutuhan. Tampilan *website* yang dibuat *bootstrap* akan menyesuaikan ukuran layar dari *browser* yang kita gunakan baik *dekstop*, *tablet* ataupun *mobile device*. Fitur ini bisa diaktifkan ataupun di-*non*-aktifkan sesuai keinginan. Sehingga, membuat *web* untuk tampilan *dekstop* saja dan apabila di-*reader* oleh *mobile browser* maka tampilan dari *web* yang dibuat sesuai layar. Dengan *bootstrap* kita juga bisa membangun *web* dinamis ataupun statis (Widyantoro, 2014).

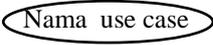
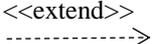
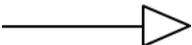
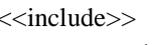
2.3.8 *UML*

Pada perkembangan teknologi perangkat lunak diperlukan adanya bahasa yang digunakan untuk memodelkan perangkat lunak yang akan dibuat dan perlu adanya standarisasi agar orang di berbagai negara dapat mengerti pemodelan perangkat lunak. Pada perkembangan teknik pemrograman berorientasi objek, muncullah sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemrograman berorientasi objek, yaitu *Unified Modelling Language* (Salahudin dan Sukamto, 2016).

2.3.8.1 *Use Case Diagram*

Use case atau *diagram use case* merupakan pemodelan untuk melakukan sistem informasi yang akan dibuat. *Diagram use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. (Salahudin dan Sukamto, 2016). Simbol-simbol yang ada pada diagram *use case* dapat dilihat pada Tabel 2.2.

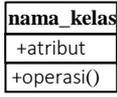
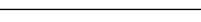
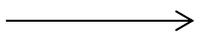
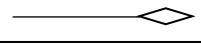
Tabel 2.2 Simbol-simbol *Use Case Diagram*

NO.	SIMBOL	NAMA	KETERANGAN
1.		<i>Use case</i>	Fungsional yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan kata kerja di awal di awal frase nama <i>use case</i> .
2.		<i>Actor</i>	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.
3.		<i>Association</i>	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.
4.		<i>Extend</i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu.
5.		<i>Generalization</i>	Hubungan generalisasi dan spesialisasi (umum – khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
6.		<i>Include</i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> di mana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini.

2.3.8.2 Class Diagram

Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas diagram memiliki atribut dan metode operasi. (Salahudin dan Sukamto, 2016). Simbol-simbol yang ada pada *class diagram* dapat dilihat pada Tabel 2.3.

Tabel 2.3 Simbol-simbol *Class Diagram*.

NO.	SIMBOL	NAMA	KETERANGAN
1.		<i>Class</i>	Kelas pada struktur sistem
2.		<i>Interface</i>	Sama dengan konsep interface dalam pemrograman berorientasi objek.
3.		<i>Association</i>	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
4.		<i>Directed association</i>	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain biasanya juga disertai dengan <i>multiplicity</i> .
5.		<i>Generalization</i>	Relasi antar kelas dengan makna generalisasi - spesialisasi (umum - khusus).
6.		<i>Dependency</i>	Relasi antar kelas dengan makna kebergantungan antar kelas.
7.		<i>aggregation</i>	Relasi antar kelas dengan makna semua bagian (<i>whole-part</i>).

2.3.8.3 Sequence Diagram

Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek (Salahudin dan Sukamto, 2016). Simbol-simbol yang ada pada *sequence* diagram dapat dilihat pada Tabel 2.4.

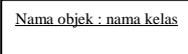
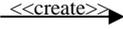
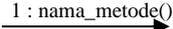
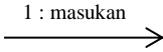
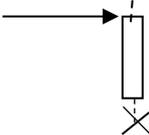
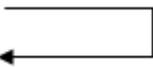
2.3.8.4 Activity Diagram

Diagram aktivitas atau *Activity diagram* menggambarkan *workflow* atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor (Salahudin dan Sukamto, 2016). Simbol-simbol yang ada pada *activity* diagram dapat dilihat pada Tabel 2.5.

2.3.8.5 Component Diagram

Diagram komponen *atau component diagram* dibuat untuk menunjukkan organisasi dan ketergantungan diantara kumpulan komponen dalam sebuah sistem (Salahudin dan Sukamto, 2016). Simbol-simbol yang ada pada *component diagram* dapat dilihat pada Tabel 2.6.

Tabel 2.4 Simbol-simbol *Sequence Diagram*.

