

(10)

SISTEM INFORMASI AUDIT PERSEDIAAN BARANG BAJA**Putu Reksa W.P¹, Nonot Wisnu Karyanto²**

Program Studi Teknik Informatika, Fakultas Teknik, Universitas Wijaya Kusuma Surabaya

reksa_putu@yahoo.com**ABSTRAK**

Audit persediaan barang baja merupakan permasalahan yang selalu dihadapi para perusahaan barang stell seperti bahan baku dan laba rugi dari barang baja tersebut. Apabila audit persediaan tersebut tidak ditangani dengan baik, akan memberikan laporan persediaan barang baja tidak efisien dan tidak transparan. Pada dasarnya, pemegang perusahaan dan para staff perusahaan ingin mengetahui persediaan barang baja yang ada.

Sehubungan dengan hal berikut, dapat diberikan solusi dengan merancang sistem audit persediaan barang baja berdasarkan sistem audit persediaan barang baja yang sudah ada yang masih kurang efektif dan transparan. Dengan adanya aplikasi ini diharapkan memudahkan para auditor dan pimpinan perusahaan dalam melakukan perhitungan bahan baku awal dengan stok (sisa bahan produksi) agar bisa mengetahui laba rugi yang terjadi pada barang jadi.

Kata kunci : *Audit Persediaan, Persediaan Barang Baja*

PENDAHULUAN**Latar Belakang**

Masalah audit persediaan barang baja merupakan permasalahan yang selalu dihadapi para perusahaan barang stell seperti bahan baku dan laba rugi dari barang baja tersebut. Apabila audit persediaan tersebut tidak ditangani dengan baik, akan memberikan laporan persediaan barang baja tidak efisien dan tidak transparan. Pada dasarnya, pemegang perusahaan dan para staff perusahaan ingin mengetahui persediaan barang baja yang ada.

Dalam melakukan audit persediaan barang baja sering menghadapi berbagai masalah. Diantaranya tidak mengetahui jumlah dari barang baku awal yang akan terpakai, sehingga jumlah antara bahan baku awal dengan stok (sisa bahan produksi) tidak sesuai dengan bahan baku terpakai. Dan jumlah dari bahan baku awal dengan stok (sisa bahan produksi) akan memberikan pengaruh terhadap laba rugi dari barang jadi. Oleh karena itu pemilik perusahaan tidak mengetahui besar kecilnya asset yang dimiliki.

Sehubungan dengan hal berikut, dapat diberikan solusi dengan merancang sistem audit persediaan barang baja berdasarkan sistem audit persediaan barang baja yang sudah ada yang masih kurang efektif dan

transparan. Dengan adanya aplikasi ini diharapkan memudahkan para auditor dan pimpinan perusahaan dalam melakukan perhitungan bahan baku awal dengan stok (sisa bahan produksi) agar

bisa mengetahui laba rugi yang terjadi pada barang jadi.

TINJAUAN PUSTAKA**Teori dan standar Auditing**

Teori Auditing merupakan tuntunan untuk melaksanakan audit yang bersifat normatif. Konsep adalah abstraksi-abstraksi yang diturunkan dari pengalaman dan observasi, dan dirancang untuk memahami kesamaan-kesamaan di dalam suatu subyek, dan perbedaan-perbedaannya dengan subyek yang lain. Standar Auditing adalah pengukur atau teknik-teknik rinci untuk melaksanakan standar, sehingga prosedur akan berubah bila lingkungan auditnya berubah.

Menurut Mautz dan Sharaf teori auditing tersusun atas lima konsep dasar yaitu :

1. Bukti (*evidence*)

Tujuan memperoleh dan mengevaluasi bukti adalah untuk memperoleh pengertian sebagai dasar untuk memberikan kesimpulan atas pemeriksaan yang dituangkan dalam pendapat auditor.

- a. *Rasionalisasi*

Merupakan pemikiran asumsi yang diterima. Misalnya penghitungan kembali oleh auditor, dan pengamatan terhadap pengendalian intern.

- b. *Emperikisme*

Merupakan pengalaman yang sering terjadi. Misalnya perhitungan dan pengujian secara fisik.

c. *Pragmatisme*

Merupakan hasil praktik. Misalnya kejadian setelah tanggal selesainya pekerjaan lapangan.

2. Kehati-hatian dalam pemeriksaan (*Due Audit Care*)

Konsep kehati-hatian dalam pemeriksaan, didasarkan pada issue pokok tingkat kehati-hatian yang diharapkan pada auditor yang bertanggung jawab. Yang disebut Prudent Auditor. Konsep ini lebih dikenal dengan Konsep Konservatif

3. Penyajian atau pengungkapan yang wajar (*Fair Presentation*)

Konsep ini menuntut adanya informasi laporan keuangan yang bebas (tidak memihak), tidak bias, dan mencerminkan posisi keuangan, hasil operasi, dan aliran kas perusahaan.

Konsep ini dijabarkan lagi dalam 3 subkonsep, yaitu:

- Accounting Propriety* : berhubungan dengan penerapan prinsip akuntansi tertentu dalam kondisi tertentu.
- Adequate Disclosure* : berkaitan dengan jumlah dan luas pengungkapan atau penyajian informasi

4. Independensi (*Independence*)

Merupakan suatu sikap mental yang dimiliki auditor untuk tidak memihak dalam melakukan audit. Konsep independensi berkaitan dengan independensi pada diri pribadi auditor secara individual, dan independensi secara bersama-sama.

5. Etika Perilaku (*Ethical Conduct*)

Etika dalam auditing berkaitan dengan perilaku yang ideal seorang auditor profesional yang independen dalam melaksanakan audit.

Pengertian Audit Sistem Informasi

Standar Auditing merupakan suatu kaidah agar mutu auditing dapat dicapai sebagaimana mestinya yang harus diterapkan dalam setiap audit atas laporan keuangan yang dilakukan auditor independen.

Standar auditing terdiri atas tiga bagian yaitu :

A. Standar Umum

- Audit harus dilaksanakan oleh seorang atau lebih yang memiliki keahlian dan pelatihan teknis yang cukup sebagai auditor.
- Dalam semua hal yang berhubungan dengan perikatan, independensi dalam sikap mental harus dipertahankan oleh auditor
- Dalam pelaksanaan audit dan penyusunan laporannya, auditor wajib menggunakan kemahiran profesionalnya dengan cermat dan seksama

B. Standar Pekerjaan

- Pekerjaan harus direncanakan sebaik-baiknya dan jika digunakan asisten harus disupervisi dengan semestinya.
- Pemahaman memadai atas pengendalian intern harus diperoleh untuk merencanakan audit dan menentukan sifat, saat, dan lingkup pengujian yang akan dilakukan
- Bukti audit kompeten yang cukup harus diperoleh melalui inspeksi, pengamatan, permintaan keterangan, dan konfirmasi sebagai dasar memadai untuk menyatakan pendapat atas laporan keuangan yang diaudit.

