

## **Bab 5**

### **Analisis**

#### **5.1. Basis Data Sistem Inventori *Spare Part***

Menganalisis sistem basis data ini dapat memberikan pemahaman mengenai keuntungannya, hambatan yang mungkin muncul, dan metode optimal yang bisa digunakan. Sistem tersebut menghasilkan keuntungan yang substansial, seperti mencatat informasi yang akurat mengenai semua suku cadang yang tersedia, memonitor tingkat persediaan secara langsung, dan memiliki kapabilitas untuk merencanakan pemeliharaan serta perbaikan dengan lebih optimal.

Dalam sistem inventori *spare part* pada PT. Lucas Djaja dilakukan analisis basis data menggunakan metode FAST (*Framework for the Application of System Thinking*). Metode FAST memiliki beberapa tahapan yaitu:

Pada definisi lingkup (*scope definition*) melibatkan elemen-elemen yang akan diintegrasikan dalam sistem, sekaligus juga yang tidak akan dimasukkan. Selama tahapan ini, alat yang digunakan adalah tabel pernyataan masalah yang bisa dilihat pada tabel 4.1. Tabel ini memiliki peran dalam mengenali, mengelompokkan, dan menentukan urutan prioritas masalah-masalah tersebut, sehingga dapat meningkatkan pemahaman tentang situasi yang dihadapi dan memungkinkan perencanaan tindakan yang diperlukan. Dalam tabel ini, informasi tentang urgensi, tingkat keterlihatan, prioritas, dan usulan solusi biasanya tertera. Dalam konteks ini, terdapat 3 pernyataan singkat masalah yang terjadi yaitu kurang sesuainya data barang di gudang, waktu pembuatan laporan persediaan yang memakan waktu, sulit mencari data barang. Tiga pernyataan masalah tersebut memiliki tingkat urgensi yang sama yaitu segera dikarenakan bila masalah tersebut tidak segera di pecahkan maka akan mempengaruhi kinerja pada sistem kerja yang ada di PT. Lucas Djaja. Tingkat prioritas pada tiga masalah tersebut secara data memiliki prioritas yang sama karena dalam tingkatan urgensi memiliki urgensi yang sama yaitu segera. Jadi dalam hal tersebut tingkatan prioritas yang pertama yaitu kurang sesuainya data

barang di gudang, karena jika data barang di dalam gudang masih kurang sesuai pembuatan laporan dan pengecekan barang tidak akan benar. Tingkat prioritas yang kedua yaitu waktu pembuatan laporan persediaan yang memakan waktu, karena dalam pencarian data harus memiliki data persediaan dalam gudang, dengan terlambatnya pembuatan laporan data persediaan akan menyebabkan pencarian barang terkendala terlambat. Tingkatan prioritas yang ketiga yaitu sulit mencari data barang.

Pada tahap analisis permasalahan (*Problem analysis*) dilakukan dengan memeriksa secara rinci sistem informasi inventori di gudang di PT Lucas Djaja dalam kondisi saat ini. Dalam fase ini, tabel matriks digunakan sebagai alat bantu yang mencakup masalah-masalah yang ada, peluang-peluang yang mungkin, serta keterbatasan-keterbatasan yang ditemui, yang bisa dilihat pada tabel 4.2. Selain itu, use case dan flowmap juga digunakan sebagai alat bantu untuk mendapatkan pandangan lebih jelas tentang bagaimana sistem berinteraksi dan berfungsi dalam konteks yang berbeda, yang bisa dilihat pada gambar 4.2 dan gambar 4.3.

Tabel matriks yang berisi masalah, peluang, dan keterbatasan disusun dengan tujuan untuk menetapkan tujuan perbaikan dalam sistem. Dalam tabel matriks ini, terdapat analisis sebab-akibat serta tujuan-tujuan yang ingin dicapai dalam perbaikan sistem. Dalam analisis sebab-akibat teridentifikasi tiga masalah berupa, urang sesuainya data barang di gudang, keterlambatan dalam proses pembuatan laporan persediaan, dan Sulit mencari data barang. Dari masalah tersebut, ditemukan sebab akibatnya, yaitu kurang telitinya pekerja dalam melakukan pendataan barang di gudang, kesulitan dalam menemukan data yang diperlukan sehingga mengakibatkan lamanya waktu yang dibutuhkan untuk pembuatan laporan, dan mencari data inventori dilakukan secara manual, sehingga memakan waktu. Melalui analisis sebab-akibat tersebut, dihasilkan solusi tujuan untuk mengatasi masalah tersebut, yakni meningkatkan keakuratan pengolahan data, mempercepat proses pembuatan laporan persediaan, Memudahkan dalam mencari data.