NO.	SIMBOL	NAMA	KETERANGAN
1.		Aktor	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.
2.		<i>Lifeline</i>	Menyatakan kehidupan suatu objek.
3.		Objek	Menyatakan objek yang berinteraksi pesan.
4.		Waktu aktif	Menyatakan objek dalam keadaan aktif dan berinteraksi.
5.		Pesan tipe create	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat
6.		Pesan tipe call	Menyatakan suatu objek memanggil operasi / metode yang ada pada objek lain atau dirinya sendiri.
7.		Pesan tipe send	Menyatakan bahwa suatu objek mengirimkan data / masukan / informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.
8.		Pesan tipe <i>destroy</i>	Menyatakan suatu objek mengakhiri hidup objek lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada <i>create</i> maka ada <i>destroy</i> .
9.		Pesan tipe <i>return</i>	Menyatakan bahwa suatu objek telah menjalankan suatu operasi atau metode menghasilkan suatu pengembalian ke objek tertentu, arah panah ke objek kembalian

Tabel 2.5 Simbol-simbol *Activity Diagram*.

NO	GAMBAR	NAMA	KETERANGAN
1		Status awal	Status awal aktivitas sistem.
2		Aktivitas	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3		<i>Decision</i>	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
4		<i>Join</i>	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
5.		Status akhir	Status akhir yang dilakukan sistem.
6.		<i>Swimlane</i>	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.
7.		<i>Fork</i>	Asosiasi percabangan dimana lebih dari satu aktivitas cabangkan menjadi satu.
8.		<i>Note</i>	Note adalah diagram diagram yang tidak memiliki pengaruh semantik pada elemen model.

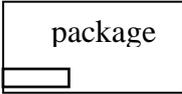
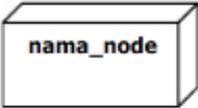
Tabel 2.6 Simbol – simbol *Component Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1.		<i>package</i>	<i>Package</i> merupakan sebuah bungkusan dari satu atau lebih komponen
2.		<i>Component</i>	Komponen sistem
3.		<i>Dependency</i>	Kebergantungan antar komponen, arah panah mengarah pada komponen yang dipakai
4.		<i>interface</i>	Sama dengan konsep <i>interface</i> pada pemrograman berorientasi objek, yaitu sebagai antarmuka komponen agar tidak mengakses langsung komponen
5.		<i>Link</i>	Relasi antar komponen

2.3.8.6 Deployment Diagram

Deployment diagram menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi. Diagram *deployment* juga dapat digunakan untuk memodelkan sistem tambahan, sistem *client*, sistem terdistribusi murni, rekayasa ulang aplikasi (Salahudin dan Sukamto, 2016). Simbol-simbol yang ada pada *deployment* diagram dapat dilihat pada Tabel 2.7.

Tabel 2.7 Simbol-Simbol *Deployment Diagram*

NO	SIMBOL	NAMA	KETERANGAN
1.		<i>Package</i>	<i>Package</i> merupakan sebuah bungkusan dari satu atau lebih <i>node</i> .
2.		<i>Node</i>	Biasanya mengacu pada perangkat keras (<i>hardware</i>), perangkat lunak yang tidak dibuat sendiri (<i>software</i>), jika didalam <i>node</i> disertakan komponen untuk mengkonsistenkan rancangan maka komponen yang diikutsertakan harus sesuai dengan komponen yang telah didefinisikan sebelum pada diagram komponen.
3.		<i>Dependency</i>	Kebergantungan antar <i>node</i> , arah panah mengarah pada <i>node</i> yang dipakai.
4.		<i>Link</i>	Relasi antar <i>node</i> .

2.3.9 Inventaris / (Inventory)

Berbagai definisi tentang persediaan (*inventory*) yang dikutip dari beberapa sumber yang dapat dilihat pada Tabel 2.8 untuk memberikan penjelasan lebih lanjut tentang sistem.

Tabel 2.8 Definisi Persediaan (*inventory*)

Sumber	Definisi
Assauri (2016:225)	Persediaan (<i>inventory</i>) adalah stok dari suatu item atau sumber daya yang digunakan dalam suatu organisasi perusahaan.
Martono (2015:210)	<i>Inventori</i> (sediaan) yang merupakan semua jenis barang yang dimiliki perusahaan dan digunakan untuk mendukung proses bisnisnya.

Dari berbagai uraian pada Tabel 2.8 dapat disimpulkan bahwa pengertian persediaan (*inventory*) merupakan stok dari suatu item atau sumber daya, bahan, material yang disimpan dan disediakan oleh suatu perusahaan untuk mendukung proses bisnisnya dan memenuhi permintaan dari konsumen atau pelanggan setiap waktu.

