

BAB II

TINJAUAN PUSTAKA

2.1. Landasan Teori

2.1.1. Analisis dan Perancangan

Analisis atau yang lebih dikenal dengan analisis sistem adalah pembelajaran sebuah sistem dan komponen-komponennya sebagai prasyarat desain sistem untuk sebuah sistem yang baru dan diperbaiki. Analisis sistem merupakan sebuah teknik pemecahan masalah yang menguraikan sebuah sistem menjadi bagian komponen dengan tujuan mempelajari seberapa bagus bagian-bagian dalam komponen tersebut bekerja dan berinteraksi untuk meraih tujuan mereka. (Whitten, 2004 : 176)

Menurut Whitten (2004 : 176) Perancangan didefinisikan sebagai tugas yang fokus pada spesifikasi solusi detail berbasis komputer. Terdapat beberapa strategi perancangan desain sistem, yaitu :

- a. Desain Struktur Modern
- b. Teknik Informasi
- c. *Prototyping*

- d. *Join Application Development (JAD)*
- e. *Rapid Application Development (RAD)*
- f. Desain Berorientasi Objek

Kadang – kadang teknik tersebut dianggap sebagai teknik yang saling bersaing, tetapi seringkali untuk beberapa jenis proyek tertentu diperlukan kombinasi dari beberapa diantaranya sehingga saling melengkapi satu sama lain.

2.1.2. Data dan Basis Data

Data adalah fakta-fakta tentang segala sesuatu di dunia nyata yang dapat direkam dan disimpan pada media komputer, sedangkan basis data adalah kumpulan yang terorganisasi dari data-data yang berhubungan sedemikian rupa sehingga mudah disimpan, dimanipulasi serta dipanggil oleh pengguna. (Nugroho, 2011 : 5).

Pendapat lain menyatakan bahwa basis data adalah koleksi dari data-data yang terorganisasi dengan cara sedemikian rupa sehingga mudah dalam disimpan dan dimanipulasi (diperbaharui, dicari, diolah dengan perhitungan-perhitungan tertentu, serta dihapus). (Nugroho, 2004 : 41).

Dalam penelitian ini penulis menggunakan basis data *MySQL* dikarenakan basis data *MySQL* merupakan basis data yang bersifat *Open Source* artinya siapa saja boleh menggunakannya dan bersifat *legal*.

2.1.3. Manajemen Produksi

Pengertian Produksi yaitu suatu kegiatan atau proses yang mentransformasikan masukan (*input*) menjadi keluaran (*output*). Dalam arti sempit produksi adalah kegiatan yang menghasilkan barang baik barang setengah

jadi, barang jadi, barang industri, suku cadang, komponen penunjang. Sedangkan pengertian manajemen produksi mencakup 3 (tiga) unsur penting yaitu adanya orang yang lebih dari satu, adanya tujuan yang ingin dicapai, dan orang yang bertanggung jawab terhadap pencapaian tujuan tersebut. (<http://ocw.gunadarma.ac.id/course/economics/management-s1/pengantar-bisnis/manajemen-produksi>)

2.1.4. Sistem

Menurut Kristanto (2003 : 1), sistem merupakan kumpulan elemen-elemen yang saling terkait dan bekerjasama untuk memproses masukan (*input*) yang ditujukan kepada sistem tersebut dan mengolah masukan tersebut sampai menghasilkan keluaran yang diinginkan. Elemen - elemen dalam sistem meliputi :

- a. **Tujuan Sistem**, merupakan tujuan dari sistem tersebut dibuat. Tujuan sistem dapat berupa tujuan organisasi, kebutuhan organisasi, permasalahan yang ada dalam suatu organisasi maupun urutan prosedur untuk mencapai tujuan organisasi.
- b. **Batasan Sistem**, merupakan sesuatu yang membatasi sistem dalam mencapai tujuan sistem. Batasan sistem dapat berupa peraturan-peraturan yang ada dalam suatu organisasi, biaya-biaya yang dikeluarkan, orang-orang yang ada dalam organisasi, fasilitas baik itu sarana dan prasarana maupun batasan yang lain.
- c. **Kontrol Sistem**, merupakan pengawasan terhadap pelaksanaan pencapaian tujuan dari sistem tersebut. Kontrol sistem dapat berupa kontrol terhadap pemasukan data (*input*), kontrol terhadap keluaran data (*output*), kontrol terhadap pengolahan data, kontrol terhadap umpan balik dan sebagainya.