*Use case diagram* memberikan gambaran visual yang mudah dimengerti tentang fungsionalitas sistem dan interaksi yang terjadi antara pengguna dan sistem tersebut. *Use case* pada sistem inventori gudang terdapat 4 aktor yang bisa dilihat pada gambar 4.2. Aktor yang terlibat adalah karyawan, teknisi, pemasok, dan direktur. Aktor "karyawan" memiliki tanggung jawab untuk melaksanakan beberapa tugas, seperti membuka aplikasi, login, membuat laporan, serta mengelola pergerakan barang masuk dan keluar. Sementara itu, aktor "teknisi" memiliki fokus pada entitas "barang keluar," yang merupakan bagian dari tanggung jawab yang juga ada pada aktor "karyawan." Pemasok, sebagai aktor, terlibat dalam tindakan menerima permintaan barang serta memberikan pasokan barang yang diminta. Di sisi lain, aktor "direktur" memiliki peran yang lebih terbatas, yaitu menerima laporan mengenai barang.

Hasil flowmap dari sistem inventori gudang di PT. Lucas Djaja yang bisa dilihat pada gambar 4.3, terlihat bahwa tindakan yang diambil oleh admin gudang adalah sebagai berikut: pertama, admin gudang membuka aplikasi dan melakukan proses login. Kemudian, admin gudang memasukkan data barang masuk yang diterima dari pemasok ke dalam database aplikasi. Data barang masuk ini kemudian dikumpulkan dan diolah untuk dibentuk menjadi sebuah laporan. Selanjutnya, admin gudang menginput pengeluaran barang dari stok sesuai dengan barang yang diambil. Setelah itu, admin gudang membuat laporan yang mencakup informasi tentang penerimaan barang, permintaan barang, dan stok barang yang ada. Laporan tersebut disiapkan untuk diserahkan kepada Manajer.

Dalam tahap analisis persyaratan (*Requirement Analysis*) terdapat dua alat yang digunakan, yaitu persyaratan fungsional dan persyaratan non-fungsional. Hasil dari pengembangan sistem inventori gudang menghasilkan persyaratan fungsional yang dapat ditemukan pada Tabel 4.3. Dalam persyaratan fungsional ini, dijelaskan aktivitas-aktivitas yang harus dilakukan oleh para aktor secara spesifik yaitu dalam aktor karyawan aktivitas fungsional menginput data barang masuk dan keluar dan membuat laporan, untuk aktor direktur aktivitas fungsional yaitu menerima laporan,

dan untuk aktor pemasok yaitu memasok barang yang dipesan. Di sisi lain, kebutuhan non-fungsional diidentifikasi dengan menggunakan kerangka PIECES, yaitu *Performance, Information, Economics, Control, Efficiency, dan Services*. Dalam konteks sistem inventori gudang, *Performance* mengacu pada kualitas kinerja yang direncanakan dalam sistem inventori, yang didasarkan pada kebutuhan sistem yang diinginkan. Di sisi lain, *Information* merujuk pada informasi yang mampu mendukung sistem informasi inventori dalam mengakses inventori dengan detail dan kecepatan yang tinggi, *Economics* merujuk pada penggunaan yang efisien dari pembangunan sistem inventori, yang dapat membawa penghematan dalam hal material dan waktu. *Control* mengacu pada kemampuan sistem inventori untuk mengontrol data inventori, yang pada gilirannya meningkatkan keamanan data. *Efficiency* menunjukkan bahwa melalui sistem inventori, semua aktivitas terkait inventori akan menjadi lebih mudah dikontrol oleh pengguna. *Services* menggambarkan bagaimana sistem inventori dapat meminimalisir kesalahan dan meningkatkan efisiensi dalam pengelolaan inventori, sehingga pelayanan terhadap pengguna menjadi lebih baik.

Tahapan Desain logis (*Logical Design*) melibatkan perencanaan komponen dan interaksi dalam suatu cara yang masuk akal, padu, serta memenuhi keperluan yang sudah diidentifikasi sebelumnya. Dalam tahap desain logis pada sistem inventori, alat yang digunakan meliputi model data konseptual, model data logis, *Entity Relationship Diagram, diagram konteks, diagram dekomposisi, serta data flow diagram*.