2.3.9.1 Fungsi dan Jenis-jenis Persediaan (*Inventory*)

Setiap organisasi perusahaan selalu berupaya untuk menjamin terdapatnya kelancaran operasi produksinya. Dalam upaya ini, perusahaan mengadakan inventaris yang tetap selalu ada. Inventaris yang diadakan perusahaan adalah dalam berbagai jenis, yang akan diuraikan berikut ini (Rahmawati, 2017):

1. Fungsi dari Inventaris

Inventaris dapat memberikan beberapa fungsi, yang akan menambah fleksibilitas operasi produksi suatu perusahaan. Sejumlah fungsi yang diberikan inventaris di antaranya adalah :

- a. Untuk dapat memenuhi antisipasi permintaan pelanggan, dimana inventaris merupakan upaya antisipasi stok, karena diharapkan dapat menjaga terdapatnya kepuasan yang diharapkan pelanggan.
- b. Untuk memisahkan berbagai *parts* atau komponen dari operasi produksi, sehingga dapat dihindari hambatan dari adanya fluktuasi, karena telah adanya inventaris ekstra guna memisahkan proses operasi produksi dengan pemasok.
- c. Untuk memisahkan operasi perusahaan dari fluktuasi permintaan, dan memberikan suatu stok barang yang akan memungkinkan dilakukannya pendekatan oleh pelanggan. Inventaris itu merupakan jenis upaya membangun ritel.
- d. Inventaris berfungsi untuk memperlancar keperluan operasi produksi, dimana inventaris dapat membangun kepercayaan dalam menghadapi terjadinya pola musiman, sehingga inventaris ini disebut sebagai inventaris musiman.

- e. Untuk dapat memanfaatkan diskon kuantitas, karena dilakukannya pembelian dalam jumlah besar, sehingga mungkin dapat mengurangi biaya barang atau biaya *delivery*-nya.
- f. Untuk memisahkan operasi produksi dengan dengan kejadian atau *event*, dimana inventaris digunakan sebagai penyangga di antara keberhasilan operasi produksi. Dengan demikian, kontinuitas operasi produksi dapat terjaga, dan dapat dihindari terdapatnya kejadian kerusakan peralatan, yang menyebabkan operasi produksi terhenti secara temporer.
- g. Untuk melindungi kekurangan stok yang dihadapi perusahaan, karena terlambatnya kedatangan *delivery* dan adanya peningkatan permintaan, sehingga kemungkinan terdapatnya resiko kekurangan pasokan.
Untuk memagari terhadap inflasi, dan meningkatnya perubahan harga.
Untuk memanfaatkan keuntungan dari siklus pesanan, dengan cara meminimalisasi pembelian, dan biaya persediaan, yang dilakukan dengan membeli dalam jumlah yang melebihi jumlah kebutuhan segera.
- h. Untuk memungkinkan perusahaan beroperasi dengan penambahan barang segera, seperti menggunakan barang yang sedang dalam proses

2. Jenis-jenis Inventaris

Menjalankan fungsi inventaris, perusahaan-perusahaan umumnya menjaga adanya empat jenis *inventory*. Keempat jenis *inventory* itu adalah: (1) bahan baku, (2) *inventory* dari barang dalam proses dikerjakan, (3) *inventory maintenance/repair/operating supplier* (MROs), (4) *inventory* barang jadi (Rahmawati, 2017).

Inventaris bahan baku dibeli dalam keadaan belum diperoses. Inventaris ini digunakan secara terpisah pasokannya dari proses produksi. Dalam penanganan inventaris bahan baku, umumnya pendekatan yang lebih disukai adalah menghilangkan perbedaan dari pemasoknya dalam kualitas, kuantitas, atau waktu deliverinya, sehingga tidak perlu dipisah-pisahkan.

Inventaris barang dalam proses atau *Work-In-Process* (WIP) adalah komponen atau bahan baku yang sedang dalam proses pengerjaan, tetapi belum selesai. WIP ada karena dari waktu yang telah digunakan dalam proses, yang berkaitan dengan produk dalam pembuatannya, disebut siklus atau *cycle time*.

Maintenance/repair/operating supplier (MROs) adalah mencurahkan untuk perlengkapan *maintenance/repair/operating* yang dibutuhkan, agar dapat terjaga mesin-mesin dan proses dapat produktif. MROs ini ada, karena terdapatnya kebutuhan dan waktu untuk perawatan dan perbaikan dari peralatan, adalah tidak dapat diketahui.

Inventaris barang jadi adalah produk yang sudah selesai diproses dan menunggu pengiriman. Barang jadi diinventorikan, karena permintaan dari para pelanggan pada masa depan adalah tidak dapat diketahui.