- d. **Input**, merupakan elemen dari sistem yang bertugas untuk menerima seluruh masukan data, dimana masukan tersebut dapat berupa jenis data, frekuensi pemasukan data dan sebagainya.
- e. **Proses**, merupakan elemen dari sistem yang bertugas untuk mengolah atau memproses seluruh masukan data menjadi suatu informasi yang lebih berguna. Misalkan sistem produksi akan mengolah bahan baku yang berupa bahan mentah menjadi bahan jadi yang siap untuk digunakan.
- f. **Output**, merupakan hasil dari input yang telah diproses oleh bagian pengolah dan merupakan tujuan akhir sistem. *Output* ini bisa berupa laporan grafik, diagram batang dan sebagainya.

2.1.5. Perancangan Basis Data

Perancangan basis data, menurut Nugroho (2011 : 43), Perancangan basis data dibagi menjadi 2 yaitu :

1. Perancangan basis data secara logika, yaitu melakukan pendekatan basis data melalui dua cara pandang yaitu menerjemahkan atau mentransformasikan model data konseptual ke bentuk relasi-relasi berdasarkan teori basis data relasional dan teori objek.
2. Perancangan basis data secara fisik, yaitu mengorganisasikan basis data ditempat penyimpanan komputer serta mendefinisikan struktur fisik dari *DBMS*.

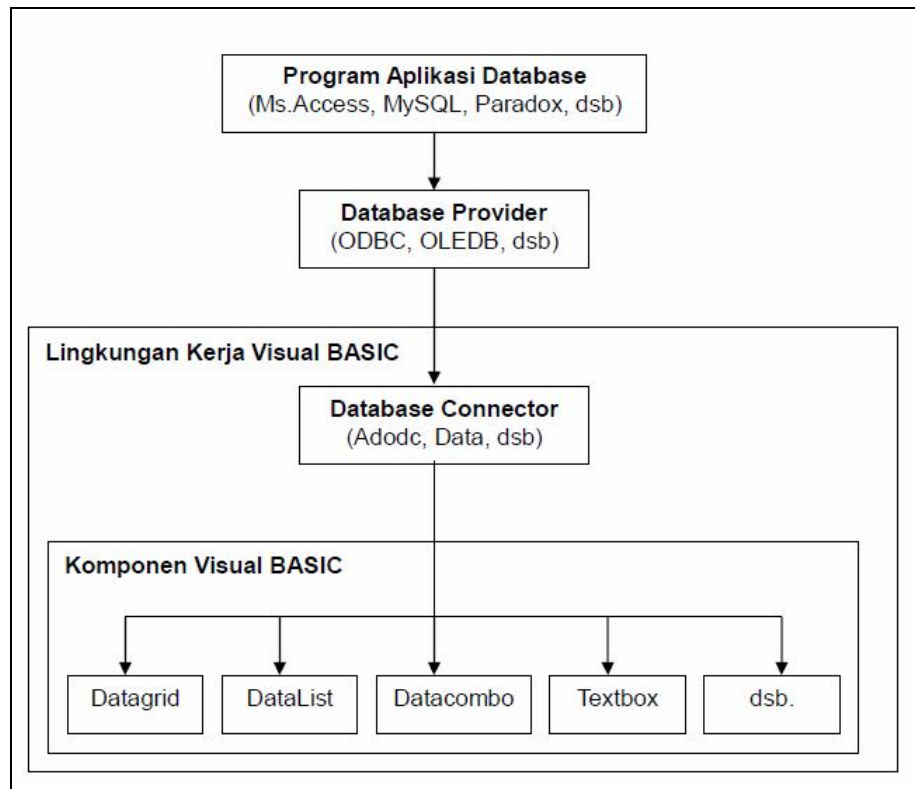
2.1.6. Sistem Basis Data

Sistem basis data menyediakan bahasa untuk mendefinisikan basis data *DDL (Data Definition Language)* serta bahasa untuk memanipulasi basis data *DML (Data Manipulation language)* untuk melakukan operasi tertentu dalam basis data. (Nugroho, 2011 : 16-17).