C. Standar Pelaporan

- Laporan auditor harus menyatakan apakah laporan keuangan telah disusun sesuai dengan prinsip akuntansi yang berlaku umum di Indonesia
- Laporan auditor harus menunjukkan atau menyatakan, jika ada, ketidakkonsistenan penerapan prinsip akuntansi dalam penyusunan laporan keuangan periode berjalan dibandingkan dengan penerapan prinsip akuntansi tersebut dalam periode sebelumnya
- Pengungkapan informatif dalam laporan keuangan harus dipandang memadai, kecuali dinyatakan lain dalam laporan auditor
- Laporan auditor harus memuat suatu pernyataan pendapat mengenai laporan keuangan secara keseluruhan atau suatu asersi bahwa pernyataan demikian tidak dapat diberikan. Jika pendapat secara keseluruhan tidak dapat diberikan, maka alasannya harus dinyatakan. Dalam hal nama auditor dikaitkan dengan laporan keuangan, maka laporan auditor harus

memuat petunjuk yang jelas mengenai sifat pekerjaan audit yang dilaksanakan, jika ada, dan tingkat tanggung jawab yang dipikul oleh auditor.

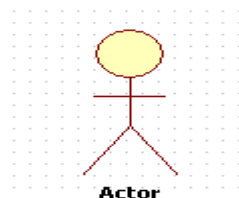
Pengertian UML

UML (*Unified Modeling Language*) adalah sebuah bahasa untuk menentukan, visualisasi, konstruksi, dan mendokumentasikan artifact (bagian dari informasi yang digunakan atau dihasilkan dalam suatu proses pembuatan perangkat lunak. Artifact dapat berupa model, deskripsi atau perangkat lunak) dari system perangkat lunak, seperti pada pemodelan bisnis dan *system non perangkat lunak* lainnya (Sholih,2005).

UML merupakan suatu kumpulan teknik terbaik yang telah terbukti sukses dalam memodelkan sistem yang besar dan kompleks. UML tidak hanya digunakan dalam proses pemodelan perangkat lunak, namun hampir dalam semua bidang yang membutuhkan pemodelan. Bagian-bagian diagram utama dari UML adalah :

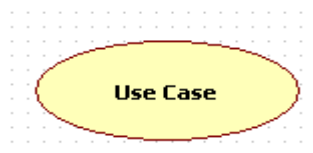
1. Use Case Diagram

Use *case diagram* menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use case* merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, *create* sebuah daftar belanja, dan sebagainya.



Gambar 2.1 aktor pada usecase

Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu.

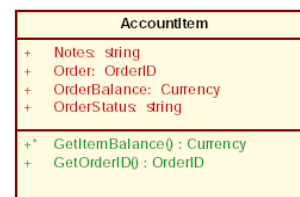


Gambar 2.2 usecase

2. Class Diagram

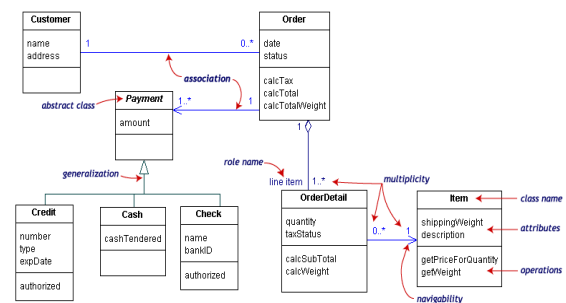
Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/property) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi).

Class diagram menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain.



Gambar 2.4 Diagram class

Contoh diagram class



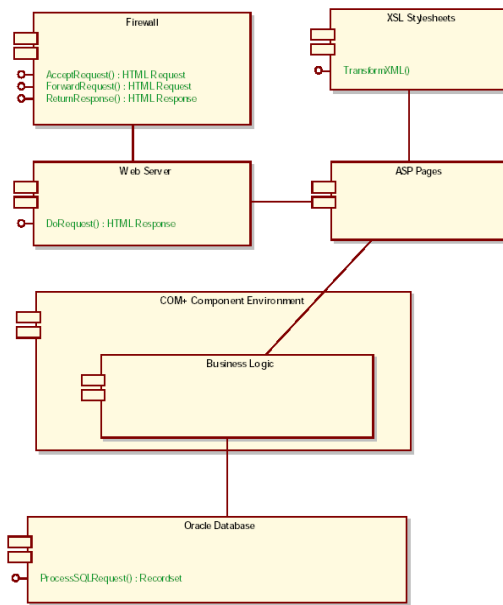
Gambar 2.5 contoh diagram class

3. Component Diagram

Component *diagram* menggambarkan struktur dan hubungan antar komponen piranti lunak, termasuk ketergantungan (*dependency*) di antaranya.

Komponen piranti lunak adalah modul berisi *code*, baik berisi *source code* maupun *binary code*, baik *library* maupun *executable*, baik yang muncul pada *compile time*, *link time*, maupun *run time*. Umumnya komponen terbentuk dari beberapa *class* dan/atau *package*, tapi dapat juga dari komponen-komponen yang lebih kecil. Komponen dapat juga berupa *interface*, yaitu kumpulan layanan yang disediakan sebuah komponen untuk komponen lain.

Contoh diagram component

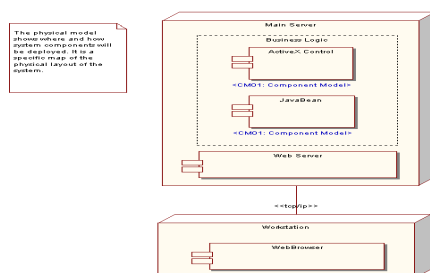


Gambar 2.6 contoh diagram komponen

4. Deployment Diagram

Deployment/physical diagram menggambarkan detail bagaimana komponen di-deploy dalam infrastruktur sistem, di mana komponen akan terletak (pada mesin, server atau piranti keras apa), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi server, dan hal-hal lain yang bersifat fisik.

Sebuah node adalah server, *workstation*, atau piranti keras lain yang digunakan untuk men-deploy komponen dalam lingkungan sebenarnya. Hubungan antar *node* (misalnya TCP/IP) dan *requirement* dapat juga didefinisikan dalam diagram ini.



Gambar 2.7 contoh diagram deployment

5. Sequence Diagram

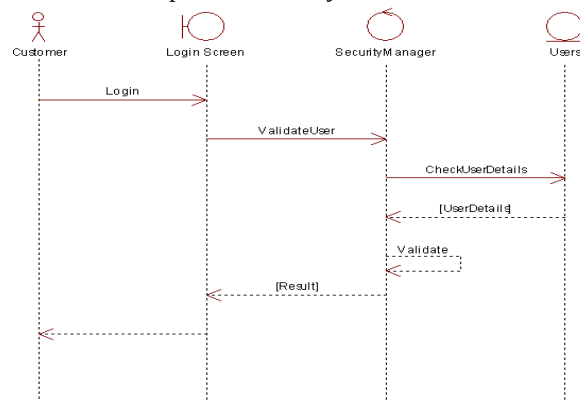
Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri atas dimensi vertikal

(waktu) dan dimensi horizontal (objek-objek yang terkait).

Sequence diagram biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah *event* untuk menghasilkan *output* tertentu. Diawali dari apa yang men-trigger aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan *output* apa yang dihasilkan.

Masing-masing objek, termasuk aktor, memiliki *lifeline* vertikal. *Message* digambarkan sebagai garis berpanah dari satu objek ke objek lainnya. Pada fase desain berikutnya, *message* akan dipetakan menjadi operasi/metoda dari *class*.