Secara umum, persediaan dapat dibedakan dalam beberapa jenis, antara lain sebagai berikut (Rahmawati, 2017):

- a. Persediaan bahan baku (*raw material*) atau yang disebut juga persediaan bahan mentah, yaitu bahan atau barang yang akan diproses lebih lanjut menjadi barang jadi
- b. Persediaan barang dalam proses (*work in process inventory*), merupakan persediaan yang telah mengalami perubahan, tetapi belum selesai.
- c. *Supplies inventory* adalah persediaan yang berfungsi sebagai penunjang dalam proses operasi atau produksi agar berjalan lancar.
- d. Persediaan barang dagangan (*merchandise inventory*), merupakan persediaan yang akan dijual kembali sebagai barang dagangan.
- e. Persediaan barang jadi (*finished goods inventory*), merupakan persediaan yang diperoleh dari hasil operasi atau produksi yang sudah selesai dan masih disimpan di gudang perusahaan.

Sedangkan menurut beberapa literatur, persediaan dapat diklasifikasikan berdasarkan fungsinya, yaitu (Rahmawati, 2017):

- a. Stok siklus (*cycle stock*), yakni jumlah persediaan yang tersedia setiap saat yang dipesan dalam ukuran lot.

- b. Stok tersumbat (*congestion stock*), persediaan dari produk yang diproduksi berkaitan dengan adanya batasan produksi, di mana banyak produk yang diproduksi pada peralatan produksi yang sama, khususnya jika biaya setiap produksinya relatif besar.
- c. Stok pengaman (*safety stock*), persediaan yang tersedia secara rata-rata untuk memenuhi permintaan dan penyaluran yang tak tentu dalam jangka pendek.
- d. Persediaan antisipasi (*anticipation inventory*), jumlah persediaan yang tersedia untuk mengatasi fluktuasi permintaan yang cukup tinggi.
- e. Persediaan *pipeline*, meliputi produk yang berada dalam perjalanan, yakni produk yang ada pada alat angkutan seperti truk antara setiap tingkat pada sistem distribusi eselon majemuk.
- f. Stok *decoupling*, digunakan dalam sistem eselon majemuk untuk mengizinkan setiap tingkat membuat keputusan masing-masing terhadap jumlah persediaan yang tersedia.

2.3.9.2 Sistem Inventaris

Sistem inventaris adalah sekumpulan kebijakan dan pengendalian, yang memonitor tingkat inventaris, dan menentukan tingkat mana yang harus dijaga, bila stok harus diisi kembali dan berapa banyak yang harus dipesan (Rahmawati, 2017).

Sistem inventaris akan memberikan kemungkinan struktur organisasi dan kebijakan operasi produksi, untuk menjaga dan mengawasi barang-barang untuk distok. Dengan sistem inventaris ini, diharapkan manajemen dapat bertanggung jawab terhadap pemesanan dan penerimaan barang yang dipesan. Hal ini dapat dilakukan dengan mengawasi waktu penempatan pesanan, dan menjaga atau mengawasi jalannya jalur dari apa yang dipesan, serta berapa banyak barang yang dipesan dan dari siapa vendor-nya (Rahmawati, 2017).

2.3.9.3 Produk

Produk adalah segala sesuatu yang dapat ditawarkan di pasar, untuk memuaskan kebutuhan dan keinginan konsumen. Produk terdiri atas barang, jasa, pengalaman, *events*, orang, tempat, kepemilikan, organisasi, informasi dan ide.

Jadi produk itu bukan hanya berbentuk suatu yang berwujud saja, seperti makanan, pakaian, dan sebagainya akan tetapi juga sesuatu yang tidak berwujud seperti pelayanan jasa. Semua diperlukan bagi pemuasan kebutuhan dan keinginan (*need and wants*) dari konsumen. Konsumen tidak hanya membeli produk sekedar memuaskan kebutuhan (*need*), akan tetapi juga bertujuan memuaskan keinginan (*wants*). Misalnya membeli bentuk sepatu, gaya warna, merek, dan harga yang menimbulkan/mengangkat *prestige* (Rahmawati, 2017).

2.3.9.4 Harga

Pengertian harga, nilai dan *utility*, merupakan konsep yang saling berhubungan. Yang dimaksud dengan utilitas ialah suatu atribut yang melekat pada suatu barang, yang memungkinkan barang tersebut, dapat memenuhi kebutuhan (*need*), keinginan (*wants*) dan memuaskan konsumen (*satisfaction*). Penetapan harga jual berasal dari harga pokok barang tersebut. Sedangkan harga pokok barang ditentukan oleh berapa besar biaya yang dikorbankan untuk memperoleh atau untuk membuat barang itu (Rahmawati, 2017).

