

PERANCANGAN SISTEM INFORMASI STOK BARANG BERBASIS WEB DI PT. MAHESA CIPTA

Minda Mora Purba¹, Chaerul Rahmat²

¹karo_727@yahoo.com, ²90arul@gmail.com

¹Universitas Dirgantara Marsekal Suryadarma

²Sekolah Tinggi Teknologi Indonesia

ABSTRAK

Persediaan yang mempunyai arti menyimpan barang dengan sengaja sebagai cadangan untuk keperluan dimasa yang akan datang. Persediaan barang dalam suatu perusahaan memegang peranan penting, mengingat tingkat perputaran barang yang cukup tinggi dan terdiri dari berbagai jenis barang. Pencatatan yang dilakukan secara manual sering kali menjadi masalah ketidaksesuaian data barang yang keluar dengan data yang tertulis. Proses pencarian data secara manual pun membutuhkan waktu yang lama. Untuk itu, perusahaan membutuhkan pengendalian yang baik untuk melindungi harta perusahaan. Penggunaan sistem komputerisasi persediaan barang diharapkan dapat menjadi solusi yang baik untuk mengurangi resiko kehilangan maupun pencurian. Sistem ini dikembangkan menggunakan metode prototype, dengan bahasa pemrograman PHP, dan *Mysql* sebagai *databasenya*.

Kata kunci: web, sistem infomasi, persediaan, PHP Mysql

1. PENDAHULUAN

1.1. Latar Belakang Masalah

Pada beberapa tahun belakangan ini semakin banyak perusahaan yang tumbuh dan berkembang seiring semakin pesatnya perkembangan ekonomi di Indonesia. Hal ini terlihat dengan adanya persaingan yang ketat dalam dunia usaha, baik perdagangan maupun perindustrian. Persaingan yang semakin ketat mengharuskan perusahaan untuk mengelola semua sumber daya yang dimiliki seoptimal mungkin.

Persediaan barang dalam suatu perusahaan memegang peranan yang cukup penting. Peran persediaan semakin penting jika terdiri dari berbagai macam jenis dan dengan tingkat perputaran barang yang cukup tinggi. Salah satu masalah yang sering timbul dalam sebuah persediaan barang secara manual adalah tidak diketahuinya jumlah dan keadaan sisa barang yang tersedia di gudang

dengan pasti. Hal ini dapat mengakibatkan barang tersimpan di gudang terlalu lama, berlebihan, kekurangan, atau bahkan terjadi kehabisan barang. Kerusakan, pemasukan yang tidak benar, lalai untuk mencatat permintaan, barang yang dikeluarkan tidak sesuai dengan pesanan, dan semua kemungkinan lainnya dapat menyebabkan catatan persediaan berbeda dengan persediaan yang sebenarnya ada di gudang. Karena cukup banyak jenis produk dan mobilitas keluar masuk barang, sehingga dikhawatirkan akan terjadi kehilangan atau pencurian stok barang. Akibatnya diperlukan pengendalian internal persediaan yang baik agar tidak terjadi penyelewengan dalam menjalankan tugas. Untuk itu, diperlukan pemeriksaan persediaan secara periodik atas catatan persediaan dengan perhitungan yang sebenarnya. Kebanyakan perusahaan melakukan perhitungan fisik setahun sekali. Namun ada juga yang melakukannya sebulan sekali, seminggu

sekali, bahkan ada yang melakukannya sekali sehari. Karena persediaan sangat rentan terhadap pencurian maupun kerusakan, maka perusahaan membutuhkan pengendalian internal yang baik dengan tujuan untuk melindungi harta perusahaan dan juga agar informasi mengenai persediaan lebih dapat dipercaya. Pengendalian internal persediaan dapat dilakukan dengan melakukan tindakan pengamanan untuk mencegah dari tindakan-tindakan yang menyimpang seperti kehilangan barang, dan salah dalam menghitung maupun mencatat sisa barang yang ada di gudang.

Proses penginputan data, pencarian data serta pembuatan laporan secara manual akan mempunyai resiko kesalahan yang cukup tinggi apalagi dalam menangani data-data yang cukup kompleks dan cukup besar. Proses pencarian data dengan cara konvensional akan memerlukan waktu yang lama. Selain itu, pembuatan laporan stok barang juga akan memerlukan penanganan dalam waktu yang lama. Penggunaan sistem komputerisasi persediaan barang yang baik diharapkan akan mengurangi resiko kehilangan ataupun pencurian terhadap persediaan barang yang ada pada perusahaan, karena setiap jenis barang sudah diklasifikasikan dan memungkinkan untuk diperiksa setiap saat.

Berdasarkan penjelasan pada latar belakang di atas, maka dilakukan penelitian untuk merancang sistem informasi stok barang berbasis web.

1.2. Rumusan Masalah

Berdasarkan latar belakang masalah yang telah dijabarkan di atas, maka didapatkan rumusan masalah yang akan dibahas adalah sebagai berikut :

1. Bagaimana merancang sistem informasi stok barang untuk membantu

penanganan stok barang di PT. Mahesa Cipta?

2. Bagaimana cara mengelola persediaan barang dengan menggunakan sistem informasi berbasis aplikasi *web*?
3. Apakah sistem informasi stok barang berbasis aplikasi *web* dapat menghasilkan laporan keluar masuk barang?
4. Apakah dengan menggunakan sistem informasi stok barang berbasis aplikasi *web* dapat mempermudah pencarian barang?

1.3. Batasan Masalah

Batasan masalah agar pembahasan dan analisa yang dilakukan dapat terarah adalah sebagai berikut:

1. Membuat pengelolaan barang dengan mengacu pada stok barang.
2. Memuat daftar masuk keluar barang yang terpakai beserta nama yang bertanggung jawab atas pengeluaran barang tersebut.
3. Hak akses dibatasi, karena hanya *administrator* yang diberikan *username* dan *password* yang dapat *login*.
4. Memungkinkan *administrator* untuk menambah, edit, dan hapus *item* barang.

1.4. Tujuan Penelitian

Ada beberapa tujuan yang ingin dicapai dalam penulisan penelitian ini sebagai berikut:

1. Mempermudah karyawan dalam proses pencatatan keluar, masuk, dan pemakaian barang serta laporannya.
2. Merancang suatu aplikasi berbasis *web* yang mudah dalam instalasi dan penggunaannya agar tidak terjadi kesalahan pencatatan maupun kehilangan barang.
3. Memperbaiki sistem kerja secara manual dengan sistem kerja yang terkomputerisasi agar mempercepat dan mengefisiensikan waktu kerja.

2. LANDASAN TEORI

2.1. Konsep Persediaan

Persediaan diterjemahkan dari kata “*inventory*” yang merupakan timbunan barang yang secara sengaja disimpan sebagai cadangan (*safety* atau *buffer-stock*) untuk menghadapi kelangkaan pada saat proses produksi sedang berlangsung. Persediaan didefinisikan sebagai barang jadi yang disimpan atau digunakan untuk dijual pada periode mendatang, yang dapat berbentuk bahan baku yang disimpan untuk diproses, barang dalam proses manufaktur dan barang jadi yang disimpan untuk dijual maupun diproses.

Pengertian persediaan menurut Ikatan Akuntan Indonesia (2011;14.5)¹, persediaan diartikan sebagai berikut: Persediaan adalah aset:

- a. Tersedia untuk di jual dalam kegiatan usaha biasa,
- b. Dalam proses produksi untuk penjualan tersebut, atau
- c. Dalam bentuk bahan atau perlengkapan untuk digunakan dalam proses produksi atau pemberian jasa.

Persediaan merupakan suatu hal yang tak terhindarkan. Menurut Baroto (2002:53) mengatakan bahwa penyebab timbulnya persediaan adalah sebagai berikut:

1. Mekanisme pemenuhan atas permintaan
Permintaan terhadap suatu barang tidak dapat dipenuhi seketika bila barang tersebut tidak tersedia sebelumnya. Untuk menyiapkan barang ini diperlukan waktu untuk pembuatan dan pengiriman, maka adanya persediaan merupakan hal yang sulit dihindarkan.

2. Keinginan untuk meredam ketidakpastian

Ketidakpastian terjadi akibat: permintaan yang bervariasi dan tidak pasti dalam jumlah maupun waktu kedatangan, waktu pembuatan yang cenderung tidak konstan antara satu produk dengan produk berikutnya, waktu tenggang (*lead time*) yang cenderung tidak pasti karena banyak faktor yang tidak dapat dikendalikan. Ketidakpastian ini dapat diredam dengan mengadakan persediaan.

3. Keinginan melakukan spekulasi yang bertujuan mendapatkan keuntungan besar dari kenaikan harga di masa mendatang.

Rangkuti (2007:15) menjelaskan adapun fungsi-fungsi persediaan oleh suatu perusahaan/pabrik adalah sebagai berikut :

1. Fungsi *Decoupling*

Adalah persediaan yang memungkinkan perusahaan dapat memenuhi permintaan pelanggan tanpa tergantung pada *supplier*². Persediaan bahan mentah diadakan agar perusahaan tidak akan sepenuhnya tergantung pada pengadaannya dalam hal kuantitas dan waktu pengiriman. Persediaan barang dalam proses diadakan agar departemen-departemen dan proses-proses individual perusahaan terjaga “kebebasannya”. Persediaan barang jadi diperlukan untuk memenuhi permintaan produk yang tidak pasti dari para pelanggan. Persediaan yang diadakan untuk menghadapi fluktuasi³ permintaan konsumen yang tidak dapat diperkirakan atau diramalkan disebut *fluctuation stock*.

2. Fungsi *Economic Lot Sizing*

¹ Ikatan Akuntan Indonesia (2011;14.5). Arti dan pengertian Persediaan

² Usaha menyalurkan atau memasarkan suatu barang (produk)

³ Ketidaktetapan atau ketidakpastian

Persediaan *lot size* ini perlu mempertimbangkan penghematan atau potongan pembelian, biaya pengangkutan per unit menjadi lebih murah dan sebagainya. Hal ini disebabkan perusahaan melakukan pembelian dalam kuantitas yang lebih besar dibandingkan biaya-biaya yang timbul karena besarnya persediaan (biaya sewa gudang, investasi, resiko, dan sebagainya).

3. Fungsi Antisipasi

Apabila perusahaan menghadapi fluktuasi permintaan yang dapat diperkirakan dan diramalkan berdasarkan pengalaman atau data-data masa lalu, yaitu permintaan musiman. Dalam hal ini perusahaan dapat mengadakan persediaan musiman (*seasonal inventories*).

2.1.1. Komponen Biaya Persediaan

Salah satu tujuan persediaan adalah mendapatkan biaya yang minimum. Oleh karena itu, menurut Nasution dan Prasetyawan (2008:121) dalam menentukan biaya persediaan perlu diketahui bahwa biaya-biaya yang mencakup dalam persediaan sebagai berikut:

1. Biaya penyimpanan (*holding costs* atau *carrying costs*), yaitu 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. Biaya-biaya yang termasuk sebagai biaya penyimpanan adalah:
 - a. Biaya fasilitas-fasilitas penyimpanan (termasuk penerangan, pendingin ruangan, dan sebagainya)
 - b. Biaya modal (*opportunity costs of capital*), yaitu *alternative* pendapatan atas dana yang diinvestasikan dalam persediaan
 - c. Biaya keusangan

- d. Biaya penghitungan fisik
- e. Biaya asuransi persediaan
- f. Biaya pajak persediaan
- g. Biaya pencurian, pengrusakan, atau perampokan
- h. Biaya penanganan persediaan dan sebagainya.

Biaya-biaya tersebut di atas merupakan variabel apabila bervariasi dengan tingkat persediaan. Apabila biaya fasilitas penyimpanan (gudang) tidak variabel, tetapi tetap, maka tidak dimasukkan dalam biaya penyimpanan per unit. Biaya penyimpanan persediaan berkisar antara 12 sampai 40 persen dari biaya atau harga barang. Untuk perusahaan manufaktur biasanya, biaya penyimpanan rata-rata secara konsisten sekitar 25 persen.

2. Biaya pemesanan atau pembelian (*ordering costs* atau *procurement costs*).

Biaya-biaya ini meliputi:

- a. Pemrosesan pesanan dan biaya ekspedisi
- b. Upah
- c. Biaya telepon
- d. Pengeluaran surat menyurat
- e. Biaya pengepakan dan penimbangan
- f. Biaya pemeriksaan (inspeksi) penerimaan
- g. Biaya pengiriman ke gudang
- h. Biaya utang lancar dan sebagainya.

Pada umumnya, biaya pemesanan (di luar biaya bahan dan potongan kuantitas) tidak naik apabila kuantitas pemesanan bertambah besar. Tetapi, apabila semakin banyak komponen yang dipesan setiap kali pesan, jumlah pesanan per periode turun, maka biaya pemesanan total akan turun. Ini berarti, biaya pemesanan total per periode (tahunan)

sama dengan jumlah pesanan yang dilakukan setiap periode dikalikan biaya yang harus dikeluarkan setiap kali pesan.

3. Biaya kehabisan atau kekurangan bahan (*shortage costs*) adalah biaya yang timbul apabila persiapan tidak mencukupi adanya permintaan bahan. Biaya-biaya yang termasuk biaya kekurangan bahan adalah sebagai berikut:
 - a. Kehilangan penjualan
 - b. Kehilangan pelanggan
 - c. Biaya pemesanan khusus
 - d. Biaya ekspedisi
 - e. Selisih harga
 - f. Terganggunya operasi
 - g. Tambahan pengeluaran kegiatan manajerial dan sebagainya.

Biaya kekurangan bahan sulit diukur dalam praktik, terutama karena kenyataannya biaya ini sering merupakan *opportunity costs* yang sulit diperkirakan secara objektif.