Activation bar menunjukkan lamanya eksekusi sebuah proses, biasanya diawali dengan diterimanya sebuah *message*. Untuk objek-objek yang memiliki sifat khusus, standar UML mendefinisikan *icon* khusus untuk objek *boundary*, *controller* dan *persistent entity*.



Gambar 2.8 contoh diagram sequential

6. Statechart diagram

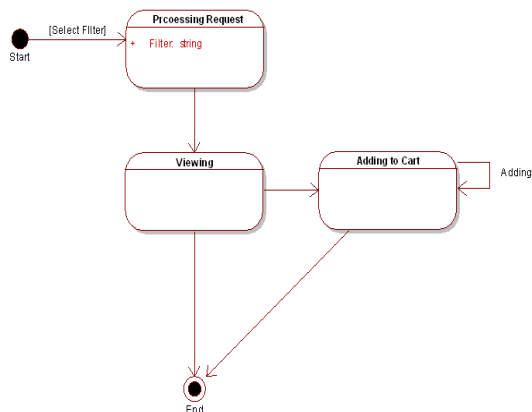
Statechart diagram menggambarkan transisi dan perubahan keadaan (dari satu *state* ke *state* lainnya) suatu objek pada sistem sebagai akibat dari *stimuli* yang diterima. Pada umumnya *statechart diagram* menggambarkan *class* tertentu (satu *class* dapat memiliki lebih dari satu *statechart diagram*).

Dalam UML, *state* digambarkan berbentuk segiempat dengan sudut membulat dan memiliki nama sesuai kondisinya saat itu. Transisi antar *state* umumnya memiliki kondisi *guard* yang merupakan syarat terjadinya transisi yang bersangkutan, dituliskan dalam kurung siku. *Action* yang dilakukan sebagai akibat dari *event* tertentu dituliskan dengan diawali garis miring. Titik awal dan akhir digambarkan berbentuk lingkaran berwarna penuh dan berwarna setengah.

Untuk menentukan apakah sebuah *class* mempunyai perilaku dinamik yang signifikan, dilakukan dengan memperlihatkan dua hal yaitu :

- *Attribut* : jika sebuah kelas mempunyai sebuah attribute yang bernama status atau *attribute* yang mempunyai nilai berbeda yang digunakan menyimpan berbagai macam kondisi yang mungkin, maka ini adalah cara yang baik mempertimbangkan bahwa objek – objek dalam sebuah kelas mempunyai perilaku dinamik.
- *Relasi* : cara berikutnya adalah memeriksa relasi antar kelas. Perhatikan *multiplicity* dalam sebuah relasi nol. Nol menunjukkan bahwa ada objek tertentu yang tidak terlibat dalam relasi.

Contoh *statechart diagram* :



Gambar 2.9 contoh diagram statechart

Definisi Web

Secara terminologi, *website* adalah kumpulan dari halaman - halaman situs, yang terangkum dalam sebuah *domain* atau *subdomain*, yang tempatnya berada di dalam *World Wide Web* (WWW) di dalam Internet. Sebuah halaman web yang berupa dokumen yang ditulis dalam format HTML (*Hyper Text Markup Language*), yang diakses melalui HTTP, yaitu sebuah protokol yang menyampaikan informasi dari *server website* untuk ditampilkan kepada para pengguna melalui *web browser* (Andri Kristianto, 2008).

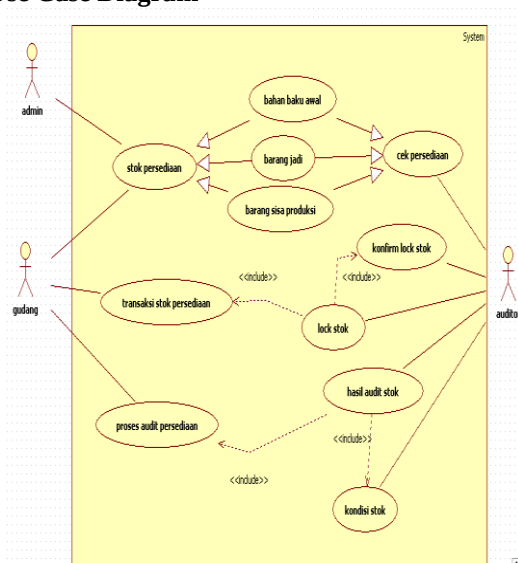
ANALISA DAN PERANCANGAN

Analisa Sistem

Analisa sistem merupakan kegiatan penguraian suatu sistem informasi yang utuh dan nyata ke dalam bagian-bagian atau komponen-komponen komputer yang bertujuan untuk

mengidentifikasi serta mengevaluasi masalah-masalah yang muncul, hambatan-hambatan yang mungkin terjadi dan kebutuhan-kebutuhan yang diharapkan sehingga mengarah kepada suatu solusi untuk perbaikan maupun pengembangan ke arah yang lebih baik dan sesuai dengan kebutuhan serta perkembangan teknologi. Analisa sistem dalam pembuatan aplikasi manajemen ruang rapat menggunakan *usecase*, *Sequential diagram*.

Use Case Diagram



Gambar 3.1 Use Case SI Audit Persediaan Barang Baja

Pada gambar 3.1 di dalam perancangan ini, sebagai *actor* adalah gudang, produksi, dan auditor. Diagram use case ini menunjukkan interaksi antara use case dan *actor* untuk sistem informasi audit persediaan barang baja, antara lain :

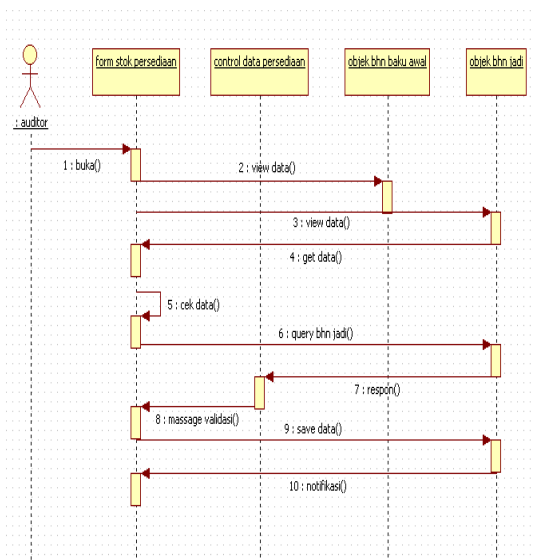
1. *Actor* gudang menggunakan beberapa *use case*, antara lain stok persediaan, transaksi stok persediaan, proses audit persediaan. Terdapat relasi *generalization* antara *usecase* stok persediaan dengan *usecase* bahan baku awal, bahan jadi, barang sisa produksi. Dimana *usecase_bahan baku awal*, *usecase_bahan jadi*, *usecase_barang sisa produksi* menjadi dasar atau bagian dari *usecase* stok persediaan.
2. *Actor* admin menggunakan beberapa *use case*, antara lain stok persediaan, transaksi stok persediaan, proses audit persediaan. Terdapat relasi *generalization* antara *usecase* stok persediaan dengan *usecase* bahan baku awal, bahan jadi, barang sisa produksi. Dimana *usecase_bahan baku awal*, *usecase_bahan jadi*,

usecase barang sisa produksi menjadi dasar atau bagian dari *usecase* stok persediaan.