*Conceptual data model* dibuat menggunakan aplikasi *Power Designer*. Dalam sistem basis data inventori, terdapat beberapa tabel seperti tabel karyawan, barang keluar, barang, barang masuk, dan pemasok. Setiap tabel memiliki atribut yang mengacu pada karakteristik atau informasi yang terkait dengan entitas dalam tabel. Di setiap tabel, ada satu atribut yang berperan sebagai kunci utama atau *primary key*. Dengan contoh didalam CDM yang bisa dilihat pada gambar 4.4 terdapat di dalam tabel karyawan memiliki beberapa atribut IDkaryawan sebagai *primary key*

yang mempunyai ciri khas dalam tabel memiliki *symbol* pagar, nama dan jabatan karyawan sebagai atribut pendukung.

*Logical data model* merupakan kelanjutan dari keterhubungan antara tabel-tabel yang ada dalam *conceptual data model* yang bisa dilihat pada gambar 4.5. Perbedaan utama antara *logical data model* dan *conceptual data model* adalah bahwa dalam *logical data model* terdapat penggunaan *foreign key*. Dalam setiap tabel, terdapat sejumlah *Foreign key* yang menjalankan fungsi ini. Sebagai contoh, dalam tabel barang keluar, terdapat penggunaan *Foreign key* yang berasal dari tabel karyawan. Selain itu, pada tabel barang, *Foreign key* digunakan untuk merujuk pada tabel barang keluar dan barang masuk. Selanjutnya, dalam tabel barang masuk, terdapat *Foreign key* yang menghubungkannya dengan tabel pemasok.

*Entity Relationship Diagram* dalam sistem inventori ada beberapa entitas yang bisa dilihat pada gambar 4.6. Didalam ERD terdapat beberapa entitas yaitu karyawan, barang keluar, barang, barang masuk, dan pemasok. Entitas tersebut dihubungkan dengan relasi. Dalam relasi ini terdapat hubungan antara entitas contohnya untuk hubungan entitas karyawan dan barang keluar yaitu mengambil 1 ke N, 1 ke N tersebut dapat diartikan sebagai satu karyawan dapat mengambil banyak barang. Secara keseluruhan ERD menjelaskan hubungan antara tabel atau entitas yang digunakan sebagai acuan pembuatan database sistem inventori.

*Diagram konteks* dalam sistem inventori yang bisa dilihat pada gambar 4.7, terdapat beberapa entitas yaitu karyawan, pemasok, dan direktur. Terlihat juga *output* dan *input* berupa *input* dari karyawan dan *output* laporan barang. Semua entitas tersebut berinteraksi kedalam sistem inventori. Sebagai contoh entitas karyawan berinteraksi dengan sistem dengan memasukan data stok barang, data barang dan permintaan barang, sedangkan sistem memberi data informasi stok barang kepada entitas karyawan.

*Diagram dekomposisi* memperlihatkan bagaimana aliran data dan proses berinteraksi dengan komponen-komponen yang lebih spesifik. Didalam *diagram dekomposisi* pada sistem inventori yang bisa dilihat pada gambar 4.8, terdapat sistem yang dipecah menjadi beberapa rincian subsistem diantaranya yaitu mengelola data barang, mengelola permintaan barang, mengelola barang masuk, mengelola barang keluar dan mengelola laporan barang. Diagram dekomposisi ini digunakan untuk membuat diagram sistem, yang bisa dilihat pada gambar 4.9. Pada diagram sistem ini menjelaskan informasi sistem inventori yang sedang berlangsung. Pada diagram tersebut karyawan menginput data barang ke dalam sistem mengelola data barang, lalu didalam sistem database akan merekam data barang yang telah di input. Jika barang yang sudah diinput kedalam sistem kurang dan perlu memesan, karyawan akan melakukan permintaan barang melalui sistem mengelola permintaan barang dan diberikan kepada pemasok. Setelah pemasok memasok barang selanjutnya data barang yang dipasok dimasukkan kedalam sistem mengelola barang masuk untuk mengupdate data stok barang yang ada di dalam sistem. Selanjutnya semua data barang yang dilakukan di dalam sistem dibuat laporan dengan database sistem mengelola laporan untuk diberikan kepada direktur. Diagram sistem ini dibuat untuk acuan membuat database sistem inventori.