2.1.2. Sistem Pengendalian Persediaan

Sistem pengendalian persediaan adalah suatu mekanisme mengenai bagaimana mengelola masukan-masukan yang berhubungan dengan persediaan menjadi output, dimana untuk ini diperlukan umpan balik agar output memenuhi standar tertentu. Mekanisme sistem ini adalah pembuatan serangkaian kebijakan yang memonitor tingkat persediaan, menentukan persediaan yang harus dijaga, kapan persediaan harus diisi, dan berapa besar pesanan harus dilakukan. Sistem ini bertujuan untuk menetapkan dan menjamin tersedianya produk jadi, barang dalam proses, komponen dan bahan baku secara optimal, dan pada waktu yang optimal. Kriteria optimal adalah minimasi biaya total yang terkait dengan persediaan, yaitu biaya penyimpanan, biaya pemesanan dan biaya kekurangan persediaan.

Variabel keputusan dalam pengendalian persediaan tradisional dapat diklasifikasikan ke dalam variabel kuantitatif dan variabel kualitatif. Secara kuantitatif, variabel keputusan pada pengendalian sistem persediaan adalah sebagai berikut:

1. Berapa banyak jumlah barang yang akan dipesan atau dibuat,
2. Kapan pemesanan atau pembuatan harus dilakukan,
3. Bagaimana mengendalikan persediaan.

Secara kualitatif, masalah persediaan berkaitan dengan sistem pengoperasian persediaan yang akan menjamin kelancaran pengelolaan persediaan adalah sebagai berikut:

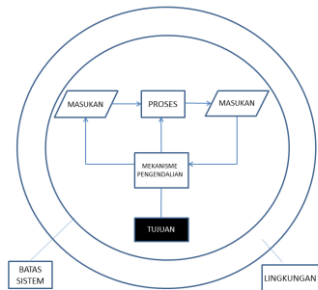
1. Jenis barang apa yang dimiliki,
2. Dimana barang tersebut berada,
3. Berapa jumlah barang yang harus dipesan,
4. Siapa saja yang menjadi pemasok masing-masing item.

Secara luas, tujuan dari sistem pengendalian persediaan adalah menemukan solusi optimal terhadap seluruh masalah yang berkaitan dengan persediaan. Dikaitkan dengan tujuan umum perusahaan, maka optimalitas pengendalian persediaan sering kali diukur dengan keuntungan maksimum yang dicapai. Karena perusahaan memiliki banyak subitem lain selain persediaan, maka mengukur kontribusi pengendalian persediaan dalam mencapai total keuntungan bukan hal mudah. Optimalisasi pengendalian persediaan biasanya diukur dengan total biaya minimum pada suatu periode tertentu (Baroto, 2002:54).

2.2. Karakteristik Sistem

Karakteristik sistem adalah sistem yang mempunyai komponen-komponen, batas sistem, lingkungan sistem,

penghubung, masukan, keluaran, pengolahan dan sasaran.



Gambar II.1. Karakteristik Sistem

1. Komponen Sistem (*components system*)

Komponen-komponen sistem tersebut dapat berupa suatu bentuk subsistem. Setiap subsistem memiliki sifat dari sistem yang menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan.

Elemen-elemen yang lebih kecil yang disebut subsistem, misalkan sistem komputer terdiri dari sub sistem perangkat keras, perangkat lunak dan manusia. Elemen-elemen yang lebih besar yang disebut supra sistem. Misalkan bila perangkat keras adalah sistem yang memiliki sub sistem CPU⁴, perangkat I/O⁵ dan memori, maka supra sistem perangkat keras adalah sistem komputer.

2. Batasan Sistem (*boundary*)

Batas sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batas sistem ini memungkinkan suatu sistem dipandang sebagai suatu kesatuan. Batas suatu sistem menunjukkan ruang lingkup dari sistem tersebut. Batasan sistem ini memungkinkan suatu sistem

dipandang sebagai satu kesatuan yang tidak dapat dipisahkan.

3. Lingkungan Luar Sistem (*environments*)

Lingkungan dari sistem adalah apapun di luar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan dapat juga bersifat merugikan sistem tersebut. Lingkungan luar yang menguntungkan merupakan energi dari sistem dan dengan demikian harus tetap dijaga dan dipelihara. Sedang lingkungan luar yang merugikan harus ditahan dan dikendalikan, kalau tidak akan mengganggu kelangsungan hidup dari sistem.

4. Penghubung Sistem (*Interface*)

Penghubung merupakan media perantara antar subsistem. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem lainnya. Keluaran (*output*) dari satu sub sistem akan menjadi masukan (*input*) untuk subsistem yang lainnya dengan melalui penghubung. Dengan penghubung satu subsistem dapat berinteraksi dengan subsistem yang lainnya membentuk satu kesatuan. Media yang menghubungkan sistem dengan subsistem yang lain disebut penghubung sistem atau interface.

5. Masukan Sistem (*Input*)

Masukan adalah energi yang dimasukkan ke dalam sistem. Masukan dapat berupa *maintenance input* dan sinyal *input*. *Maintenance input* adalah energi yang dimasukkan supaya sistem tersebut dapat beroperasi. Sinyal *input* adalah energi yang diproses untuk didapatkan keluaran.

⁴ Central Processing Unit

⁵ Input/Output

6. **Pengolahan Sistem (*Process*)**
Suatu sistem dapat mempunyai suatu bagian pengolah atau sistem itu sendiri sebagai pengolahnya. Pengolah yang akan merubah masukan menjadi keluaran. Suatu sistem produksi akan mengolah masukan berupa bahan baku dan bahan-bahan yang lain menjadi keluaran berupa barang jadi atau data yang dibutuhkan sesuai permintaan dari *user* (pengguna).
7. **Keluaran Sistem (*Output*)**
Keluaran adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan. Keluaran dapat merupakan masukan untuk subsistem yang lain atau kepada supra sistem.
8. **Sasaran dan Tujuan Sistem (*Objective and Goal*)**
Suatu sistem pasti mempunyai tujuan atau sasaran yang pasti dan bersifat deterministik. Kalau suatu sistem tidak mempunyai sasaran, maka operasi sistem tidak akan ada gunanya. Sasaran dari sistem sangat menentukan sekali masukan yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuan yang telah direncanakan.

2.3. Konsep Dasar Informasi

Definisi dari informasi adalah data yang diolah menjadi bentuk yang lebih berarti bagi yang menerimanya. Menurut *Raymond Mcleod* informasi adalah data yang telah diolah menjadi bentuk yang memiliki arti bagi si penerima dan bermanfaat bagi pengambilan keputusan saat ini atau mendatang. Secara umum informasi dapat di definisikan sebagai hasil pengolahan data dalam bentuk yang lebih berguna dan lebih berarti bagi penerimanya.

Sumber dari informasi adalah data. Data adalah kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata. Kejadian-kejadian adalah sesuatu yang terjadi pada saat tertentu. Data ini akan bermanfaat jika hasil dari informasi yang ada telah sesuai dengan kemauan si penerima dan menjadi bahan masukan untuk melakukan tindakan selanjutnya.

Data merupakan bentuk yang masih mentah, belum dapat bercerita banyak sehingga perlu diolah lebih lanjut. Data diolah melalui suatu metode untuk menghasilkan informasi. Data yang diolah melalui suatu model menjadi informasi, penerima kemudian menerima informasi tersebut, membuat suatu keputusan dan melakukan tindakan, yang berarti menghasilkan suatu tindakan yang lain yang akan membuat sejumlah data kembali.

Fungsi-fungsi informasi adalah sebagai berikut:

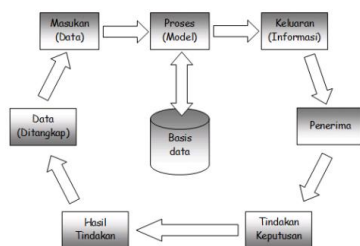
- a. Untuk meningkatkan pengetahuan bagi si pemakai
- b. Untuk mengurangi ketidakpastian dalam proses pengambilan keputusan
 1. pemakai
- c. Menggambarkan keadaan yang sebenarnya dari sesuatu hal.

Kualitas informasi bias terhadap *error*, karena kesalahan cara pengukuran dan pengumpulan, kegagalan mengikuti prosedur pemrosesan, kehilangan atau data tidak terproses, kesalahan perekaman atau koreksi data, kesalahan file histori atau master, kesalahan prosedur pemrosesan ketidakberfungsian sistem. Informasi yang berkualitas harus akurat, tepat pada waktunya dan relevan, maksudnya adalah:

- a. Akurat berarti informasi harus bebas dari kesalahan-kesalahan dan tidak bias atau menyesatkan.

- b. Tepat waktu berarti informasi yang datang pada penerima tidak boleh terlambat. Informasi yang sudah usang tidak akan mempunyai nilai lagi. Karena informasi merupakan landasan di dalam pengambilan keputusan.
- c. Relevan berarti informasi tersebut mempunyai manfaat untuk pemakainya. Relevansi informasi untuk tiap-tiap orang berbeda-beda.

Nilai Informasi ditentukan dari dua hal, yaitu manfaat dan biaya mendapatkannya. Suatu informasi dikatakan bernilai bila manfaatnya lebih efektif dibandingkan dengan biaya mendapatkannya. Pengukuran nilai informasi biasanya dihubungkan dengan analisis *cost effectiveness*⁶ atau *cost benefit*⁷. Biaya mendapatkan informasi biasanya tergantung dari proses informasi yang didapatkan oleh penerima untuk kapan digunakan data tersebut. Informasi akan dinilai mahal apabila informasi tersebut didapatkan secara cepat dan dalam kejadian yang sedang berlangsung saat itu juga.



Gambar II.2 Siklus Informasi

2.3.1. Jenis-Jenis Informasi

1. Informasi Berdasarkan Fungsi

Jenis informasi berdasarkan fungsi adalah informasi yang berdasarkan materi dan kegunaannya. Informasi jenis ini meliputi:

- Informasi yang menambah pengetahuan, misalnya peristiwa bencana alam.
- Informasi yang mengajari membaca, misalnya informasi pendidikan.
- Informasi edukatif, misalnya tulisan trik menjadi mahasiswa aktif, tips menjadi siswa berprestasi, dan lain-lain.

2. Informasi berdasarkan format penyajian

Informasi jenis ini adalah informasi berdasarkan bentuk penyajian informasi. Jenis informasi ini antara lain seperti lukisan, foto, karikatur, tulisan teks, dan lain-lain.

3. Informasi berdasarkan lokasi peristiwa

Informasi berdasarkan lokasi peristiwa adalah informasi berdasarkan lokasi peristiwa berlangsung. Misalnya seperti informasi dari dalam negeri dan informasi dari luar negeri.

4. Informasi berdasarkan bidang kehidupan

Informasi berdasarkan kehidupan adalah informasi berdasarkan bidang-bidang kehidupan yang ada, misalnya pendidikan, sastra, budaya, ilmu pendidikan, dan sebagainya.

5. Informasi berdasarkan penyampaiannya

Informasi berdasarkan penyampaiannya meliputi informasi yang disediakan secara berkala, secara tiba-tiba, setiap saat, atau diperoleh berdasarkan permintaan.

2.4. Perancangan Sistem

Perancangan adalah penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi (Syifaun Nafisah, 2003:2).

⁶ Keefektifan biaya

⁷ Biaya manfaat

Perancangan sistem dirancang dalam bentuk bagan alir sistem (*system Flowchart*) yang merupakan alat bentuk grafik yang dapat digunakan untuk menunjukkan urutan-urutan proses dari sistem. Terdapat beberapa teori mengenai pengertian perancangan sistem, diantaranya :

- Menurut Satzinger, Jackson, dan Burd (2012:5) perancangan sistem adalah kegiatan merancang detil dan rincian dari sistem yang akan dibuat sehingga sistem tersebut sesuai dengan *requirement* yang sudah ditetapkan dalam tahap analisa sistem.
- O'Brien dan Marakas (2009:639) menjelaskan bahwa perancangan sistem adalah sebuah kegiatan dari hasil analisa sistem sehingga dapat memenuhi kebutuhan dari pengguna termasuk diantaranya perancangan *user interface*, data dan aktivitas proses.
- Bentley dan Whitten (2009:160) melalui buku yang berjudul "*system analysis and design for the global enterprise*" menjelaskan bahwa perancangan sistem adalah teknik pemecahan masalah dengan melengkapi komponen-komponen kecil menjadi kesatuan komponen sistem kembali ke sistem yang lengkap. Teknik ini diharapkan dapat menghasilkan sistem yang lebih baik.

2.5. Alat Bantu Pemodelan

Pemodelan sistem merupakan cara untuk memudahkan dalam melakukan perancangan perangkat lunak. Dimana tujuan dari pemodelan ini adalah untuk memahami perilaku sistem yang akan dirancang dengan lebih mendetail dan menggambarkan sistem secara lengkap dan sederhana sehingga dapat digunakan untuk menjelaskan jalannya sistem. Dalam perancangan suatu sistem informasi, dibutuhkan beberapa alat

bantu perancangan sistem agar analisa dan hasil yang ingin dicapai dapat mencapai sebuah hasil yang maksimal. Alat bantu (tools) yang digunakan dalam menganalisis perangkat lunak ini adalah sebagai berikut:

2.5.1. Unified Modeling Language (UML)

Unified Modeling Language adalah bahasa standar yang digunakan untuk menjelaskan dan memvisualisaikan artifak dari proses analisis dan disain berorientasi objek. UML menyediakan standar pada notasi dan diagram yang bisa digunakan untuk memodelkan suatu sistem. UML dikembangkan oleh 3 pendekar "berorientasi objek", yaitu Grady Booch, Jim Rumbaugh, dan Ivar Jacobson. UML menjadi bahasa yang bisa digunakan untuk berkomunikasi dalam perspektif objek antara user dengan developer, antara developer dengan developer, antara developer analis dengan developer disain, dan antara developer disain dengan developer pemrograman.

UML memungkinkan developer melakukan permodelan secara visual, yaitu penekanan pada penggambaran, bukan didominasi oleh narasi. Permodelan visual membantu untuk menangkap struktur dari objek, mempermudah penggambaran interaksi antara elemen dalam sistem, dan mempertahankan konsistensi antara disain dan implementasi dalam pemrograman.