1. *DDL (Data Definition Language)*, skema basis data dengan sekumpulan definisi yang diekspresikan dengan bahasa khusus yang dinamakan *Data Definition Language*.
2. *DML (Data Manipulation Language)*, adalah bahasa yang memungkinkan pengguna untuk mengakses atau memanipulasi data dalam sistem basis data yang bertipe relasional.

2.1.7. MySQL Dan Visual Basic

Pemrograman Basis Data (*Database Programming*) merupakan suatu bentuk pemrograman alternatif untuk pengolahan database. Dengan pemrograman basis data didapat secara leluasa mengatur tampilan dan alur kerja sebuah basis data dengan lebih baik. *Visual BASIC* merupakan salah satu bahasa pemrograman yang telah mendukung pemrograman basis data. *Visual BASIC* dapat dihubungkan dengan program aplikasi pengolah data lain seperti *Access, MySQL, SQL Server* dan sebagainya.



Gambar 2.1. Alur kerja Pemrograman Basis Data dengan *Visual Basic*

Untuk dapat menghubungkan *Visual BASIC* dengan *database MySQL*, digunakan komponen *ADO Data Control (ADODC)*. Komponen ini dapat dihubungkan dengan beberapa komponen yang digunakan untuk mengakses data seperti *textbox*, *datagrid*, dan sebagainya. Untuk dapat menggunakan ADODC, ada beberapa properti yang harus kita lakukan pengaturan.

Tabel 2.1. Proporeti ADODC dalam *Visual Basic*

Properti	Fungsi
ConnectionString	Untuk mendefinisikan database yang akan digunakan Adodc.
RecordSource	Untuk mendefinisikan tabel/query yang akan digunakan Adodc.
CommandType	Untuk mendefinisikan perintah yang digunakan Adodc.

2.1.8. *UML (Unified Modelling Language)*

Menurut Munawar (2005 : 17) *UML (Unified Modelling Language)* adalah alat bantu yang menyediakan bahasa pemodelan visual yang memungkinkan bagi pengembang sistem untuk membuat cetak biru atas visi mereka dalam bentuk yang baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi (*sharing*) dan mengkomunikasikan rancangan mereka dengan yang lain.

UML adalah sebuah “bahasa” yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. *UML*, menawarkan sebuah standar untuk merancang model sebuah sistem.

a. Tujuan *UML*

Adapun tujuan dari *UML* adalah sebagai berikut :

1. Memberikan model yang siap pakai, bahasa permodelan visual yang *ekspresif* untuk mengembangkan model dan dimengerti secara umum.
2. Memberikan bahasa permodelan yang bebas dari berbagai bahasa pemrograman dan proses rekayasa.
3. Menyatukan praktek-praktek yang terdapat dalam permodelan.

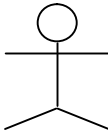
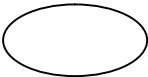
b. Diagram-diagram dalam *UML*

Ada beberapa diagram dalam *UML (Unified Modelling Language)* antara lain :

1. *Use Case Diagram*, menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case* merepresentasikan sebuah interaksi

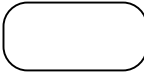
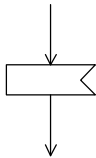
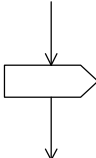
antara aktor dengan sistem. *Use case* merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, meng-*create* sebuah daftar belanja dan sebagainya. Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu.