3. Actor auditor menggunakan beberapa use case, antara lain cek persediaan, cek produksi, lock stok, hasil audit stok, kondisi stok. Terdapat relasi *generalization* antara *usecase* stok persediaan dengan *usecase* bahan baku awal, bahan jadi, barang sisa produksi. Dimana *usecase* bahan baku awal, *usecase* bahan jadi, *usecase* barang sisa produksi menjadi dasar atau bagian dari *usecase* stok persediaan. Kemudian Terdapat relasi *generalization* antara *usecase* stok produksi dengan *usecase* barang dalam proses dan bahan baku terpakai. Dimana *usecase* barang dalam proses dan *usecase* bahan baku terpakai menjadi dasar atau bagian dari *usecase* stok produksi. Relasi *include* antara *usecase* hasil audit stok dengan *usecase* kondisi stok, berarti apabila actor auditor menjalankan *usecase* hasil audit stok maka *usecase* kondisi stok juga harus dijalankan.

Diagram Sequential

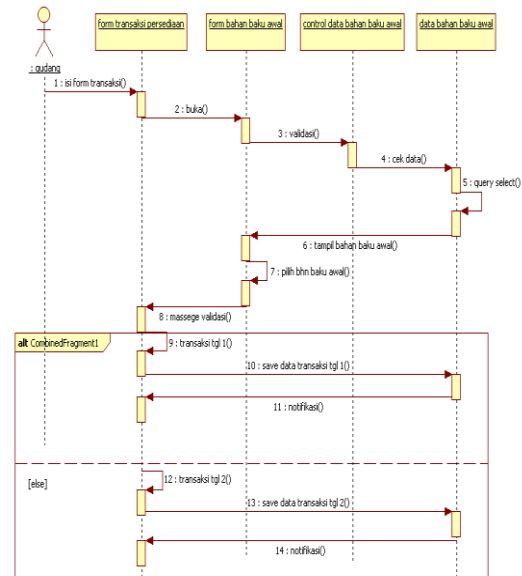
Diagram sekuensial atau *sequence diagram* digunakan untuk menunjukkan aliran fungsionalitas dalam *usecase*. Diagram ini memperlihatkan interaksi yang menekankan pada pengiriman pesan (*message*) dalam suatu waktu tertentu.



Gambar 3.2 Diagram sekuensial cek persediaan

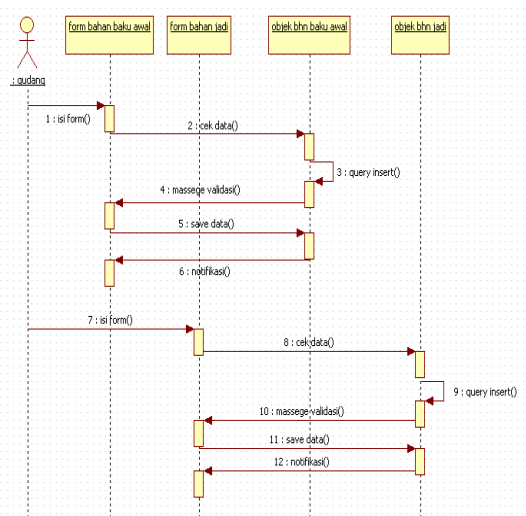
Pada gambar 3.2 dijelaskan auditor melakukan buka pada objek *form* stok persediaan. *Form* stok persediaan ini menjadi *interface* bagi actor auditor, selanjutnya *form* stok persediaan melakukan view data di *object* bahan jadi, kemudian

form stok persediaan menampilkan data dan cek data persediaan. Apabila semua proses telah selesai maka *form* stok persediaan akan melakukan *query* ke *object* bahan jadi. Kemudian *form* stok persediaan melakukan *query* simpan ke *object* bahan jadi untuk di simpan ke database.



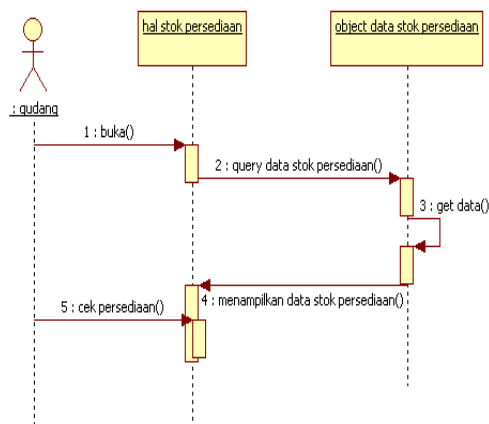
Gambar 3.3 Diagram sekuensial transaksi stok persediaan

Pada gambar 3.3 dijelaskan gudang mengisi *form* transaksi persediaan, kemudian buka *form* bahan baku awal setelah itu pada *control* data bahan baku awal melakukan validasi data, dan data bahan baku awal melakukan *query select*. Kemudian data bahan baku awal ditampilkan pada *form* bahan baku awal dan memilih data, dan pada *form* transaksi persediaan melakukan transaksi bahan baku awal. Apabila semua proses telah selesai maka *form* transaksi persediaan melakukan *query* ke data bahan baku awal untuk disimpan ke database



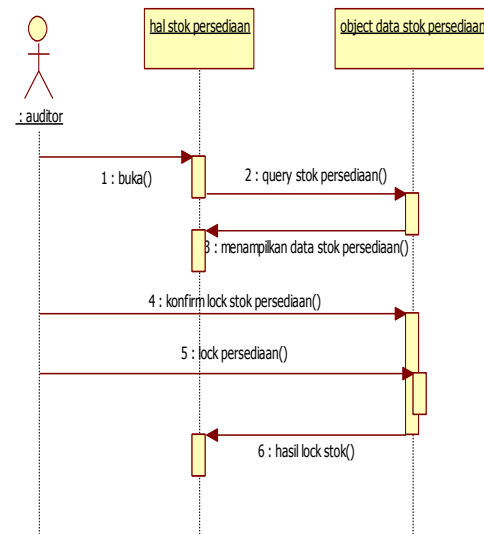
Gambar 3.4 Diagram sekuensial stok persediaan

Pada gambar 3.4 dijelaskan gudang isi *form* bahan baku awal dan *object* data bahan baku awal melakukan cek data dan *query insert* kemudian *object* data bahan baku awal memberikan pesan validasi ke *form* bahan baku awal. Apabila semua proses telah selesai maka *form* bahan baku awal melakukan *query* ke *object* bahan baku awal untuk disimpan ke database.



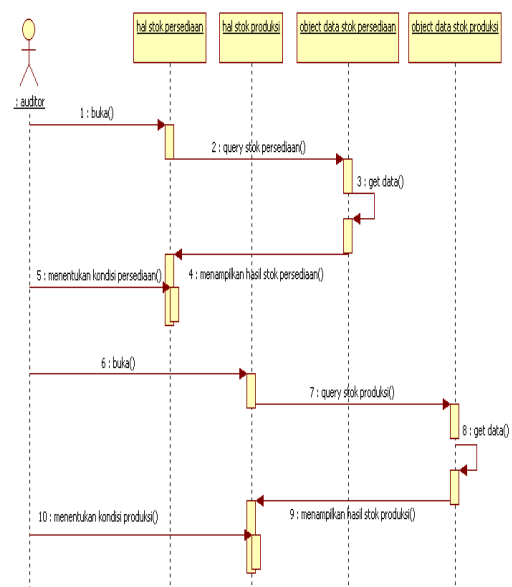
Gambar 3.5 Diagram sekuensial stok persediaan audit

Pada gambar 3.5 dijelaskan gudang membuka halaman stok persediaan. Kemudian *query* data dilakukan dan diterima oleh *object* data stok persediaan. Setelah *query* diterima maka data stok persediaan akan ditampilkan pada halaman stok persediaan.