2.3.9.5 Pemasok

Pemasok adalah suatu organisasi atau lembaga yang menyediakan atau memasok produk atau pelayanan kepada konsumen (Rahmawati, 2017).

2.3.9.6 Penjualan

Penjualan adalah suatu sistem kegiatan pokok perusahaan untuk memperjual-belikan barang dan jasa yang perusahaan hasilkan. Dalam penjualan terdapat dua macam yaitu sistem penjualan tunai dan sistem penjualan kredit (Rahmawati, 2017).

2.3.9.7 Gudang

Gudang adalah tempat penyimpanan sementara dan pengambilan inventaris untuk mendukung kegiatan operasi bagi proses operasi berikutnya, ke lokasi distribusi, atau kepada konsumen akhir. Gudang berfungsi untuk :

1. Menyimpan barang untuk sementara waktu sambil menunggu giliran untuk diproses.
2. Memantau pergerakan dan status barang.
3. Meminimumkan biaya pergerakan barang, peralatan, dan karyawan.

4. Menyediakan media komunikasi dengan konsumen mengenai barang.
5. Titik penyeimbang aliran inventaris dan barang.

Jika inventaris berlebihan, diletakkan di gudang. Setelah itu, bagian gudang harus memastikan bahwa inventaris/barang disimpan baik, terjaga kondisinya, dan tercatat statusnya sehingga tidak ada modal perusahaan yang hilang akibat kesalahan pengawasan di gudang (Rahmawati, 2017).

2.3.10 Blackbox Testing

Black-Box Testing (pengujian kotak hitam) yaitu menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Pengujian dimaksudkan untuk mengetahui apakah fungsi – fungsi, masukan, dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan.

Pengujian kotak hitam dilakukan dengan membuat kasus uji yang bersifat mencoba semua fungsi dengan memakai perangkat lunak apakah sesuai dengan spesifikasi yang dibutuhkan. Kasus uji yang dibuat untuk melakukan pengujian kotak hitam harus dibuat dengan benar dan kasus salah, misalkan untuk kasus proses login maka kasus uji yang dibuat adalah :

1. Jika *user* memasukkan nama pemakai (*username*) dan kata sandi (*password*) yang benar.
2. Jika *user* memasukkan nama pemakai (*username*) dan kata sandi (*password*) yang salah, misalnya nama pemakai benar tapi kata sandi salah, atau sebaliknya, atau keduanya salah (Winarno, dkk, 2014).

Pengujian *black-box* berfokus pada persyaratan fungsional perangkat lunak. Pengujian *black-box* memungkinkan perekayasa perangkat lunak mendapatkan serangkaian kondisi *input* yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program. Pengujian *black-box* bukan merupakan alternatif dari teknik *white-box*, tetapi merupakan pendekatan komplementer yang kemungkinan besar mampu mengungkap kelas kesalahan daripada metode *white-box*.

Pengujian *black-box* berusaha menemukan kesalahan dalam kategori sebagai berikut: (1) fungsi-fungsi yang tidak benar atau hilang, (2) kesalahan *interface*, (3) kesalahan dalam struktur data atau akses *database external*, (4) kesalahan kinerja, (5) inisialisasi dan kesalahan terminasi

Tidak seperti pengujian *white-box* yang dilakukan pada saat awal proses pengujian, pengujian *black-box* cenderung diaplikasikan selama tahap akhir pengujian. Karena pengujian *black-box* memperhatikan struktur kontrol, maka perhatian berfokus pada *domain* informasi. Pengujian di desain untuk menjawab pertanyaan-pertanyaan berikut :

1. Bagaimana validitas fungsional diuji ?
2. Kelas *input* apa yang akan membuat *test case* menjadi baik ?
3. Apakah sistem sangat sensitif terhadap harga input tertentu ?
4. Bagaimana batasan dari suatu data diisolasi ?
5. Kecepatan data apa dan volume data apa yang dapat ditolelir oleh sistem ?
6. Apa pengaruh kombinasi tertentu dari data terhadap operasi sistem ?

Mengaplikasikan teknik *black-box*, maka serangkaian *test case* akan memenuhi kriteria berikut ini : (1) *test case* yang mengurangi, dengan harga lebih dari satu, jumlah *test case* tambahan yang harus didesain untuk mencapai pengujian yang dapat dipertanggungjawabkan, dan (2) *test case* yang memberi tahu mengenai kehadiran atau ketidakhadiran kelas kesalahan, daripada memberitahu kesalahan yang berhubungan hanya dengan pengujian spesifik yang ada (Pressman, 2002).