Beberapa tujuan atau fungsi dari penggunaan UML, diantaranya:

- Dapat memberikan bahasa pemodelan visual kepada pengguna dari berbagai macam pemrograman maupun proses rekayasa.
- Dapat menyatukan praktek-praktek terbaik yang ada dalam pemodelan.
- Dapat memberikan model yang siap untuk digunakan, merupakan bahasa

pemodelan visual yang ekspresif untuk mengembangkan sistem dan untuk saling menukar model secara mudah.







- Dapat berguna sebagai blue print, sebab sangat lengkap dan detail dalam perancangannya yang nantinya akan diketahui informasi yang detail mengenai koding suatu program.
- Dapat memodelkan sistem yang berkonsep berorientasi objek, jadi tidak hanya digunakan untuk memodelkan perangkat lunak (*software*) saja.
- Dapat menciptakan suatu bahasa permodelan yang nantinya dapat dipergunakan oleh manusia maupun oleh mesin.

Sedangkan jenis-jenis diagram UML, yaitu:

a. Use case Diagram

Use Case diagram yaitu salah satu jenis diagram pada UML yang menggambarkan interaksi antara sistem dan aktor, use case diagram juga dapat men-deskripsikan tipe interaksi antara si pemakai sistem dengan sistemnya.







Tabel II.1 Use Case Diagram

Simbol	Keterangan
	Aktor : Mewakili peran orang, sistem yang lain, atau alat ketika berkomunikasi
	Use Case : Abstraksi dan interaksi antara sistem dan aktor
	Association : Abstraksi dari penghubung antara aktor dengan use case
	Generalisasi : Menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan use case
	Menunjukkan bahwa suatu use case seluruhnya merupakan fungsionalitas dari use case lainnya
	Menunjukkan bahwa suatu use case merupakan tambahan fungsional dari use case lainnya jika suatu kondisi terpenuhi

b. Activity Diagram

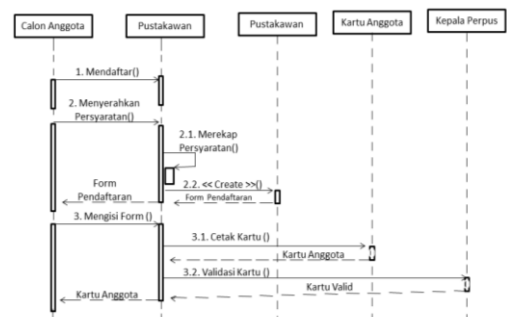
Activity diagram atau diagram aktivitas yaitu salah satu jenis diagram pada UML yang dapat memodelkan proses-proses apa saja yang terjadi pada sistem.

Tabel II.2 Activity Diagram

No	Gambar	Nama	Keterangan
1		Activity	Memperlihatkan bagaimana masing-masing kelas saling berinteraksi satu sama lain
2		Action	State dari sistem yang mencerminkan eksekusi dari suatu aksi
3		Initial Node	Bagaimana objek di bentuk atau diawali
4		Activity final node	Bagaimana objek di bentuk diakhiri
5		Decision	Digunakan untuk menggambarkan suatu keputusan /tindakan yang harus diambil pada kondisi tertentu
6		Line Connector	Digunakan untuk menghubungkan satu simbol dengan simbol lainnya

c. Sequence Diagram

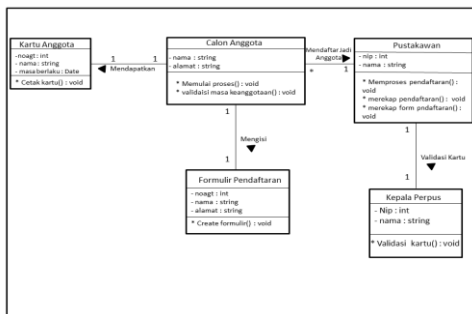
Sequence diagram yaitu salah satu jenis diagram pada UML yang menjelaskan interaksi objek yang berdasarkan urutan waktu, sequence diagram juga dapat menggambarkan urutan atau tahapan yang harus dilakukan untuk dapat menghasilkan sesuatu seperti pada use case diagram.



Gambar II.3 Contoh Sequence Diagram

d. Class Diagram

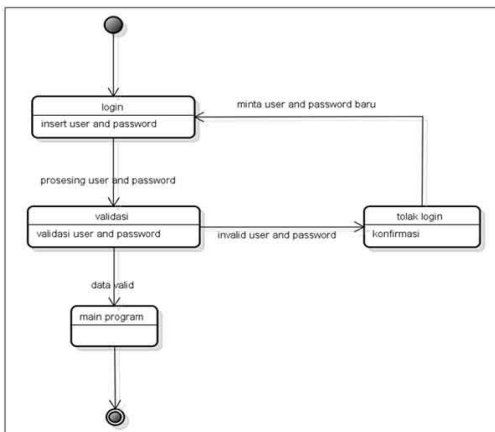
Class diagram yaitu salah satu jenis diagram pada UML yang digunakan untuk menampilkan kelas-kelas maupun paket-paket yang ada pada suatu sistem yang nantinya akan digunakan. Jadi diagram ini dapat memberikan sebuah gambaran mengenai sistem maupun relasi-relasi yang terdapat pada sistem tersebut.



Gambar II.4 Contoh Class Diagram

e. State Machine Diagram

State machine diagram yaitu salah satu jenis diagram pada UML yang menggambarkan transisi maupun perubahan keadaan suatu objek pada sistem.

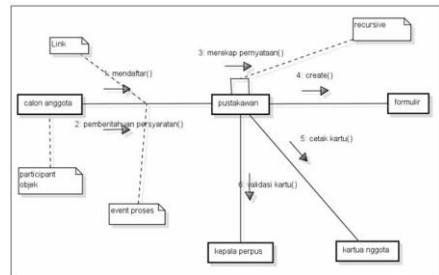


Gambar II.5 Contoh State Machine Diagram

f. Communication Diagram

Communication diagram yaitu salah satu jenis diagram pada UML yang dapat menggambarkan tahapan terjadinya suatu aktivitas dan diagram ini

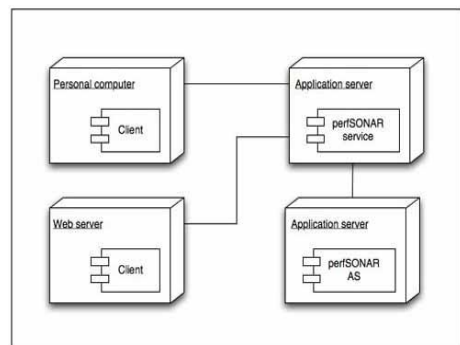
juga menggambarkan interaksi antara objek yang ada pada sistem. Hampir sama seperti sequence diagram akan tetapi communication diagram lebih menekankan kepada peranan masing-masing objek pada sistem.



Gambar II.6 Contoh Communication Diagram

g. Deployment Diagram

Deployment diagram yaitu salah satu diagram pada UML yang menunjukkan tata letak suatu sistem secara fisik, dapat juga dikatakan untuk menampilkan bagian-bagian software yang terdapat pada hardware dan digunakan untuk menerapkan suatu sistem dan hubungan antara komponen hardware. Jadi Deployment diagram intinya untuk menunjukkan letak software pada hardware yang digunakan sistem.



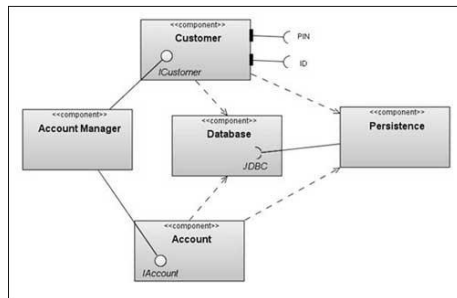
Gambar II.7 Contoh Deployment Diagram

f. Communication Diagram

h. Component Diagram

Component diagram yaitu salah satu jenis diagram pada UML yang menggambarkan software pada suatu

sistem. Component diagram merupakan penerapan software dari satu ataupun lebih class, dan biasanya berupa file data atau .exe, source code, table, dokumen dsb.



Gambar II.8 Contoh Component Diagram

2.6. Perangkat Lunak Pendukung

Komputer tidak akan lepas dari perangkat lunak (*software*) karena merupakan salah satu bagian terpenting dari komputer itu sendiri. Komputer merupakan sistem elektronik yang fungsinya memanipulasi data yang cepat dan tepat serta akurat yg telah di rancang dan diorganisasikan supaya secara otomatis menerima atau menyimpan data *input* atau masukan, kemudian memprosesnya dan menghasilkan *output* di bawah pengawasan suatu langkah-langkah instruksi program yg tersimpan di memori (*stored program*). Berikut dijelaskan beberapa aplikasi (*software*) yang digunakan dalam pembuatan aplikasi sistem stok barang berbasis web sebagai berikut:

2.6.1. Database

Basis data (*database*) adalah kumpulan data yang disimpan secara sistematis di dalam komputer yang dapat diolah atau dimanipulasi menggunakan perangkat lunak (program aplikasi) untuk menghasilkan informasi. Pendefinisian basis data meliputi spesifikasi berupa tipe data, struktur data dan juga batasan-batasan pada data yang akan disimpan. Basis data merupakan aspek yang sangat

penting dalam sistem informasi karena berfungsi sebagai gudang penyimpanan data yang akan diolah lebih lanjut. Basis data menjadi penting karena dapat mengorganisasi data, menghindari duplikasi data, menghindari hubungan antar data yang tidak jelas dan juga *update* yang rumit.

Proses memasukkan dan mengambil data ke dan dari media penyimpanan data memerlukan perangkat lunak yang disebut dengan sistem manajemen basis data (*database management system*). DBMS merupakan sistem perangkat lunak yang memungkinkan pengguna basis data (*database user*) untuk memelihara, mengontrol dan mengakses data secara praktis dan efisien. Dengan kata lain, semua akses ke basis data akan ditangani oleh DBMS. DBMS ini menjadi lapisan yang menghubungkan basis data dengan program aplikasi untuk memastikan bahwa basis data tetap terorganisasi secara konsisten dan dapat diakses dengan mudah.

Ada beberapa fungsi yang harus ditangani DBMS seperti mengolah pendefinisian data, menangani permintaan pengguna untuk mengakses data, memeriksa sekuriti dan integriti data yang didefinisikan oleh DBA (*Database Administrator*), menangani kegagalan dalam pengaksesan data yang disebabkan oleh kerusakan sistem maupun media penyimpanan (*disk*) dan juga menangani unjuk kerja semua fungsi secara efisien. Tujuan utama DBMS adalah untuk memberikan tinjauan abstrak data kepada pengguna.

Jadi sistem menyembunyikan informasi tentang bagaimana data disimpan, dipelihara dan juga bisa diakses secara efisien. Pertimbangan efisien di sini adalah rancangan struktur data yang kompleks tetapi masih bisa digunakan

oleh pengguna awam tanpa mengetahui kompleksitas strukturnya. Adapun Manfaat Penggunaan Database, yaitu:

- Kecepatan dan Kemudahan
Database memiliki kemampuan dalam menyeleksi data sehingga menjadi suatu kelompok yang terurut dengan cepat. Hal inilah yang akhirnya dapat menghasilkan informasi yang dibutuhkan secara cepat pula. Seberapa cepat pemrosesan data oleh *database* tergantung pula pada perancangan databasenya.
- Pemakaian Bersama-sama
Suatu *database* bisa digunakan oleh siapa saja dalam suatu perusahaan. Sebagai contoh *database* mahasiswa dalam suatu perguruan tinggi dibutuhkan oleh beberapa bagian, seperti bagian admin, bagian keuangan, bagian akademik. Kesemua bidang tersebut membutuhkan *database* mahasiswa namun tidak perlu masing-masing bagian membuat *databasenya* sendiri, cukup *database* mahasiswa satu saja yang disimpan di *server* pusat. Nanti aplikasi dari masing-masing bagian bisa terhubung ke *database* mahasiswa tersebut.
- Kontrol data terpusat
Masih berkaitan dengan point ke dua, meskipun pada suatu perusahaan memiliki banyak bagian atau divisi tapi *database* yang diperlukan tetap satu saja. Hal ini mempermudah pengontrolan data seperti ketika ingin mengupdate data mahasiswa, maka kita perlu *mengupdate* semua data di masing-masing bagian atau divisi, tetapi cukup di satu *database* saja yang ada di server pusat.
- Menghemat biaya perangkat
Dengan memiliki *database* secara terpusat maka di masing-masing divisi tidak memerlukan perangkat untuk menyimpan database berhubungan *database* yang dibutuhkan

hanya satu yaitu yang disimpan di *server* pusat, ini tentunya memangkas biaya pembelian perangkat.

- Keamanan Data
Hampir semua aplikasi manajemen database sekarang memiliki fasilitas manajemen pengguna. Manajemen pengguna ini mampu membuat hak akses yang berbeda-beda disesuaikan dengan kepentingan maupun posisi pengguna. Selain itu data yang tersimpan di *database* diperlukan *password* untuk mengaksesnya.
- Memudahkan dalam pembuatan Aplikasi baru
Dalam poin ini *database* yang dirancang dengan sangat baik, sehingga jika perusahaan memerlukan aplikasi baru tidak perlu membuat database yang baru juga, atau tidak perlu mengubah kembali struktur database yang sudah ada. Sehingga pembuat aplikasi atau *programmer* hanya cukup membuat atau mengatur antarmuka aplikasinya saja.

2.6.2. XAMPP

XAMPP ialah perangkat lunak bebas yang mendukung banyak sistem operasi, merupakan campuran dari beberapa program. Yang mempunyai fungsi sebagai *server* yang berdiri sendiri (*localhost*), yang terdiri dari program MySQL database, Apache HTTP Server, dan penerjemah ditulis dalam bahasa pemrograman PHP dan Perl.