Gambar 3.6 Diagram sekuensial lock stok

Pada gambar 3.6 dijelaskan auditor membuka halaman stok persediaan. Kemudian halaman stok persediaan melakukan *query* stok persediaan dan diterima oleh *object* data stok persediaan, setelah menerima *query* maka *object* data stok persediaan menampilkan data stok persediaan di halaman stok persediaan. Setelah semua proses selesai maka auditor akan melakukan konfirmasi lock stok persediaan, dimana konfirmasi lock stok persediaan ini memberikan peringatan bahwa stok persediaan tersebut akan di lock oleh auditor.



Gambar 3.7 Diagram sekuensial hasil stok

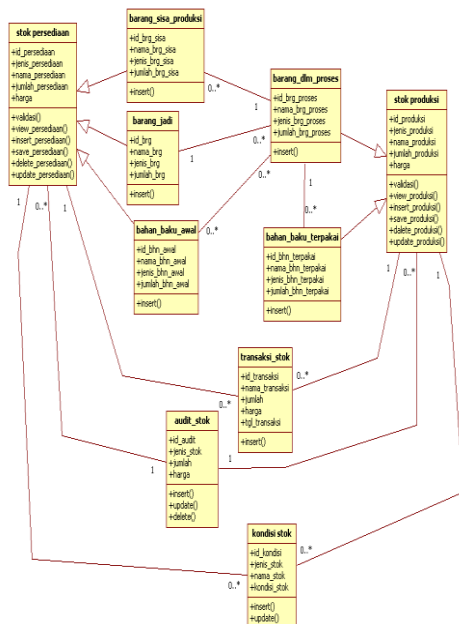
Pada gambar 3.7 dijelaskan auditor membuka halaman stok persediaan dan halaman stok produksi. Kemudian halaman stok persediaan melakukan *query* stok persediaan dan diterima oleh *object* data stok persediaan, setelah menerima *query* maka *object* data stok persediaan menampilkan data stok persediaan di halaman stok persediaan. Setelah semua proses selesai maka auditor akan menentukan kondisi stok, dimana kondisi stok tersebut diketahui jika persediaan telah dilakukan *insert* data pada waktu adanya transaksi.

Perancangan Sistem

Perancangan sistem dibangun untuk memenuhi kebutuhan user mengenai gambaran yang jelas tentang perancangan sistem yang akan dibangun serta diimplementasikan. Perancangan sistem yang digunakan dalam membangun aplikasi management ruang rapat adalah dengan menggunakan Class Diagram, State machine Diagram, Componen Diagram, deployment Diagram dan database diagram.

Class Diagram

Class diagram adalah diagram yang menunjukkan interaksi antar class / relasi antar class dalam sebuah sistem. Disamping itu diagram class juga memberikan pandangan global atas sebuah sistem. Hal tersebut tercermin dari class – class yang ada dan relasinya satu dengan yang lain.

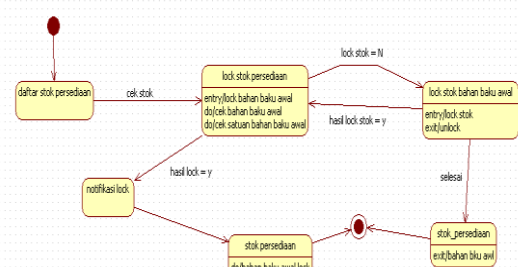


Gambar 3.8 Diagram class audit persediaan barang baja

Pada gambar 3.8 barang sisa produksi, barang jadi, dan bahan baku awal *generalization* dengan stok persediaan karena barang sisa produksi, barang jadi, dan bahan baku awal mewarisi atribut dan operasi yang didefinisikan secara *public* oleh stok persediaan. Kemudian barang dalam proses dan bahan baku terpakai merupakan *generalization* dengan stok produksi karena barang dalam proses dan bahan baku terpakai mewarisi atribut dan operasi yang didefinisikan secara *public* oleh stok produksi. Barang sisa produksi memiliki satu dalam proses, barang jadi bisa diolah banyak barang dalam proses. Dari barang dalam proses bisa menggunakan banyak bahan baku terpakai. Stok persediaan bisa melakukan banyak transaksi stok. Kemudian stok produksi juga bisa melakukan banyak transaksi stok. audit stok bisa mengaudit banyak stok persediaan, kemudian audit stok juga bisa mengaudit banyak stok produksi.

Statechart Diagram

Diagram statechart menunjukkan siklus hidup sebuah objek tunggal, dari saat dibuat sampai objek tersebut dihapus. Diagram statechart merupakan cara tepat untuk memodelkan perilaku dinamis sebuah *class*. Pada diagram *class* yang telah dibuat diatas terdapat satu kelas yang memiliki perilaku dinamis yaitu *class lock* stok. Perilaku dinamik *class* tersebut di tunjukkan dengan atribut yang mempunyai nilai berbeda yang digunakan untuk menyimpan berbagai macam kondisi. Pada *class lock* stok terdapat atribut stok yang dapat menyatakan beberapa kondisi diantaranya tidak *lock* stok dan hasil *lock* stok.



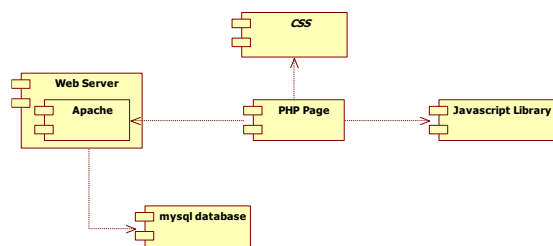
Gambar 3.9 Diagram statechart untuk lock stok

Pada gambar 3.9 diatas dapat dijelaskan suatu stok dapat dimulai dari daftar stok persediaan. Audit stok bergerak ke *lock* stok persediaan. Pada kondisi ini apabila *lock* stok sama dengan *lock* bernilai N, maka kondisi stok persediaan itu belum diaudit. Tetapi apabila hasil *lock* stok bernilai Y maka kondisi stok persediaan tersebut sedang diaudit.

Komponent Diagram

Terdapat beberapa komponen penyusun system informasi manajemen rapat. komponen tersebut diantaranya class css yang menjadi komponen untuk memperindah interface. Class php page sebagai penyusun halaman aplikasi. Class javascript library sebagai compile javascript. Class webserver sebagai komponen yang menampilkan php page di browser. Class mysql database sebagai komponen yang bertugas untuk menyiapkan hak akses kepada banyak pengguna terhadap data – data yang diperlukan oleh pengguna dan melayani proses penyimpanan data. Class page php memiliki relasi dependensi dengan class css, class webserver dan class library javascript. Hal ini berarti class php page tidak bisa dikompilasi sebelum class web server, class css dan class javascript library selesai dikompilasi.