Tahapan analisis keputusan (*Decission Analysis*) yang bisa dilihat pada tabel 4.4. Didalam tabel tersebut terdapat dua kandidat untuk pengembangan sistem inventori. Dalam kandidat tersebut terdapat perbedaan dalam segi perangkat lunak, *framework* aplikasi, server, komputerisasi. Pada dua kandidat tersebut hanya terpilih satu kandidat untuk pengembangan sistem inventori yaitu kandidat kedua dengan digunakan perangkat lunak *xampp*, *php mysql* dan *framework* aplikasi menggunakan *CodeIgniter*. Dengan terpilihnya kandidat kedua memiliki keuntungan lebih fleksibel untuk pembuatan aplikasi, memiliki kontrol sesuai yang dibutuhkan pengguna dan juga dengan menggunakan kerangka aplikasi *CodeIgniter* akan mempermudah membuat aplikasi karena dengan kerangka tersebut sudah sesuai dengan aplikasi yang diinginkan pengguna.

## 5.2. Rancangan Sistem Inventori

Dalam tahapan ini terdapat beberapa *tools* yang digunakan yaitu *physical data model*, perancangan struktur database serta rancangan sistem. *Physical data model* adalah representasi konkrit dari bagaimana data akan diorganisasi dan disimpan dalam sistem basis data atau penyimpanan data fisik. *physical data model* didapatkan dari logical data model lalu ditransformasikan menjadi model data secara fisik yang bisa dilihat pada gambar 5.1. Di dalam PDM memiliki lima tabel secara fisik yaitu tabel karyawan, barang keluar, barang, barang masuk, pemasok. Tabel tersebut selanjutnya akan digunakan didalam sistem inventori yang dimasukan didalam sistem *php mysql* menjadi tabel induk.

Tahapan selanjutnya yaitu *prototype* sistem yang dibuat dengan *php mysql* dan menggunakan *framework CodeIgniter* tentang sistem inventori. Dengan menggunakan *CodeIgniter* sistem akan lebih tertata karena dengan kerangka tersebut sistem sudah dibuat dengan baik oleh pembuat yang lebih ahli pada bidangnya. Tetapi kerangka *CodeIgniter* kurang sempurna dengan sistem yang sedang diteliti, dengan itu sistem harus disempurnakan dengan menambahkan/mengurangi fitur dalam sistem.

Pada sistem yang dibuat terdapat beberapa tampilan halaman yaitu tampilan *dashboard*, *supplier*, barang, barang masuk, barang keluar, cetak laporan. Didalam tampilan beranda terdapat total data barang yang ada pada sistem basisdata, total stok barang yang ada pada sistem, total nilai transaksi barang perbulan yang dipresentasikan dengan grafik batang, total transaksi barang masuk dan keluar yang dipresentasikan dengan diagram lingkaran, stok barang minimum yang jika barang kurang dari *safety stock* akan tampil pada kolom stok barang dan bisa langsung membuat pesanan barang dengan mengklik kolom tambah pasok, lima transaksi barang masuk dan barang keluar.

Tampilan data master yang isinya tampilan form pemasok dan barang. Pada form pemasok terdapat data pemasok dan tambah data pemasok. Pada form barang

terdapat beberapa form yaitu form satuan barang pada form satuan barang ini adalah nama satuan barang yang ditambah kedalam sistem, satuan barang juga bisa ditambah jika ada satuan barang baru. Form jenis barang pada form ini yaitu jenis barang yang dimasukkan kedalam sistem. Selanjutnya ada form data barang didalam form ini berisikan data barang yang ada didalam sistem, didalam form juga terdapat fitur tambah barang jika ada barang baru yang di input kedalam sistem.

Pada tampilan transaksi terdapat form barang masuk dan keluar, pada form barang masuk terdapat riwayat data barang masuk dan didalamnya terdapat fitur untuk menginput barang masuk yang baru. Pada form barang keluar terdapat riwayat data barang masuk dan didalamnya terdapat fitur untuk menginput barang keluar yang baru.

Pada tampilan report terdapat form cetak laporan yang isi didalamnya terdapat pilihan untuk menampilkan laporan barang masuk, barang keluar, laporan barang, dan juga grafik total transaksi. Dalam menampilkan laporan terdapat fitur untuk memilih dari tanggal laporan yang akan pengguna tampilkan agar lebih fleksibel.