Nama XAMPP merupakan singkatan dari X (empat sistem operasi), Apache, MySQL, PHP dan Perl. Program ini tersedia di bawah GNU *General Public License* dan bebas, adalah mudah untuk menggunakan web server yang dapat melayani tampilan halaman web yang dinamis. Melalui program ini, *programmer web* dapat menguji aplikasi web yang dikembangkan dan

mempresentasikannya ke pihak lain secara langsung dari komputer, tanpa perlu terkoneksi ke internet. XAMPP juga dilengkapi fitur manajemen database PHPMyAdmin seperti pada server hosting sungguhan, sehingga pengembangan web dapat mengembangkan aplikasi web berbasis database secara mudah.

XAMPP adalah singkatan yang setiap huruf adalah:

- X : Program ini dapat dijalankan di banyak sistem operasi, seperti Windows, Linux, Mac OS, dan Solaris.
- A : Apache, server aplikasi Web. Apache tugas utama adalah untuk menghasilkan halaman web yang benar kepada pengguna terhadap kode PHP yang sudah dituliskan oleh pembuat halaman web. jika perlu kode PHP juga berdasarkan yang tertulis, dapat database diakses dulu (misalnya MySQL) untuk mendukung halaman web yang dihasilkan.
- M : MySQL, server aplikasi database. Pertumbuhannya disebut SQL singkatan dari Structured Query Language. SQL merupakan bahasa terstruktur yang difungsikan untuk mengolah database. MySQL dapat digunakan untuk membuat dan mengelola database dan isinya. Bisa juga memanfaatkan MySQL guna untuk menambahkan, mengubah, dan menghapus data dalam database.
- P : PHP, bahasa pemrograman web. Bahasa pemrograman PHP adalah bahasa pemrograman untuk membuat web yang server-side scripting. PHP digunakan untuk membuat halaman web dinamis. Sistem manajemen database yang sering digunakan dengan PHP adalah MySQL. namun PHP juga mendukung Pengelolaan sistem database Oracle, Microsoft

Access, Interbase, d-base, PostgreSQL, dan sebagainya.

- P : Perl, bahasa pemrograman untuk semua tujuan, pertama kali dikembangkan oleh Larry Wall, mesin Unix. Perl dirilis pertama kali tanggal 18 Desember 1987 yang ditandai dengan keluarnya Perl 1. Pada versi-versi selanjutnya, Perl juga tersedia untuk berbagai sistem operasi Unix (SunOS, Linux, BSD, HP-UX), juga tersedia untuk sistem operasi seperti DOS, Windows, PowerPC, BeOS, VMS, EBCDIC, dan PocketPC.

2.6.3. MySQL

MySQL merupakan sebuah perangkat lunak atau *software* sistem manajemen basis data SQL atau DBMS *Multithread* dan multi user. MySQL sebenarnya merupakan turunan dari salah satu konsep utama dalam *database* untuk pemilihan atau seleksi dan pemasukan data yang memungkinkan pengoperasian data dikerjakan secara mudah dan otomatis. MySQL diciptakan oleh Michael "Monty" Widenius pada tahun 1979, seorang *programmer* komputer asal Swedia yang mengembangkan sebuah sistem *database* sederhana yang dinamakan UNIREG yang menggunakan koneksi low-level ISAM *database engine* dengan *indexing*.

Adapun kelebihan MySQL dalam penggunaannya dalam *database* adalah:

- Free atau gratis sehingga MySQL dapat dengan mudah untuk mendapatkannya.
- MySQL stabil dan tangguh dalam pengoperasiannya
- MySQL mempunyai sistem keamanan yang cukup baik
- Sangat mendukung transaksi dan mempunyai banyak dukungan dari komunitas

- Sangat fleksibel dengan berbagai macam program
- Perkembangan dari MySQL sangat cepat

Selain kelebihan yang disampaikan diatas, ada beberapa kekurangan yang dimiliki oleh mySQL, diantaranya:

- Kurang mendukung koneksi bahasa pemrograman seperti *Visual basic* atau biasa kita kenal dengan sebutan VB, Foxpro, Delphi dan lain-lain sebab koneksi ini menyebabkan *field* yang dibaca harus sesuai dengan koneksi dari bahasa pemrograman visual tersebut.
- Data yang dapat ditangani belum besar dan belum mendukung *widowing function*.

SQL dapat digunakan secara berdiri sendiri maupun di lekatkan pada bahasa pemrograman seperti C, dan Delphi. Elemen SQL Elemen dari SQL yang paling dasar antara lain pernyataan, nama, tipe data, ekspresi, konstanta dan fungsi bawaan.

- Pernyataan
Perintah dari SQL yang digunakan untuk meminta sebuah tindakan kepada DBMS. Pernyataan dasar SQL antara lain :
 1. ALTER : Merubah struktur tabel
 2. COMMIT : Mengakhiri eksekusi transaksi
 3. CREATE : Membuat tabel, indeks
 4. DELETE : Menghapus baris pada sebuah tabel
 5. DROP : Menghapus tabel, indeks
 6. GRANT : Menugaskan hak terhadap basis data kepada user
 7. INSERT : Menambah baris pada tabel
 8. REVOKE : Membatalkan hak kepada basis data
 9. ROLLBACK : Mengembalikan pada keadaan semula apabila transaksi gagal dilaksanakan

10. SELECT : Memilih baris dan kolom pada sebuah tabel
11. UPDATE : Mengubah value pada baris sebuah tabel

- Nama
Nama digunakan sebagai identitas, yaitu identitas bagi objek pada DBMS. Misal : tabel, kolom dan pengguna.

- Tipe data
Tipe data yang ada dalam MYSQL:
 - a. Tipe data numerik antara lain:
 1. TINYINT : Nilai integer yang sangat kecil
 2. SMALLINT : Nilai integer yang kecil
 3. MEDIUMINT : Nilai integer yang sedang
 4. INT : Nilai integer dengan nilai standar
 5. BIGINT : Nilai integer dengan nilai besar
 6. FLOAT : Bilangan decimal dengan single-precision
 7. DOUBLE : Bilangan decimal dengan double-precision
 8. DECIMAL(M,D) : Bilangan float yang dinyatakan sebagai string. M : jumlah digit yang disimpan, D : jumlah angka dibelakang koma.

- b. Tipe data String antara lain :
 1. CHAR : Karakter yang memiliki panjang tetap sebanyak n
 2. VARCHAR : Karakter yang memiliki panjang tidak tetap yaitu maksimum n
 3. TINYBLOB : BLOB dengan ukuran sangat kecil
 4. BLOB : BLOB yang memiliki ukuran kecil
 5. MEDIUMBLOB : BLOB yang memiliki ukuran sedang
 6. LONGBLOB : BLOB yang memiliki ukuran besar
 7. TINYTEXT : teks dengan ukuran sangat kecil

8. TEXT : teks yang memiliki ukuran kecil
 9. MEDIUMTEXT : teks yang memiliki ukuran sedang
 10. LONGTEXT : teks yang memiliki ukuran besar
 11. ENUM : kolom diisi dengan satu member enumerasi
 12. SET : Kolom dapat diisi dengan beberapa nilai anggota himpunan
- c. Tipe data tunggal dan jam :
1. DATE : date memiliki format tahun-bulan-tanggal
 2. TIME : time memiliki format jam-menit-detik
 3. DATETIME : gabungan dari format date dan time
- Ekspresi
Ekspresi digunakan untuk menghasilkan/menghitung nilai. Misalnya: jumlah = harga-diskon. Ekspresi aritmatika antara lain:
 1. + : tambah
 2. - : kurang
 3. / : bagi
 4. * : kali
 - Fungsi bawaan
Fungsi adalah subprogram yang dapat menghasilkan suatu nilai apabila fungsi tersebut dipanggil. Fungsi Agregat adalah fungsi yang digunakan untuk melakukan summary, statistik yang dilakukan pada suatu tabel/query.
 1. AVG(ekspresi) : digunakan untuk mencari nilai rata-rata dalam kolom dari tabel.
 2. COUNT(x) : digunakan untuk menghitung jumlah baris dari sebuah kolom dari tabel
 3. MAX(ekspresi) : digunakan untuk mencari nilai yang paling besar dari suatu

2.6.4. Bahasa Pemrograman PHP

Sejarah PHP bermula pada tahun 1994 ketika programmer kelahiran

Denmark yang sekarang berdomisili di Canada, Rasmus Lerdorf membuat sebuah script (kode program) dengan bahasa Perl untuk web pribadinya. Salah satu kegunaan script ini adalah untuk menampilkan resume pribadi dan mencatat jumlah pengunjung ke sebuah website.

Dengan alasan untuk meningkatkan performa, Rasmus Lerdorf kemudian membuat ulang kode program tersebut dalam bahasa C. Ia juga mengembangkannya lebih lanjut sehingga memiliki script tersebut memiliki kemampuan untuk memproses form HTML dan berkomunikasi dengan database.

Lerdorf menyebut kode program ini sebagai Personal Home Page/Forms Interpreter atau PHP/FI. Inilah asal mula penamaan PHP digunakan. PHP/FI dapat digunakan untuk membuat aplikasi web dinamis sederhana. Lerdorf kemudian merilis kode tersebut ke publik dengan sebutan Personal Home Page Tools (PHP Tools) version 1.0. Perilisan ini diumumkan pada 8 Juni 1995 di alamat comp.infosystems.www.authoring.cgi, sebuah group diskusi Usenet.

Alamat website yang tertera di atas sudah tidak dipakai lagi. Dari pengumuman ini kita dapat mengetahui kebutuhan programmer saat itu, seperti mencari cara untuk membuat halaman yang dapat diproteksi dengan password, kemudahan dalam penanganan form, dan kemampuan mengakses data dari halaman lain.

Pengumuman itu juga mencerminkan bahwa PHP pada awalnya adalah sebuah framework bahasa C yang memiliki banyak fungsi dalam mengembangkan web, bukan sebuah bahasa pemrograman utuh seperti yang kita kenal sekarang.

PHP disebut bahasa pemrograman *server side* karena PHP diproses pada komputer *server*. Hal ini berbeda dibandingkan dengan bahasa pemrograman *client-side* seperti JavaScript yang diproses pada web browser (client).

Dalam beberapa tahun perkembangannya, PHP menjelma menjadi bahasa pemrograman web yang powerful dan tidak hanya digunakan untuk membuat halaman web sederhana, tetapi juga website populer yang digunakan oleh jutaan orang seperti wikipedia, wordpress, joomla, dll.

Saat ini PHP adalah singkatan dari PHP: Hypertext Preprocessor, sebuah kepanjangan rekursif, yakni permainan kata dimana kepanjangannya terdiri dari singkatan itu sendiri: PHP: *Hypertext Preprocessor*. PHP dapat digunakan dengan gratis (free) dan bersifat Open Source. PHP dirilis dalam lisensi *PHP License*, sedikit berbeda dengan lisensi GNU *General Public License* (GPL) yang biasa digunakan untuk proyek *Open Source*. Untuk pembuatan web, kode PHP biasanya di sisipkan kedalam dokumen HTML. Karena fitur inilah PHP disebut juga sebagai *Scripting Language* atau bahasa pemrograman script.

Kemampuan string-parsing juga bisa dianggap sebagai banyaknya kemungkinan yang ditawarkan oleh PHP. Dalam hal ini, PHP memiliki lebih dari 85 function untuk memanipulasi string. Kelebihan ini tak hanya akan menawarkan keleluasaan untuk melakukan operasi string.

Cara penulisan PHP :

PHP bersifat case sensitif yang artinya semua penulisannya harus sesuai dengan kamus data yang tersedia. Secara

mendasar php memiliki struktur sebagai berikut:

```
<?php
//Ini contoh komentar dalam php
echo "<h2>Contoh sederhana penulisan
PHP</h2>";
$nama="Satria Multimedia";
echo "<p>$nama</p>";
$nilai=10;
$hasil=$nilai+10;
echo $hasil;
?>
```

Kode diatas adalah contoh sederhana struktur PHP, berikut penjelasannya:

1. Awal kode harus diawali dengan "<?php" dan diakhiri dengan "?>", dua perintah tersebut harus ada pada tiap bagian yang memanggil fungsi php.
2. Dalam barisan perintah php, kita bebas menyisipkan komentar dengan diawali tanda "//". Komentar digunakan untuk mempermudah mengingat kumpulan baris dengan proses yang sama
3. echo digunakan untuk mengirimkan hasil ke browser, sehingga yang dikirimkan adalah bagian yang berada setelah perintah echo
4. setiap variabel harus diawali dengan tanda dolar "\$" dan dapat dipanggil kapanpun dalam halaman yang sama.

2.6.5. Notepad ++

Notepad++ adalah suatu teks editor yang berjalan pada *Operating System* (OS). Notepad ++ menggunakan komponen Scintilla untuk dapat menampilkan dan menyunting teks dan berkas kode sumber dari berbagai bahasa pemrograman. Notepad++ didistribusikan sebagai *software gratis* (*Free*) dan dapat digunakan sebagai perangkat lunak bebas atau *Open Source* yang memiliki lisensi dari GNU *General Public License*.

Notepad++ bisa mengenal tag dan kode dalam berbagai bahasa pemrograman. Fitur pencarian tingkat lanjut dan pengeditan teks yang tersedia juga cukup ampuh, dan sangat membantu tugas seorang *programmer* atau *developer* dalam menyelesaikan sebuah skrip kode dalam programnya. Program Notepad++ banyak digunakan oleh kalangan pengguna komputer di bidang bahasa pemrograman aplikasi desktop dan web.

Bahasa pemrograman yang didukung oleh Notepad ++ adalah bahasa C ++ karena fungsi-fungsi yang dimasukkan kedalam fungsi dan kata-katanya akan berubah sesuai dengan makna bahasa C++. Kegunaan Notepad ++ ini sangat banyak, yaitu untuk berbagai macam bahasa pemrograman seperti ActionScript, ASP, Assembler, autolt, Batch, C++, C#, COBOL, Fortran, HTML, JavaScript, JSP, Matlab, Perl, Ruby, Python, XML dan masih banyak lagi.