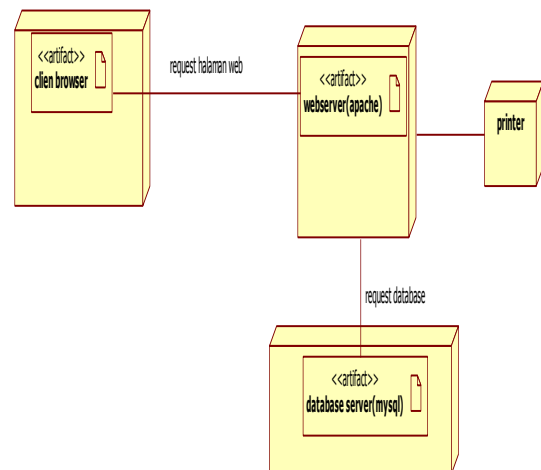
Class webserver memiliki relasi dependensi dengan class mysql database. Hal ini berarti class webserver tidak bisa dikompilasi sebelum class mysql database dikompilasi. Untuk selengkapnya dijelaskan pada gambar 3.15



Gambar 3.10 Diagram komponen audit persediaan barang baja

Deployment Diagram

Diagram deployment menggambarkan bagaimana konfigurasi perangkat keras dimana sistem akan diinstalasikan. Diagram ini memberi simbol untuk segala bentuk perangkat keras, proses – proses dan hubungan perangkat dengan proses seperti: komputer, printer, router, switch, jaringan, dan sebagainya.

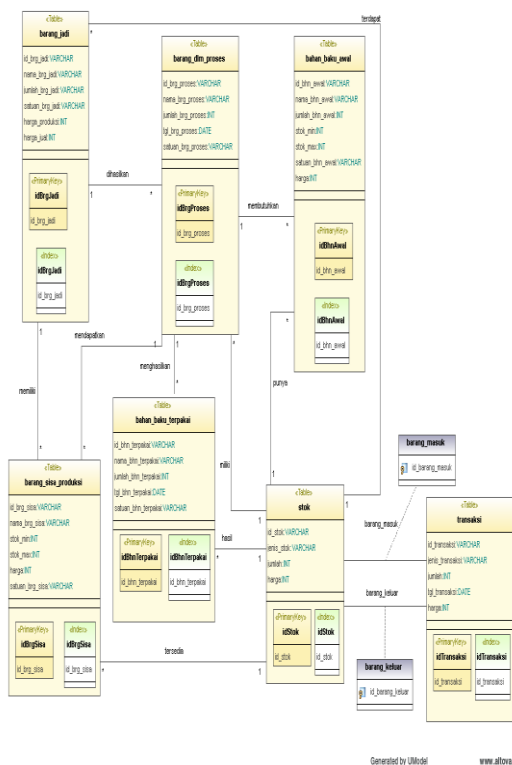


Gambar 3.11 Diagram Deployment audit persediaan barang baja

Client browser digunakan untuk menampilkan halaman web. Webserver menangani permintaan / request dari browser dan secara dinamis men-generate halaman web dan kode program untuk dijalankan dan ditampilkan di client. Database server bertugas untuk menyiapkan hak akses kepada banyak pengguna terhadap data – data yang diperlukan oleh pengguna dan melayani proses penyimpanan data. Sedangkan printer adalah peralatan yang digunakan untuk mencetak data sesuai dengan permintaan pengguna.

Database Diagram

Database diagram merupakan suatu model untuk menjelaskan hubungan antar data dalam **basis data** berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi. Gambar berikut menunjukkan bagaimana UML memodelkan sebuah aplikasi berkaitan dengan database model.



Gambar 3.12 Diagram database audit persediaan barang baja

Pada gambar 3.12 diatas dapat dijelaskan hubungan relasinya seperti berikut :

1. Tabel barang jadi berasosiasi dengan tabel barang dalam proses dengan nama relasi asosiasi dihasilkan yang berarti barang jadi bisa dihasilkan banyak barang dalam proses.
2. Tabel barang jadi berasosiasi dengan tabel barang sisa produksi dengan nama relasi asosiasi memiliki yang berarti barang jadi bisa memiliki banyak barang sisa produksi.
3. Tabel barang jadi berasosiasi dengan tabel stok dengan nama relasi asosiasi terdapat yang berarti banyak barang jadi terdapat dalam satu stok.
4. Tabel barang dalam proses berasosiasi dengan tabel bahan baku awal dengan nama relasi asosiasi membutuhkan yang berarti barang dalam proses bisa membutuhkan banyak bahan baku awal.
5. Tabel barang dalam proses berasosiasi dengan tabel bahan baku terpakai dengan nama relasi menghasilkan yang berarti barang dalam proses bisa menghasilkan banyak bahan baku terpakai.
6. Tabel barang dalam proses berasosiasi dengan tabel barang sisa produksi dengan nama relasi mendapatkan yang berarti barang dalam proses

bisa mendapatkan banyak barang sisa produksi.

7. Tabel barang dalam proses berasosiasi dengan tabel stok dengan nama relasi asosiasi miliki yang berarti banyak barang dalam proses miliki dalam satu stok.
8. Tabel barang sisa produksi berasosiasi dengan tabel stok dengan nama relasi asosiasi tersedia yang berarti banyak barang sisa produksi tersedia dalam satu stok.
9. Tabel bahan baku terpakai produksi berasosiasi dengan tabel stok dengan nama relasi asosiasi hasil yang berarti banyaknya hasil bahan baku terpakai dalam satu stok.

HASIL DAN PEMBAHASAN

Dalam menggunakan sistem informasi manajemen rapat user membutuhkan hardware dan software untuk mendukung dalam kinerja dan proses sistem yang akan dibuat agar kinerja sistem bisa maksimal.

Kebutuhan Perangkat Keras (*Hardware*)

Perangkat keras komputer atau hardware sangat mendukung dalam kinerja sistem. Semakin baik spesifikasi perangkat keras yang ada maka kinerja sistem pun akan semakin baik. Spesifikasi perangkat keras yang digunakan adalah sebagai berikut :

1. Processor : Intel(R) Pentium(R) Core IM i3-350M processor
2. RAM : 2 GB (sebaiknya lebih)
3. VGA : On-Board
4. Keyboard dan mouse

Dengan spesifikasi perangkat keras komputer yang digunakan dapat membantu dalam pengolahan data.

Kebutuhan Perangkat Lunak (*Software*)

Aplikasi sistem informasi manajemen rapat ini memerlukan perangkat lunak (*Software*) agar mampu beroperasi sesuai dengan yang diharapkan. Adapun perangkat lunak (*Software*) yang digunakan, yaitu :

1. Sistem Operasi : windows, linux, mac os .
2. Web browser seperti Mozilla Firefox, opera, google chrom .

Uji Coba System

Dalam melakukan pengujian sistem, yang dilakukan sesuai dengan ketentuan pengajuan rapat

seperti dijelaskan pada sub bab 2.1.3 telah menghasilkan beberapa skenario meliputi:

Menampilkan informasi jumlah stok

Untuk mengetahui informasi jumlah stok jika terjadi transaksi stok pada persediaan, gudang bisa masuk kehalaman from data stok yang akan dilihat stok. Contoh : gudang akan melihat jumlah stok persediaan bahan baku awal dengan mencari nama bahan baku awal **Tembaga** dan jumlah stok 466, seperti pada gambar 5.1 berikut.

Gambar 5.1 form cari menurut nama bahan

Setelah gudang melakukan pencarian menurut nama bahan yang dipilih maka akan keluar data bahan menurut hasil pencarian tersebut, seperti gambar 5.2 berikut.

Gambar 5.2 Form hasil pencarian menurut nama bahan

Pada saat gudang melihat stok bahan baku awal tersebut, jumlah stok belum terjadi transaksi stok. Ketika terjadi transaksi pada stok dengan nama bahan baku awal **Tembaga** maka jumlah stok akan secara langsung ter-update. Contoh : pada stok bahan baku awal terjadi transaksi stok untuk barang sisa produksi sebanyak 6 bahan, seperti gambar 5.3 berikut.

Gambar 5.3 Form transaksi bahan baku awal

Maka jumlah bahan baku awal yang jumlah awalnya sebanyak 466 akan langsung berkurang menjadi 460 bahan. Seperti pada gambar 5.4 di bawah ini.

Gambar 5.4 Form stok bahan baku awal

Untuk memperjelas proses simulasi melihat stok persediaan bahan baku awal perhatikan data bahan baku awal pada gambar 5.5. Data tersebut diambil dari database bahan baku awal

id	nama_bhn_awl	jumlah_bhn_awl	stok_min	stok_max	satuan	harga_awl	tgl_masuk_awl	kunci
1	seng	190	12	2000	TON	3000	2012-10-18	N
2	biiji besi	400	40	50000	KG	4000	2012-10-18	N
3	Batu Bara	340	3	4000	TON	5000	2012-10-20	N
4	Baja Strategang	234	2	100	TON	5000	2012-10-25	N
5	Baja Struktur	544	30	500	KG	700	2012-10-22	N
6	Aluminium Frame 3"	567	30	900	TON	3000	2012-10-30	N
7	Aluminium Pol s/s	452	10	100	KG	3000	2012-11-09	N
8	Besi Stainless steel	768	21	100	TON	89300	2012-11-20	N
9	Besi Tulangan Beton	684	30	100	TON	20700	2012-11-22	N
10	Besi Hollow Galvanis 4/4	245	10	900	TON	44600	2012-11-23	N
11	Homogeneous tile Bx. Gra...	856	10	100	KG	691900	2012-11-22	N
12	Aluminium	311	12	1000	TON	45600	2012-11-20	N
13	Tembaga	466	10	666	KG	8700	2012-11-20	N

Gambar 5.5 Daftar bahan baku awal

Auditor melakukan audit sewaktu - waktu

Pada proses ini, auditor akan melakukan audit stok persediaan bahan baku awal pada sewaktu – waktu. Jika proses ini dijalankan oleh auditor maka mengirimkan status lock stok persediaan bahan baku awal kepada bagian gudang. Di mana bagian gudang tidak bisa melakukan kegiatan apapun didalam stok persediaan bahan baku awal tersebut. Perhatikan gambar 5.6 berikut ini.

Gambar 5.6 Form data bahan baku awal

Pada gambar 5.6 dijelaskan, form data bahan baku awal tersebut terdapat pada halaman auditor. Form data bahan baku awal tersebut akan diaudit oleh auditor, data bahan baku awal yang akan diaudit adalah bahan baku awal **tembaga**. Stok persediaan bahan baku awal yang akan diaudit, terlebih dahulu auditor harus melihat detail dan *men-download* dari data bahan baku awal yang akan diaudit tersedut. Perhatikan gambar 5.7 berikut ini.

Gambar 5.7 Form detail data bahan baku awal

Pada gambar 5.7 dijelaskan, setelah auditor melihat detail bahan baku awal menurut data nama bahan baku awal maka auditor selanjutnya harus *men-download* data bahan baku awal yang akan diaudit, perhatikan gambar 5.8 berikut ini.

Gambar 5.8 Form *download* data bahan baku awal

Setelah data stok bahan baku awal tersebut *didownload* maka auditor melakukan audit terhadap bahan baku awal yang sudah *didownload*. Jika auditor sudah selesai melakukan audit terhadap stok persediaan bahan baku awal tersebut, maka auditor melakukan input hasil dari audit yang sudah dilakukan. Perhatikan gambar 5.9 berikut ini.

Gambar 5.9 Form hasil audit

Pada gambar 5.9 dijelaskan, setelah auditor melakukan audit bahan baku awal **tembaga** maka auditor meng-*inputkan* data hasil bahan baku awal **tembaga**. Sehingga pada form data bahan baku awal **tembaga** dibagian kolom pengaturan data bahan baku awal tampil icon fungsinya untuk memberikan informasi bahwa data bahan baku awal **tembaga** sudah diaudit, perhatikan gambar 5.10 berikut ini.

Gambar 5.10 Form data bahan baku awal setelah diaudit

Pada gambar 5.10 dijelaskan, jika pada stok bahan baku awal sudah tampil icon fungsinya untuk memberikan informasi bahwa data bahan baku awal **tembaga** sudah diaudit maka bagian gudang juga akan menerima hal yang serupa, dan data bahan baku awal tidak bisa dilakukan edit dan hapus data stok bahan baku awal, perhatikan gambar 5.11

Gambar 5.11 Form data bahan baku awal gudang

Auditor melakukan *Lock* stok persediaan

Pada proses ini auditor melakukan *lock* stok persediaan terhadap stok persediaan bahan baku awal. Proses *lock* stok persediaan ini digunakan oleh auditor untuk melakukan *lock* stok persediaan yang akan diaudit oleh auditor, agar bagian gudang tidak bias melakukan kegiatan seperti edit dan hapus pada stok persediaan yang akan diaudit. Perhatikan gambar 5.12 berikut.

Gambar 5.12 Form *lock* stok persediaan bahan baku awal

Pada gambar 5.12 dijelaskan, setelah auditor melakukan pencarian data bahan baku awal berdasarkan jenis stok dan satuan, maka akan tampil data yang dicari. Seperti gambar 5.13 berikut.

Gambar 5.13 Form *lock* bahan baku awal

Pada gambar 5.13 dijelaskan, setelah auditor melakukan pencarian tersebut maka auditor sudah bisa melakukan *lock* stok terhadap bahan baku awal yang akan diaudit. Contoh : auditor akan melakukan *lock* stok terhadap bahan baku awal *Seng*. Perhatikan gambar 5.14 berikut.

Gambar 5.14 Form *lock* bahan baku awal *seng*.

Pada gambar 5.14 dijelaskan, pada data bahan baku awal *seng* tersebut sudah melalui proses *lock* stok. Maka pada bagian gudang akan menerima notifikasi *lock* stok persediaan. Perhatikan gambar 5.15 berikut.

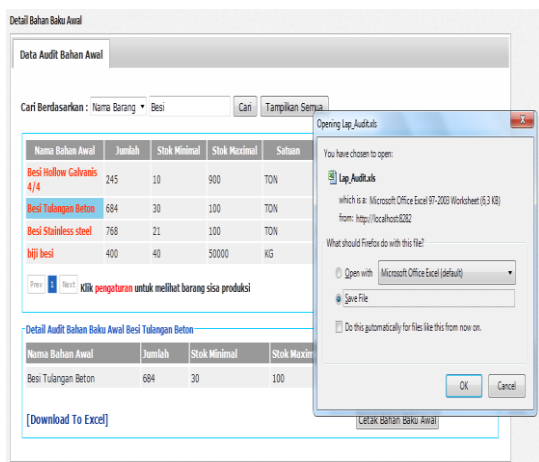
Gambar 5.15 Form notifikasi *lock* stok persediaan

Pada gambar 5.15 dijelaskan, notifikasi *lock* stok persediaan tersebut hanya memberikan informasi bahwa data stok persediaan bahan baku awal *seng* telah sudah *di-lock* oleh auditor, maka bagian gudang tidak bisa melakukan proses edit dan hapus pada stok persediaan bahan baku awal tersebut dan pada bagian pengaturan tampil icon . Perhatikan gambar 5.16 berikut.