2.6.6. CSS Bootstrap

Bootstrap adalah sebuah library *framework*⁸ CSS yang dibuat khusus untuk bagian pengembangan *front-end*⁹ website. Bootstrap juga merupakan salah satu *framework* HTML, CSS dan javascript yang paling populer di kalangan *web developer* yang digunakan untuk mengembangkan sebuah website yang responsive. Sehingga halaman website nantinya dapat menyesuaikan sesuai dengan ukuran monitor device (desktop, tablet, ponsel) yang digunakan pengguna disaat mengakses website dari browser.

Dengan menggunakan Bootstrap seorang *developer* dapat dengan mudah dan cepat dalam membuat *front-end*

⁸ Kerangka kerja yang berisi berbagai fungsi, plugin, dan konsep

⁹ Tampilan depan/utama

sebuah website, hanya perlu memanggil class-class yang diperlukan, misalnya membuat tombol, grid, tabel, navigasi dan lainnya.

Bootstrap telah menyediakan kumpulan komponen *class interface* dasar yang telah dirancang sedemikian rupa untuk menciptakan sebuah tampilan yang menarik dan ringan. Selain komponen *class interface*, bootstrap juga memiliki grid yang berfungsi untuk mengatur *layout* pada halaman website. Selain itu *developer* juga dapat menambahkan class dan CSS sendiri, sehingga memungkinkan untuk membuat desain yang lebih variatif. Salah satu contoh website yang menggunakan framework bootstrap yaitu twitter. Bootstrap sendiri sebenarnya dikembangkan oleh *developer* twitter sehingga bootstrap sering juga disebut dengan “*twitter bootstrap*”.

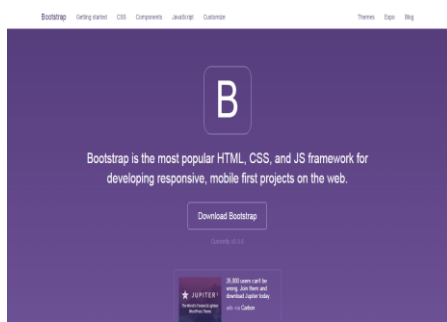
Beberapa alasan mengapa saat ini cukup banyak pengembang yang menggunakan Bootstrap dalam membuat front-end website, yaitu karena beberapa kelebihan yang dimiliki oleh Bootstrap itu sendiri yang antara lain:

- Dapat mempercepat waktu proses pembuatan *front-end website*.
- Tampilan Bootstrap yang sudah cukup terlihat modern.
- Tampilan Bootstrap sudah responsive, sehingga mendukung segala jenis resolusi, baik itu PC, tablet, dan juga smartphone.
- Website menjadi sangat ringan ketika diakses, karena Bootstrap dibuat dengan sangat terstruktur.

Bootstrap dilengkapi dengan beberapa komponen JavaScript dalam bentuk *jQuery plugin*. Mereka menyediakan elemen antarmuka pengguna tambahan seperti kotak dialog, tooltips. Bootstrap juga memperluas fungsi

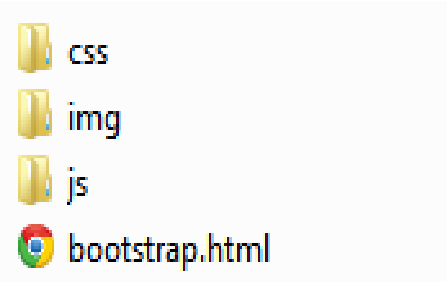
beberapa elemen antarmuka yang ada, termasuk fungsi auto-lengkap untuk bagian masukan. Dalam versi 2.0, plugin JavaScript berikut ini sudah: Modal, Dropdown, Scrollspy, Tab, Tooltip, Popover, Alert, Button, Ciutkan, Carousel dan Typeahead.

Tidak hanya tampilan statis saja yang disediakan Bootstrap, tetapi juga dapat menambahkan tampilan yang dinamis, atau ada sedikit animasinya, sebab di framework ini disediakan juga plugin Javascript. *Framework* ini mendukung web responsive. Jadi, tampilan akan beralih ukurannya tergantung pada resolusi monitor device yang digunakan pengguna pada saat mengakses website dari browsernya.



Gambar II.9 Tampilan Situs Bootstrap

Didalam situsnya, Bootstrap menjelaskan bagaimana cara menggunakan setiap fungsi yang ada seperti membuat tombol, navigasi, formulir, gambar, dan berbagai fungsi HTML lainnya yang lebih responsif dan menarik.



Gambar II.10 Layout file CSS Bootstrap

Berikut isi dari masing-masing direktori/file bootstrap, yaitu:

- Direktori “css” memiliki empat buah file di dalamnya, yaitu:
 1. bootstrap.css
 2. bootstrap.min.css
 3. bootstrap-responsive.css
 4. bootstrap-responsive.min.css
- Direktori “img” memiliki dua buah file di dalamnya, yaitu:
 1. glyphsicons-halflings.png
 2. glyphsicons-halflings-white.png
- Direktori “js” memiliki dua buah file di dalamnya, yaitu:
 1. bootstrap.js
 2. bootstrap.min.js

2.7. Tahapan Pengembangan Aplikasi

Dalam tahap pengembangan aplikasi stok barang berbasis web ini, penulis mencoba menggunakan metode *prototype*¹⁰. Penulis menggunakan metode ini karena memungkinkan pengembangan membuat sebuah model software. Metode ini baik digunakan apabila client tidak bisa memberikan informasi yang maksimal mengenai kebutuhan yang diinginkan. Juga karena penulis beranggapan proses pengembangan yang sangat terstruktur ini membuat potensi kerugian akibat kesalahan pada proses sebelumnya sangat besar dan acap kali mahal karena membengkaknya biaya pengembangan ulang.

2.7.1 Tahapan Metode Prototype

1. Tahap pengumpulan kebutuhan dimana pengguna dan pengembang bersama-sama mendefinisikan format dan kebutuhan keseluruhan perangkat lunak, mengidentifikasi semua kebutuhan, dan garis besar sistem yang akan dibuat.
2. Tahap selanjutnya adalah membangun *prototyping*. Dalam tahap ini,

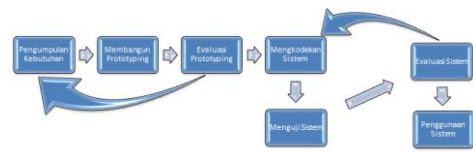
¹⁰Model awal

penulis mencoba membangun prototyping dengan membuat perancangan sementara yang berpusat pada penyajian kepada pengguna (misalnya dengan membuat input dan contoh outputnya).

3. Selanjutnya tahap evaluasi *prototyping*, yaitu evaluasi yang dilakukan oleh pengguna apakah *prototyping* yang dibangun sudah sesuai dengan keinginan pengguna. Jika sudah sesuai maka langkah keempat akan diambil. Jika tidak, maka *prototyping* diperbaiki dengan mengulang langkah 1, 2, dan 3.
4. Tahap selanjutnya adalah pengkodean sistem. Dalam tahap ini *prototyping* yang sudah disepakati diterjemahkan ke dalam bahasa pemrograman yang sesuai. Dalam tahap ini diuji apakah sistem tersebut telah sesuai dengan desain awal atau belum.
5. Tahap berikutnya menguji sistem. Setelah sistem sudah menjadi suatu perangkat lunak yang siap pakai, harus dites dahulu sebelum digunakan. Pengujian ini dilakukan dengan *White Box*, *Black Box*, *Basis Path*, pengujian arsitektur dan lain-lain.
6. Berikutnya evaluasi sistem. Pengguna mengevaluasi apakah sistem yang sudah jadi sesuai dengan yang diharapkannya. Jika sudah, maka langkah ketujuh dilakukan, jika belum maka mengulangi langkah 4 dan 5.
7. Terakhir adalah menggunakan sistem. Perangkat lunak yang telah diuji dan diterima pengguna siap untuk digunakan oleh pengguna.

2.7.2 Tahapan Model Perancangan

Dari tahapan metode *prototype* di atas, maka dapat dilihat model perancangan yang penulis buat adalah sebagai berikut:



Gambar II.11. Model Perancangan

3. ANALISA SISTEM BERJALAN

3.1 Analisa Kebutuhan Sistem

Analisa kebutuhan sistem ini akan memberikan gambaran tentang struktur organisasi perusahaan dengan aplikasi yang berjalan, permasalahan yang ada, serta apa yang dibutuhkan dari sistem tersebut.

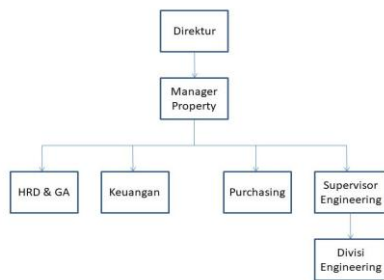
3.1.1 Struktur Organisasi Perusahaan

PT. Mahesa Cipta merupakan perusahaan yang bergerak dibidang *building management & contractor*. Didirikan pada tahun 2012 di daerah SCBD (*Sudirman Centre Business District*) kav.52-53 lot 18 yang berada di jalan Jend. Sudirman Jakarta Pusat. PT. Mahesa Cipta mengelola sebuah gedung perkantoran untuk disewakan kepada perusahaan yang membutuhkan tempat kerja yang nyaman dan berlokasi strategis di tengah ibu kota.

Untuk menunjang kegiatan pekerjaan, maka pihak pengelola selalu menyiapkan berbagai macam peralatan dan bahan baku guna memperbaiki kerusakan yang sewaktu-waktu bisa terjadi, seperti peralatan listrik (kabel, lampu, stopkontak, saklar, dll), suku cadang lift (break switch, door shoe, spring fermator, dll), suku cadang AC (Freon, AC Cleaner, remote, analizer, dll) dan *civil* (semen, compound, mesin las, bor, gergaji besi, sealant, dll).

PT. Mahesa Cipta mempunyai struktur organisasi perusahaan dimana fungsi dan tugas dari setiap bagian berbeda. Struktur organisasi dapat

menggambarkan secara jelas pemisahan kegiatan dan bagaimana aktivitas dan fungsi dibatasi. Di dalam struktur organisasi perusahaan juga harus menjelaskan hubungan antara wewenang siapa yang melapor atau bertanggung jawab kepada siapa jika terjadi suatu pertanggung jawaban tentang apa yang dikerjakan.



Gambar III.1. Struktur Organisasi PT. Mahesa Cipta

Keterangan fungsi dari tiap-tiap bagian adalah sebagai berikut:

1. Direktur

- Menjalankan bisnis perusahaan.
- Menetapkan kebijakan-kebijakan perusahaan.
- Merumuskan dan menetapkan strategi bisnis perusahaan.
- Menyetujui anggaran tahunan perusahaan.
- Memilih, menetapkan, dan mengawasi tugas dari karyawan hingga *manager*.

2. *Manager Property*

- Mengelola dan mengendalikan operasional sebuah gedung
- Merencanakan, mengkoordinasikan, mengontrol pelaksanaan kerja, kegiatan operasional dan perbaikan untuk meningkatkan keamanan, kenyamanan serta kepuasan *tenant/penghuni*.
- Memastikan kegiatan perawatan dan perbaikan (*maintenance and*

repair) atas seluruh aset dan fasilitas gedung dengan metode perawatan dan perbaikan kerja yang benar dan tepat guna.

3. HRD & GA

- Membuat perencanaan mengenai kebutuhan karyawan perusahaan.
- Bertanggung jawab sebagai koordinator seluruh aktivitas perekrutan karyawan.
- Memberikan motivasi kepada karyawan agar dapat menunjukkan kinerja yang optimal.
- Mengelola mutasi dan rotasi karyawan.
- Menyusun program pelatihan karyawan demi memenuhi kebutuhan bisnis perusahaan.
- Bertanggung jawab atas kinerja seluruh karyawan perusahaan.

4. Divisi Keuangan (*Finance*)

- Melakukan pengelolaan keuangan perusahaan.
- Melakukan penginputan semua transaksi keuangan.
- Melakukan transaksi keuangan perusahaan.
- Melakukan pembayaran kepada supplier.
- Melakukan penagihan kepada customer.
- Mengontrol aktivitas keuangan atau transaksi keuangan perusahaan.
- Membuat laporan mengenai aktivitas keuangan perusahaan.
- Melakukan verifikasi terhadap keabsahan dokumen yang diterima.
- Melakukan evaluasi budget.
- Menyiapkan dokumen penagihan invoice atau kuitansi tagihan beserta kelengkapannya.
- Melakukan rekonsiliasi.

5. *Purchasing*

- Mencari dan menganalisa calon supplier yang sesuai dengan material yang dibutuhkan.

- Melakukan negosiasi harga sesuai standar kualitas material dan memastikan tanggal pengiriman material.
 - Melakukan koordinasi dengan pihak supplier mengenai kelengkapan dokumen.
 - Pendukung material sesuai standar mutu yang berlaku.
 - Berkoordinasi dengan PIC dan Gudang tentang jadwal dan jumlah material yang akan diorder.
 - Bersedia melakukan pembelian dilapangan / keluar kantor.
 - Membuat laporan pembelian & pengeluaran barang (inventory, material dll).
 - Melakukan pemilihan/seleksi rekanan pengadaan sesuai kriteria perusahaan.
 - Bekerjasama dengan departemen terkait untuk memastikan kelancaran operasional perusahaan.
 - Memastikan kesedian barang/material melalui mekanisme audit/*control stock*, dll.
6. *Supervisor Engineering*
- Membuat protap–protap (prosedur tetap)
 - Membuat standard operasi dan *maintenance* (perbaikan).
 - Mengatur jadwal dan penugasan pelaksana.
 - Melakukan inspeksi.
 - Memberikan laporan kepada buiding manager secara rutin.
 - Mengatur penggunaan peralatan dan bahan.
7. *Divisi Engineering*
- Melaksanakan pekerjaan pengoperasian, perbaikan dan perawatan alat dan fisik gedung.
 - Memberikan masukan perihal jadwal pemeliharaan rutin dan pemeliharaan perbaikan.
 - Melakukan inspeksi dan pencatatan (*checklist* harian secara rutin).
- Membuat laporan kepada *chief engineering*.