Gambar 5.16 Form data bahan baku awal setelah *di-lock*

Auditor melakukan *input* hasil audit.

Pada proses ini auditor melakukan *input* hasil audit terhadap stok persediaan yang sebelumnya sudah melalui proses audit oleh auditor. Maka auditor melakukan *input* hasil audit harus sesuai dengan data stok persediaan yang sudah di audit. Contoh : auditor sudah melakukan audit terhadap stok persediaan bahan baku awal Besi Tulang Beton terlebih dahulu auditor melakukan *download* data stok persediaan yang akan diaudit. Perhatikan gambar 5.17 berikut.



Gambar 5.17 Form *download* audit data bahan baku awal

Setelah data stok bahan baku awal tersebut *didownload* maka auditor melakukan audit terhadap bahan baku awal yang sudah *didownload*. Jika auditor sudah selesai melakukan audit terhadap stok persediaan bahan baku awal tersebut, maka auditor melakukan *input* hasil dari audit yang sudah dilakukan. Perhatikan gambar 5.18 berikut ini.

Detail Bahan Baku Awal

Nama Bahan Baku Awal* : Tembaga

Jumlah Bahan Baku Awal* : 466

Stok Minimal* : 10

Stok Maximal* : 666

Satuan* : KG

Harga* : 8700

Untuk isian yang bertanda () tidak bisa di-edit*

Form Hasil Audit Bahan Baku Awal

Status Audit* : Selesai

Hasil Audit* : Cocok

Kondisi Bahan* : Bagus

Type Audit* : Stok Persediaan

Tanggal Audit : 2012-12-20

Untuk isian yang bertanda () harus diisi*

Simpan Batal

Gambar 5.18 Form hasil audit data bahan baku awal

Pada gambar 5.18 dijelaskan, setelah auditor melakukan *input* hasil audit menurut data stok persediaan. Maka data stok persediaan tersebut akan muncul detail untuk hasil audit stok persediaan tersebut. Jadi hasil audit stok persediaan menurut barang yang diaudit. Perhatikan gambar 5.19

Bahan Baku Awal

Cari Berdasarkan : Nama Bahan : Tembaga

Nama Bahan Awal	Jumlah	Stok Minimal	Stok Maximal	Satuan	Harga Awal	Tanggal
Tembaga	466	10	666	KG	Rp. 8.700	20 November 2012

Detail Audit Bahan Baku Awal Tembaga

Nama Bahan Baku	Jumlah	Stok Minimal	Stok Maximal	Satuan	Harga Satuan	Total Harga	Tanggal
Tembaga	466	10	666	KG	Rp. 8.700	Rp. 4.054.200	20 November 2012

Detail Audit Bahan Baku Awal Tembaga

Status Audit	Hasil Audit	Kondisi Bahan	Audit Type	Tanggal Audit
Selesai	Cocok	Bagus	Stok Persediaan	20 Desember 2012

Close

Gambar 5.19 Form detail hasil audit

Mengirim Data Stok Persediaan Akan Diaudit.

Didalam proses ini bagian gudang yang mempunyai hak untuk melakukan proses tersebut. Bagian gudang akan mengirim stok persediaan yang akan diaudit ke bagian auditor. Jika bagian gudang

akan mengirim stok persediaan tersebut, maka gudang terlebih dahulu harus cek detail. Perhatikan gambar 5.20 berikut.

Gambar 5.20 Form detail data audit bahan baku awal

Setelah bagian gudang melihat detail data stok persediaan seperti gambar 5.20, maka selanjutnya gudang me-klik edit pada bagian pengaturan. Fungsinya adalah untuk mengirim stok persediaan ke bagian auditor untuk dilaksanakan audit. Perhatikan gambar 5.21.

Gambar 5.21 Form kirim data bahan baku awal

Pada saat gudang mengirimkan data stok persediaan tersebut, maka fungsi pengaturan pada stok persediaan akan muncul icon , yang fungsinya adalah sebagai informasi bahwa stok tersebut sudah dikirim kepada auditor. Perhatikan gambar 5.22.

Gambar 5.22 Form hasil kirim data bahan baku awal

KESIMPULAN DAN SARAN

Kesimpulan

Berdasarkan uraian pembahasan analisa dan pengujian yang telah dilakukan, maka dapat diambil kesimpulan terhadap sistem informasi audit persediaan barang baja yaitu sebagai berikut :

1. Mengetahui informasi jumlah stok pada saat terjadi transaksi stok.
2. Auditor bisa melakukan audit pada sewaktu – waktu.
3. Auditor bisa me-lock stok setelah dan sebelum melakukan transaksi pada stok persediaan.
4. Untuk mempermudah auditor melakukan input hasil audit yang sudah dilaksanakan.
5. Mempermudah bagian gudang mengirim data stok persediaan yang akan diaudit.

Saran

Saran saran yang dapat diambil dari kesimpulan diatas adalah sebagai berikut :

1. Untuk pengguna aplikasi Sistem Infomasi Audit Persediaan Barang Baja ini, jika menginginkan penambahan pengguna atau penambahan user maka bisa langsung menghubungi pihak admin.
2. Pada Sistem Informasi Audit Persediaan Barang Baja ini, masih belum adanya proses penghitungan laba rugi terhadap stok persediaan masuk dan stok persediaan keluar.
3. Untuk menu login belum ada menu registrasi user baru, sebagai pengguna atau bagian devisi dari Sistem Informasi Audit Persediaan Barang Baja.
4. Untuk hasil *output* seperti laporan audit masih sebatas ruang lingkup audit intern.
5. Belum terdapat notifikasi, jika bagian gudang mengirimkan data stok persediaan ke bagian auditor untuk dilakukan audit stok persediaan.

DAFTAR PUSTAKA

- [1] Dennis, Alan., Wixom, Barbara Haley, 2005. *System Analysis and Design with UML Version 2.0*, Addison-Wesley, Massachusetts.
- [2] Downes, J.E. Goodman, 2003. *Dictionary of Finance & Investment Terms*, Baron's Financial Guides, Canada.
- [3] Jogiyanto H.M., 1995. *Analisis dan Desain Sistem Informasi : Pendekatan Terstruktur Teori dan Praktek Aplikasi Bisnis*. Andi Offset, Yogyakarta.

- [4] Mitchell, John S., 2006. *Physical Audit Persediaan Handbook*, CLARION Technical, Boston.
- [5] Simanjuntak, Binsar, 2008. *Buletin Teknis Standar Akuntansi Pemerintahan Nomor 07*
- [6] Sudrajat, Irwan, 2007. *Lifecycle Asset Management*
- [7] Whitten, Jeffrey L., 2004. *System Analysis and Design Methods*, McGraw-Hill, New York.