3.1.2 Analisa Sistem Berjalan

Analisis difokuskan pada sistem stok barang yang terdapat di PT. Mahesa Cipta. Berdasarkan metode analisis yang digunakan, maka berikut gambaran sistem yang sedang berjalan pada sistem stok barang di PT. Mahesa Cipta.

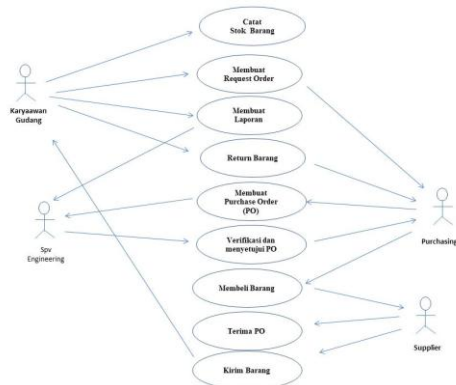
1. Bagian gudang membuat laporan persediaan barang berdasarkan jumlah stok yang tersisa. Laporan persediaan diberikan kebagian *purchasing*.
2. Selanjutnya *purchasing* membuat daftar barang yang akan dibeli dan ditandatangani oleh *spv engineering*.
3. Setelah mendapat persetujuan *spv engineering* untuk melakukan pembelian.
4. Laporan persediaan digunakan untuk bahan acuan sebagai surat permintaan barang, sebelum diserahkan ke supplier maupun toko oleh *purchasing*.
5. Setelah *supplier*/toko mengirimkan barang serta faktur pembelian. Bagian gudang akan mencocokkan terlebih dahulu setelah membuat laporan barang masuk.
6. Laporan barang yang masuk dibuat rangkap dua, rangkap pertama diserahkan kepada *spv engineering*, sedangkan rangkap kedua diarsipkan dibagian *purchasing*.
7. Jika ada barang yang tidak sesuai, bagian gudang membuat retur barang rangkap 2, rangkap pertama diarsipkan dan rangkap kedua diserahkan kepada supplier beserta barangnya.

Tabel III.1. Aktor dalam sistem berjalan

No	Aktor	Keterangan
1	Karyawan gudang	Mencatat persediaan barang dan melaporkannya.
2	Purchasing	Orang yang membeli kebutuhan persediaan barang gudang.
3	Spv Engineering	Kepala teknisi yang mengontrol persediaan barang gudang, juga keluar masuk barang.
No	Aktor	Keterangan
4	Supplier	Mengirim permintaan barang dari customer (<i>Puchasing</i>).

3.1.2.1 Use Case Diagram Sistem Berjalan

Melihat alur kerja permintaan kebutuhan persediaan barang diatas, maka dapat dilihat use case diagram dari peran aktor sistem jalan sebagai berikut:



Gambar III.2 Use Case Diagram Sistem Berjalan

Skenario Use Case Persediaan Barang Sistem Berjalan

Skenario use case adalah narasi tentang aktivitas dalam suatu use case diagram. Adapun skenario use case untuk membuat laporan persediaan barang adalah sebagai berikut:

Nama use case : Laporan persediaan barang

Aktor : Karyawan Gudang

Deskripsi : Meliputi proses pendataan barang gudang, pelaporan sisa barang, dan permintaan penambahan stok barang.

Tabel III.2. Skenario use case laporan persediaan barang

Aktor	Sistem
Karyawan gudang melakukan pengecekan dan pendataan barang gudang.	Karyawan gudang mencocokkan data yang ada dengan stok barang sebenarnya.
	Karyawan gudang membuat laporan sisa barang yang sudah habis.
	Karyawan gudang melaporkan hasil pendataan dan meminta penambahan barang ke bagian <i>purchasing</i> .

Skenario Use Case penambahan barang gudang

Skenario use case merupakan narasi tentang aktivitas dalam suatu use case diagram. Adapun skenario penambahan barang gudang yang berjalan adalah sebagai berikut:

Tabel III.3. Skenario use case Penambahan barang gudang

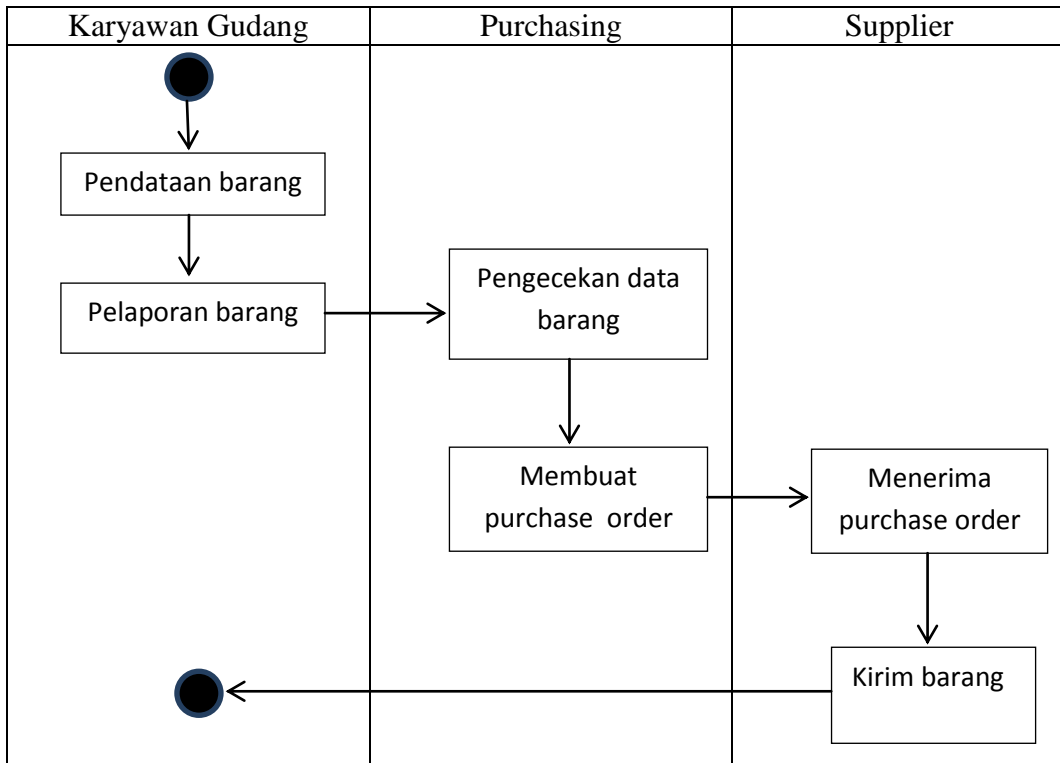
Aktor	Sistem
Bagian <i>purchasing</i>	<i>Spv engineering</i> selaku pengatur penggunaan peralatan dan bahan gudang melakukan pengecekan data bersama <i>purchasing</i> .
	<i>Spv engineering</i> menyetujui pengadaan barang yang habis. <i>Purchasing</i> melakukan pembelian barang.
	Pembelian barang dilakukan melalui telepon, email, maupun datang langsung ke toko.
	Stok barang gudang kembali bertambah.

3.1.2.2 Activity Diagram Laporan Persediaan Barang

Dari analisa yang penulis lakukan di PT. Mahesa Cipta, secara garis besar

bentuk Activity diagram laporan per-
sediaan barang dapat dilihat seperti

berikut:



Gambar III.3 Activity Diagram sistem yang berjalan

Kelebihan dan Kekurangan Sistem Berjalan

Kelebihan Sistem Berjalan

1. Data persediaan barang dapat diberikan secara rinci stok barang apa saja yang masih tersedia maupun yang akan habis dan yang telah habis.
2. Purchasing lebih mudah membelanjakan dana yang ada apabila rincian barang sudah tersedia.

Kekurangan Sistem Berjalan

1. Pengecekan stok barang gudang kurang teliti sehingga terjadi kesalahan pencatatan.
2. Pencatatan barang masih bersifat manual sehingga data yang diberikan rawan hilang dan mudah dimanipulasi.

3. Pengeluaran barang tidak melalui sistem sehingga membingungkan pencatatan stok barang di gudang.

3.1.3 Analisa Kebutuhan Sistem Perusahaan

Dari hasil penelitian yang penulis lakukan terdapat berbagai masalah, sehingga penulis mengusulkan:

1. Sistem terkomputerisasi terkait masalah stok barang gudang guna mendukung kinerja yang lebih efisien dan efektif agar memudahkan pendataan barang.
2. Sistem terkomputerisasi dapat mencegah kehilangan data stok barang gudang.

4. PERANCANGAN DAN IMPLEMENTASI SISTEM

4.1. Perancangan Sistem

Berikut adalah perancangan proses-proses yang penulis usulkan :

1. Perancangan proses permintaan barang
 - a. Administrator bertugas mengelola, mengatur, dan membuat data user dan data barang dalam aplikasi sistem persediaan barang
 - b. Teknisi meminta barang yang diperlukan
 - c. Karyawan gudang melakukan login untuk mengecek stok barang
2. Perancangan proses pengolahan data
 - a. Karyawan gudang bertugas melihat barang dan mencatat keluar masuk barang
 - b. Teknisi bertugas membuat laporan kerja, dan melihat data user
 - c. *Purchase* bertugas membeli dan melihat barang
 - d. Manager dan Supervisor melihat laporan pembelian dan stok barang
 - e. *Supplier* bertugas mengirimkan permintaan barang yang dibutuhkan

4.2. Use Case Diagram Sistem Usulan

Pembuatan *use case diagram* sistem usulan mempunyai beberapa langkah dalam proses penelitiannya, diantaranya:

1. Identifikasi *Actor*
Aktor yang ada dalam sistem usulan, antara lain:

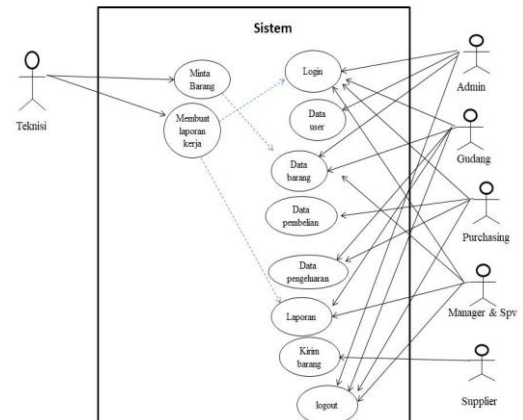
Tabel IV.1. Aktor dalam sistem usulan

No	Aktor	Keterangan
1	Manager/ Supervisor	Orang bisa melihat stok barang dan semua jenis laporan
2	Administrator	Orang yang bertugas membuat dan mengelola aplikasi sistem persediaan barang

3	Purchasing	Orang yang bertanggung jawab dalam proses pembelian barang dan mempunyai akses melihat dan menambah barang
4	Supplier	Orang yang mengirim permintaan barang dari customer (<i>Puchasing</i>).
5	Karyawan Gudang	Orang yang mengurus masalah gudang dan mempunyai hak untuk melihat barang, mencatat keluar masuk barang
6	Teknisi	Orang yang meminta keperluan barang yang akan dipakai dan membuat laporan kerja

2. Identifikasi Use Case

Berikut ini use case yang penulis usulkan :



Gambar IV.1. Use case diagram usulan

Tabel IV.2. Skenario Use case sistem usulan

No	Nama Use Case	Keterangan	Aktor
1	Login	Use case menggambarkan kegiatan login dari semua pengguna aplikasi	Admin, Manager, Spv, Purchasing, Teknisi, Karyawan Gudang
2	Data User	Use case ini menggambarkan pengelolaan	Administrator

		n data user dari sistem	
3	Data barang	Use case ini berisi persediaan stok dan masuk barang yang ada digudang	Administrator, Gudang, Manager, Spv
4	Data Pembelian	Use case ini berisi tentang data kebutuhan barang yang akan dibeli	Purchasing
5	Data pengeluaran	Use case ini berisi tentang pengeluaran barang dari gudang	Karyawan gudang, Purchasing
6	Laporan	Use case ini berisi tentang laporan pekerjaan, laporan pembelian, laporan pengeluaran, dan laporan stok barang	Teknisi, Gudang, Manager, Spv
7	Kirim barang	Use case ini berisi tentang pengiriman barang dari supplier ke pelanggan (dalam hal ini purchasing)	Supplier

1. Skenario Use Case Login

Skenario use case merupakan narasi tentang aktifitas dalam suatu use case diagram. Adapun use case login dalam sistem yang diusulkan adalah sebagai berikut:

Nama Use Case : Login

Aktor : Admin, Manager, Spv, Purchase, teknisi, gudang

Deskripsi : Meliputi proses pengisian username dan password

Tabel IV.3. Skenario Use case Login

Aktor	Sistem
Admin mengisi username dan password.	
Admin memilih daftar admin.	Sistem memproses username dan password di database aplikasi sistem peresediaan barang.
	Jika username dan password benar, sistem akan menampilkan halaman utama, jika salah makan sistem minta admin isi kembali username dan password dengan benar.
Admin mengelola halaman-halaman menu admin aplikasi sistem persediaan barang.	

2. Skenario Use case Data Barang

Skenario use case merupakan narasi tentang aktifitas dalam suatu use case diagram. Adapun skenario use case data barang yang diusulkan adalah sebagai berikut:

Nama Use case : Mengisi data barang

Aktor : Admin dan karyawan gudang

Deskripsi : Meliputi pengolahan data dan stok barang

Tabel IV. 4. Skenario use case data barang

Aktor	Sistem
Karyawan gudang mengisi data barang dan jumlah.	Sistem menyimpan data barang kedalam database aplikasi sistem persediaan barang.
	Jika data yang dimasukkan sudah benar dan lengkap, maka sistem akan menampilkan pesan bahwa

	data berhasil disimpan. Namun jika belum lengkap, maka sistem akan menampilkan pesan bahwa data belum lengkap diisi.
Admin mengecek dan mengelola data barang yang sudah dimasukkan.	

3. Skenario Use case Pembelian

Skenario use case merupakan narasi tentang aktifitas dalam suatu use case diagram. Adapun skenario use case data pembelian yang diusulkan adalah sebagai berikut:

Nama Use Case : Data Pembelian

Aktor : Karyawan gudang, Purchasing, Spv, Supplier

Deskripsi : Meliputi proses pembelian barang

Tabel IV.5. Skenario Use case data pembelian

Aktor	Sistem
Karyawan gudang mengecek sisa stok barang	Sistem menampilkan daftar stok barang yang habis
Karyawan gudang mencatat barang yang harus dibeli	
Karyawan gudang menyerahkan daftar pembelian barang ke purchasing	
Purchasing membuat PO dan memberikan daftar pembelian barang ke Supervisor	
Supervisor menyetujui daftar pembelian barang	
Purchasing mengirim PO ke supplier	
Supplier menerima dan mengirim barang	

yang diminta purchasing	
Purchasing menerima barang dan memberikan ke bagian gudang	
Karyawan gudang mengecek barang dan input stok barang	Sistem akan menyimpan data jumlah stok barang yang sudah di input

4. Skenario Use case Pengeluaran

Skenario use case merupakan narasi tentang aktifitas dalam suatu use case diagram. Adapun skenario use case data pengeluaran yang diusulkan adalah sebagai berikut:

Nama Use Case : Data Pengeluaran

Aktor : Karyawan gudang, Purchasing

Deskripsi : Meliputi proses pengeluaran barang

Tabel IV.6. Skenario Use case data Pengeluaran

Aktor	Sistem
Teknisi meminta keperluan barang kepada bagian gudang	
Karyawan gudang memeriksa stok barang	Sistem menampilkan barang yang diminta beserta jumlahnya
Karyawan gudang mengambil barang yang diperlukan teknisi	Sistem mengurangi jumlah stok barang yang diambil
Karyawan gudang menyerahkan barang yang dibutuhkan teknisi	
Karyawan gudang memberikan informasi tentang barang keluar kepada purchasing	

5. Skenario Use case Laporan

Skenario use case merupakan narasi tentang aktifitas dalam suatu use case diagram. Adapun skenario use case

laporan yang diusulkan adalah sebagai berikut:

Nama Use Case : Laporan

Aktor : Manager, Supervisor

Deskripsi : Meliputi semua jenis laporan

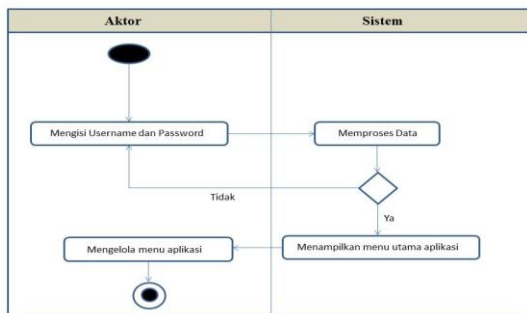
Tabel IV.7. Use case Laporan

Aktor	Sistem
User memilih jenis laporan	Sistem menampilkan laporan yang dipilih
User memilih laporan yang akan ditampilkan/print	Sistem menampilkan laporan beserta keterangan yang dipilih

4.3. Activity Diagram Sistem Usulan

4.3.1. Activity Diagram Login

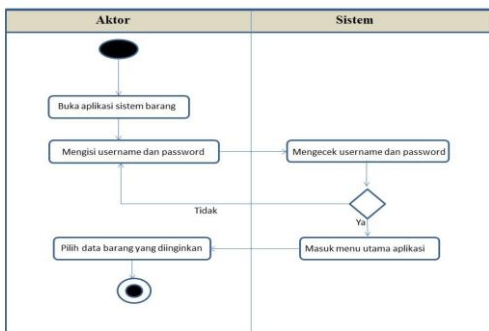
Berikut ini *activity diagram* yang diusulkan :



Gambar IV.2. Activity Diagram Login

4.3.2. Activity Diagram barang

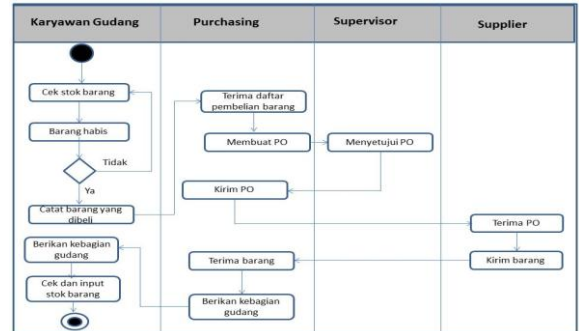
Adapun *activity diagram* data barang yang diusulkan adalah sebagai berikut :



Gambar IV.3. Activity Diagram data barang

4.3.3. Activity Diagram Pembelian

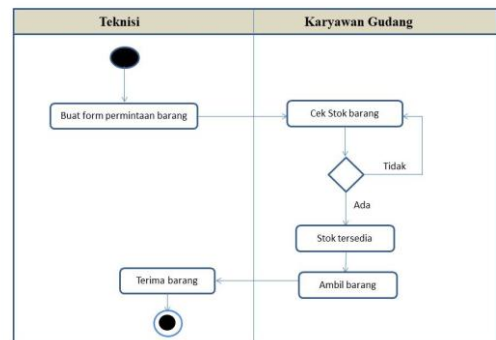
Adapun *activity diagram* data pembelian yang diusulkan adalah sebagai berikut :



Gambar IV.4. Activity Diagram data pembelian

4.3.4. Activity Diagram Pengeluaran

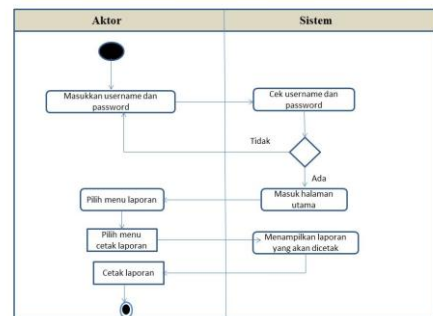
Adapun *activity diagram* data pengeluaran yang diusulkan adalah sebagai berikut :



Gambar IV.5. Activity Diagram data pengeluaran

4.3.5. Activity Diagram Laporan

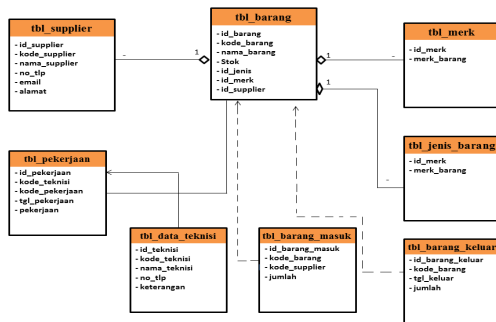
Adapun *activity diagram* laporan yang diusulkan adalah sebagai berikut :



Gambar IV.6. Activity Diagram laporan

4.3.6. Relasi Antar Kelas

Relasi adalah hubungan atau koneksi antar kelas. Kelas-kelas tersebut saling terhubung (bekerja sama) untuk menyelesaikan suatu masalah. Berikut adalah atribut dan relasi antar kelas yang terbentuk :



Gambar IV.7. Relasi Antar Kelas

4.4. Perancangan Basis Data

Spesifikasi basis data yang memberikan penjelasan secara detail tentang masing-masing tabel yang digunakan dalam sistem informasi stok barang adalah sebagai berikut:

1. Tabel barang

Nama File : barang

Primary key : id_barang

Tabel IV.8. Spesifikasi basis data tabel barang

No	Nama Field	Tipe	Ukuran	Keterangan
1	id_barang	integer	11	Primary Key
2	kode_barang	varchar	10	Kode barang
3	nama_barang	varchar	50	Nama Barang
4	Stok	integer	5	Stok barang
5	id_jenis	integer	11	Jenis barang
6	id_merk	integer	11	Merk barang
7	id_supplier	varchar	10	Kode supplier

2. Tabel barang_keluar

Nama File : barang_keluar

Primary key : id_barang_keluar

Tabel IV.9. Spesifikasi basis data tabel barang keluar

No	Nama Field	Tipe	Ukuran	Keterangan
1	id_barang_keluar	integer	11	Primary Key
2	kode_barang	varchar	20	Kode barang
3	tgl_keluar	date		Tanggal barang keluar
4	jumlah	int	11	Jumlah pengeluaran

3. Tabel barang_masuk

Nama File : barang_masuk

Primary key : id_barang_masuk

Tabel IV.10. Spesifikasi basis data tabel barang masuk

No	Nama Field	Tipe	Ukuran	Keterangan
1	id_barang_masuk	integer	11	Primary Key
2	kode_barang	varchar	10	Kode barang
3	kode_supplier	varchar	10	Kode nama supplier
4	jumlah	integer	3	Jumlah barang masuk

4. Tabel supplier

Nama File : supplier

Primary key : id_supplier

Tabel IV.11. Spesifikasi basis data tabel supplier

No	Nama Field	Tipe	Ukuran	Keterangan
1	id_supplier	integer	11	Primary Key
2	kode_supplier	varchar	10	Kode supplier
3	nama_supplier	varchar	50	Nama supplier
4	no_tlp	varchar	50	Telepon supplier
5	email	varchar	15	Email supplier
6	alamat	varchar	50	alamat supplier

5. Tabel teknisi
 Nama File : teknisi
 Primary key : id_teknisi

Tabel IV.12. Spesifikasi basis data teknisi

No	Nama Field	Tipe	Ukuran	Keterangan
1	id_teknisi	integer	10	Primary Key
2	kode_teknisi	varchar	10	Kode teknisi
3	nama_teknisi	varchar	20	Nama teknisi
4	no_tlp	varchar	12	Telepon teknisi
5	keterangan	varchar	100	Keterangan teknisi

6. Tabel user
 Nama File : user
 Primary key : id_user

Tabel IV.13. Spesifikasi basis data user

No	Nama Field	Tipe	Ukuran	Keterangan
1	id_user	integer	11	Primary Key
2	nama_user	varchar	100	Nama user
3	username	varchar	50	Username untuk login
4	password	varchar	50	Password username
5	Level	varchar	20	Level login user

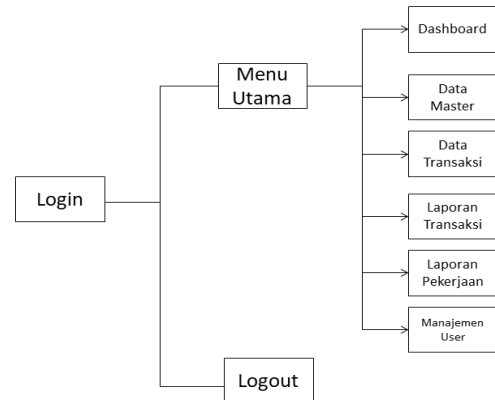
4.5. Perancangan Antar Muka

Perancangan antar muka merupakan mekanisme komunikasi antar pengguna (*user*) dengan sistem. Tujuan utama dari perancangan antar muka adalah agar tampilan bersifat komunikatif dan mudah digunakan oleh *user* sehingga tujuan awal pembuatan dari aplikasi bisa tercapai.

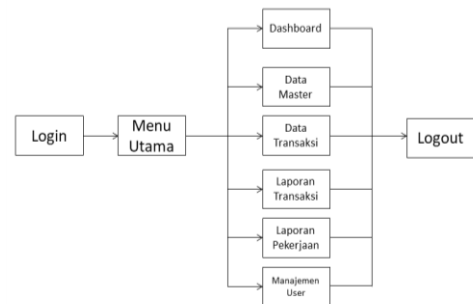
4.5.1 Struktur Menu

Struktur menu merupakan perancangan sistem yang dapat mengintegrasikan sebuah data dalam suatu sistem dengan disertai instruksi-instruksi yang ada pada pilihan sebuah menu. Fungsi-fungsi yang dirancang pada tahap perancangan ini

dibagi ke dalam beberapa menu yang bertujuan untuk memudahkan dalam pengoperasian program. Kategori yang ada pada struktur menu dibedakan berdasarkan hak akses yang dimiliki oleh masing-masing *user*.

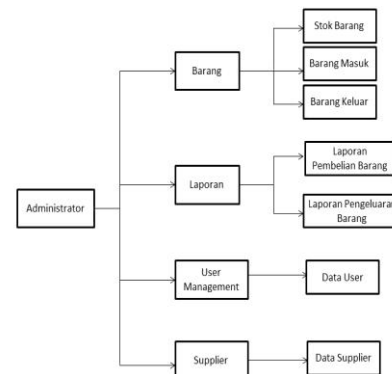


Gambar IV.8. Struktur Menu Utama Login Aplikasi



Gambar IV.9. Struktur Menu Utama Logout Aplikasi

4.5.2. Struktur Menu Administrator



Gambar IV.10. Struktur Menu Administrator

4.6. Perancangan Input

Perancangan input merupakan awal dimulainya proses informasi. Input perlu direncanakan untuk mengkonversikan data mentah ke dalam informasi yang berguna untuk mendapatkan data dalam perancangan. Perancangan tersebut adalah sebagai berikut:

4.6.1. Perancangan Form Login

Tampilan ini merupakan halaman awal saat *user* akan memulai menggunakan program aplikasi. Maka setiap *user* harus mengisi dahulu *form* dari *username* maupun *form password*.

Gambar IV.11. Rancangan form login

4.6.2. Rancangan Menu Utama

Rancangan ini menampilkan seluruh menu utama dari aplikasi yang akan dibuat.

Gambar IV.12. Rancangan Menu Utama

4.7. Implementasi

Implementasi merupakan hasil akhir dari sebuah perancangan yang dibangun berdasarkan data mentah yang telah dikonversikan ke dalam script program untuk mendapatkan hasil *output* berupa desain web beserta informasi didalamnya. Adapun tahapan implementasi adalah sebagai berikut:

1. Tahapan implementasi tampilan
2. Tahapan implementasi penulisan *script*
3. Tahapan pengujian

4.7.1. Tahapan Implementasi Tampilan

4.7.1.1. Tampilan Menu Stok Barang

Menu ini berisi keterangan mengenai stok barang yang ditampilkan kepada *user*. Berikut ini menu stok barang:

Gambar IV.13. Tampilan Menu Stok Barang

4.7.1.2. Tampilan Menu Supplier

Dalam implementasi menu supplier ini merupakan tampilan dimana informasi mengenai data penjual disimpan. Adapun tampilan menu supplier adalah sebagai berikut :

PT. MAHESA CIPTA						
MENU UTAMA		Data Supplier				
Searching	Buat Data Baru		XXX		Cari	
Dashboard	No	Kode Supplier	Nama Supplier	No. tip	Email	Aksi
Data Master >	XXX	XXX	XXX	XXX	XXX	Update Delete
Data Transaksi >						
Laporan Transaksi >						
Laporan Pekerjaan >						
Manajemen User						
Logout						

Gambar IV.14. Implementasi Menu Supplier

4.7.1.3. Tampilan Menu Barang Masuk

Implementasi ini melihatkan bagaimana seorang *user purchasing* memasukkan data stok barang kedalam sistem melalui menu barang masuk.

PT. MAHESA CIPTA				
MENU UTAMA		Data Barang Masuk		
Searching	Buat Data Baru			
Dashboard	No	Kode Transaksi	Tanggal Transaksi	Pilihan
Data Master >	XXX	XXX	XXX	Detail Hapus Cetak
Data Transaksi >				
Laporan Transaksi >				
Laporan Pekerjaan >				
Manajemen User				
Logout				

Gambar IV.15. Tampilan Menu Barang Masuk

4.7.1.4. Tampilan Menu Keluar Barang

Dalam menu ini dijelaskan bagaimana *user gudang* mengambil barang sesuai permintaan dan data tersebut tercatat di menu keluar barang.

PT. MAHESA CIPTA				
MENU UTAMA		Data Barang Keluar		
Searching	Buat Data Baru			
Dashboard	No	Kode Transaksi	Tanggal Transaksi	Pilihan
Data Master >	XXX	XXX	XXX	Detail Hapus Cetak
Data Transaksi >				
Laporan Transaksi >				
Laporan Pekerjaan >				
Manajemen User				
Logout				

Gambar IV.16. Tampilan Menu Keluar Barang

4.7.1.5. Tampilan Laporan Pekerjaan

Implementasi ini merupakan menu untuk *user manager* maupun *supervisor* melihat data laporan.

PT. MAHESA CIPTA						
MENU UTAMA		<< Data Pekerjaan Teknisi				
Searching	Buat Data Baru		Search		Cari	
Dashboard	No	Kode Pekerjaan	Nama Teknisi	Tanggal	Pekerjaan	Aksi
Data Master >						Update Delete Cetak
Data Barang						
Data Transaksi >						
Transaksi Barang Masuk						
Transaksi Barang Keluar						
Laporan Transaksi >						
Laporan Pekerjaan >						
Manajemen User						
Logout						

Gambar IV.17. Tampilan Menu Laporan Pekerjaan

4.7.1.6. Tampilan Menu User

Bagian ini merupakan menu yang dikhususkan untuk *user administrator* membuat, edit, maupun menghapus data *user* yang berhak masuk (*login*) kedalam sistem web informasi stok barang.

PT. MAHESA CIPTA					
MENU UTAMA		<< Data user			
Searching	Buat Data Baru		Search		Cari
Dashboard	No	Nama User	Username	Level	Aksi
Data Master >					Update Delete
Data Barang					
Data Transaksi >					
Transaksi Barang Masuk					
Transaksi Barang Keluar					
Laporan Transaksi >					
Laporan Pekerjaan >					
Manajemen User					
Logout					

Gambar IV.18. Tampilan Menu User

4.7.2. Tahapan implementasi penulisan script

Program ini dibuat menggunakan bahasa pemrograman PHP (*Hypertext Preprocessor*), *XAMPP* sebagai *software* untuk mengelola *database*, dan *sublime text* sebagai teks editor untuk penulisan *script*.

4.7.3. Tahapan implementasi pengujian

Sebelum program siap untuk digunakan, maka program harus benar-

benar bebas dari kesalahan (*error*). Untuk itu perlu dilakukan serangkaian pengujian untuk menemukan kesalahan yang mungkin bisa terjadi saat penulisan *script* program.

Proses pengujian sistem ini dilakukan setelah program melalui serangkaian pengetesan. Dalam proses tersebut

dilakukan pemeriksaan untuk mencari kesalahan dan kelemahan yang mungkin masih terjadi. Proses pengujian aplikasi stok barang meliputi pengujian *black box* dan pengujian *white box* terhadap menu program apakah sudah sesuai dengan apa yang diharapkan atau tidak.

4.7.3.1. Pengujian *Black Box*

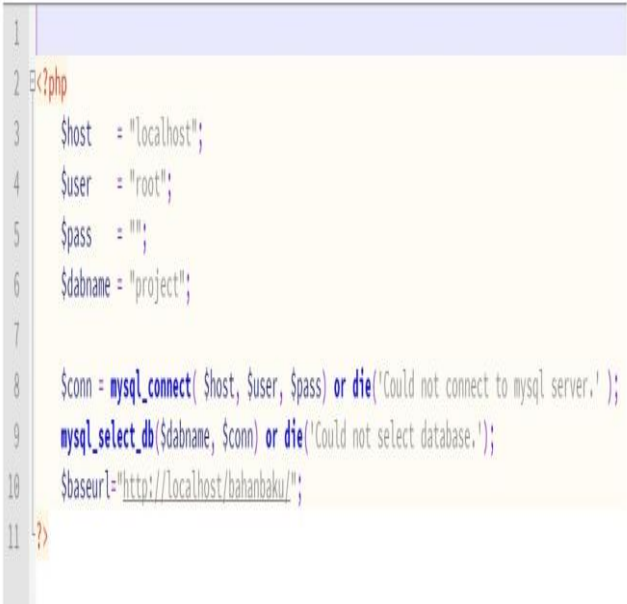
Tabel IV. 14. Hasil Pengujian *Black Box*

No	Rancangan Proses	Hasil yang diharapkan	Hasil
1	Halaman Awal	Tampilan pertama dari aplikasi yang dijalankan	OK
2	Halaman Login	Masuk ke halaman login	OK
3	Menu Utama	Masuk ke halaman menu utama	OK
4	Sub Menu User	Masuk ke halaman data user, tambah, edit, hapus, dan cari user	OK
5	Sub Menu Barang	Masuk ke halaman Stok Barang, tambah, edit, hapus, dan cari barang	OK
6	Sub Menu Supplier	Masuk ke halaman supplier, tambah, edit, hapus, dan cari supplier	Ok
7	Sub Menu Laporan	Masuk ke halaman laporan, buat, edit, hapus, dan cari laporan	Ok
8	Sub Menu Barang Masuk	Masuk ke halaman barang masuk, tambah, edit, hapus, dan cari barang masuk	OK
9	Sub Menu Barang Keluar	Masuk ke halaman barang keluar, tambah, edit, hapus, dan cari barang masuk	OK

4.7.3.2. Pengujian *White Box*

Tabel IV.15. Hasil Pengujian *White Box*

Test Case	Hasil yang diharapkan	Koreksi
<p>1. Memeriksa <i>error and event handling</i> pada saat login</p>  <pre> 1 <script> 2 session_start(); 3 require('db.php'); 4 </script> 5 <doctype html> 6 <html> 7 <head> 8 <meta charset=utf-8> 9 <title>PT. MAHESA CPTA</title> 10 <link rel=stylesheet href=css/style.default.css type=text/css /> 11 <script type=text/javascript src=/js/jquery-1.8.3.min.js /> 12 <script type=text/javascript src=/js/jquery-migrate-1.1.1.min.js /> 13 </head> 14 <body class=loginbody> 15 16 <div class=loginwrapper> 17 <div class=loginwrap index10 animate bounceDown> 18 <div class=loginTitle> PT. MAHESA CPTASilahkan Login</div> 19 <div class=loginwrapperinner> 20 <script include=login.php /> 21 </div> 22 </div> 23 <div class=loginshadow animate fadeUp></div> 24 </div> 25 26 <script type=text/javascript> 27 jQuery.noConflict(); 28 29 jQuery(document).ready(function(){ 30 31 var animate = jQuery.browser.webkit? 'webkitAnimationEnd' : 'animationend'; 32 jQuery('loginwrap').bind(animate,function(){ 33 jQuery(this).removeClass('animate bounceDown'); 34 }); 35 36 jQuery('username,password').focus(function(){ 37 if(jQuery(this).hasClass('error')) jQuery(this).removeClass('error'); 38 }); 39 40 jQuery('loginform button').click(function(){ 41 if(jQuery.browser.msie) { 42 if(jQuery('username').val() == '' jQuery('password').val() == '') { 43 if(jQuery('username').val() == '') jQuery('username').addClass('error'); else jQuery('username').removeClass('error'); 44 if(jQuery('password').val() == '') jQuery('password').addClass('error'); else jQuery('password').removeClass('error'); 45 jQuery('loginwrap').addClass('animate vibrate').bind(animate,function(){ 46 jQuery(this).removeClass('animate vibrate'); 47 }); 48 } else { 49 jQuery('loginwrapper').addClass('animate fadeUp').bind(animate,function){ 50 jQuery('loginform').submit(); 51 }); 52 } 53 return false; 54 } 55 }); 56 </script> 57 </body> 58 </html> </pre>	<ol style="list-style-type: none"> 1. <i>Error handling</i> lengkap 2. Tersedia <i>event handler</i> untuk <i>event handling</i> 	<p>-</p> <p>-</p> <p>-</p>

<p>2. Memeriksa <i>error and handling</i> pada saat koneksi database</p>  <pre> 1 2 <?php 3 \$host = "localhost"; 4 \$user = "root"; 5 \$pass = ""; 6 \$dbname = "project"; 7 8 \$conn = mysql_connect(\$host, \$user, \$pass) or die('Could not connect to mysql server. '); 9 mysql_select_db(\$dbname, \$conn) or die('Could not select database. '); 10 \$baseurl="http://localhost/bahanbaku/"; 11 -?> </pre>	<ol style="list-style-type: none"> 1. <i>Error handling</i> lengkap 2. Tersedia <i>event handler</i> untuk <i>event handling</i> 	
--	--	--

5. PENUTUP

5.1. Kesimpulan

Setelah semua tahap penelitian dilakukan, maka penulis mengambil kesimpulan sebagai berikut :

1. Penelitian ini mampu merancang sebuah sistem informasi stok barang untuk membantu penanganan stok barang di PT. Mahesa Cipta.
2. Sistem aplikasi ini mengelola stok barang dengan menggunakan sistem informasi berbasis *web*.
3. Sistem aplikasi ini menghasilkan laporan keluar masuk barang berdasarkan tanggal, kode barang, maupun nama barang.
4. Sistem aplikasi ini mampu mempermudah untuk mengetahui

informasi mengenai stok barang yang tersedia maupun yang sudah habis.

5. Sistem aplikasi ini dapat mempermudah pencarian barang.

5.2. Saran

Dalam upaya untuk pengembangan sistem aplikasi ini, ada beberapa hal yang penulis sarankan :

1. Sistem aplikasi ini bisa diberikan menu tambahan sesuai kebutuhan perusahaan.
2. Menambah fasilitas *help* agar sistem aplikasi ini lebih *user friendly*.
3. Pengembangan *interface* sistem agar tampilan lebih menarik.

DAFTAR PUSTAKA

- Arief M. Rudianto. 2011. *Pemrograman Web Dinamis menggunakan PHP dan MySQL*. Yogyakarta: Andi.
- Arifianto, Rahmat. 2014. Pengertian Flowchart. Diakses tanggal 2 Februari 2017, dari <https://rahmatarifianto.wordpress.com/2014/11/20/pengertian-flowchart-dan-jenis-jenisnya/>
- Asnawi, Choerun. 2012. *Modul Web Dasar*. Yogyakarta: Andi.
- Fatansyah. 2007. *Basis Data*. Bandung : informatika.
- Hidayatullah, Priyanto. 2014. *Pemrograman Web*. Bandung: Informatika. 2015
- <https://www.youtube.com/channel/UCkXmLjEr95LVtGuIm3l2dPg> diakses tanggal 15 Mei 2018
- Junaedi, Ahmad. 2008. Pengertian dan fungsi PHP. Diakses tanggal 5 Februari 2017 dari <http://www.duniaikom.com/pengertian-dan-fungsi-php-dalam-pemograman-web/>
- Kertahadi. 2007. *Dasar-dasar Sistem Informasi*. Yogyakarta: STIKOM
- M Anastasia Meilan. 2013. *Trik Menguasai PHP + JQUERY berbasis Linux & Windows*. Yogyakarta: Lokomedia. 2013
- McLeod. 2010. *Management Information System*, Jakarta : Salemba Empat
- Puspita, Ade. 2013, Pengertian DFD. Diakses tanggal 2 Febuari 2017 dari <https://adepuspita28.wordpress.com/2013/10/26/pengertian-data-flow-diagram-dfd-dan-contoh-gambar-dfd/>
- Pratama, Dimas. 2010, Karakteristik sistem. Diakses tanggal 5 Februari 2017 dari <https://dimas347.wordpress.com/2010/12/08/karakteristik-sistem/>
- Setyadi. 2010, Konsep Dasar Sistem. Diakses tanggal 5 Februari 2017 dari <http://setya21.blogspot.co.id/2010/10/konsep-dasar-sistem.html>