Tabel 2.2 Simbol-simbol *Use Case Diagram*

No.	Simbol	Nama	Keterangan
1.		<i>Actor</i>	Merupakan kesatuan <i>eksternal</i> yang berinteraksi dengan sistem.
2.		<i>Use Case</i>	Rangkaian / uraian sekelompok yang saling terkait dan membentuk sistem.
3.		<i>Relationship</i>	Hubungan antara Pelaku/Aktor dengan <i>Use case</i>

2. *Activity Diagram*, menggambarkan berbagai alur aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alur berawal, *decicion* yang mungkin terjadi, dan bagaimana mereka berakhir.

Tabel 2.3 Simbol Activity Diagram

No.	Simbol	Nama	Keterangan
1.		<i>Initial State</i>	Titik awal dimulai <i>activity</i> .
2.		<i>Final State</i>	<i>Finish</i> (Akhir <i>Activity</i>)
3.		<i>State</i>	<i>Initial Activity</i>
4.		<i>Action State</i>	<i>Activity</i>
5.		<i>Decision</i>	Pilihan untuk mengambil keputusan
6.		<i>Fork</i>	Menunjukkan kegiatan yang dilakukan secara paralel
7.		<i>Receive</i>	Menerima pengiriman
8.		<i>Send</i>	Proses pengiriman

3. *Collaboration Diagram*, menggambarkan interaksi antar objek seperti *sequence diagram*, tetapi lebih menekankan pada peran masing-masing objek dan bukan pada waktu penyampaian *message*. Setiap *message* memiliki *sequence number*, dimana *message* dari *level* tertinggi memiliki nomor 1. *Message* dari *level* yang sama memiliki *prefiks* yang sama.

4. *Statechart Diagram*, menggambarkan transisi dan perubahan keadaan (dari satu *state* ke *state* lainnya) suatu objek pada sistem sebagai akibat dari *stimuli* yang diterima. Pada umumnya *statechart diagram* menggambarkan *class* tertentu (satu *class* dapat memiliki lebih dari satu *statechart diagram*).
5. *Sequence Diagram*, menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna dan *display*) berupa *message* yang digambarkan terhadap waktu. *Sequence Diagram* terdiri atas dimensi *vertikal* (waktu) dan dimensi *horizontal* (objek–objek yang terkait).
6. *Class Diagram*, *Class* adalah sebuah spesifikasi yang jika diinstansi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class Diagram* menggambarkan keadaan (atribut/*property*) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metode/fungsi).
7. *Component Diagram*, menggambarkan struktur dan hubungan antar komponen piranti lunak, termasuk ketergantungan (*dependency*) di antaranya.
8. *Deployment Diagram*, menggambarkan detail bagaimana komponen di-*deploy* dalam infrastruktur sistem, di mana komponen akan terletak (pada mesin, *server* atau piranti keras apa), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi *server*, dan hal–hal lain yang bersifat fisik.

2.2. Penelitian Sebelumnya

2.2.1. Analisis dan Perancangan Sistem Basis Data Manajemen Produksi pada PT. X

Penyimpanan informasi bagi perusahaan dapat dipermudah dengan penggunaan teknologi informasi melalui *Database Management System*. PT. X merupakan salah satu perusahaan yang bergerak di bidang manufaktur dan industri yang menghasilkan saringan knalpot untuk diekspor ke luar negeri. Produksi saringan knalpot pada PT. X terdiri atas beberapa proses, di mana setiap prosesnya melibatkan sebuah mesin. Pada setiap proses produksi, akan dilakukan pendataan mengenai informasi dari bahan-bahan produksi yang berguna untuk dianalisis lebih lanjut oleh bagian administrasi. Dengan meningkatnya permintaan akan saringan knalpot, maka produksi dan data mengenai bahan produksi juga bertambah. Oleh karena tidak adanya sistem yang dapat menangani data tersebut, maka data yang disimpan menjadi tidak terorganisir. Hal ini menyebabkan penurunan kinerja para buruh yang mengakibatkan terhambatnya proses produksi. PT. X membutuhkan sistem yang dapat mengatur data mengenai bahan produksi secara efektif. Karena itu, diangkatlah topik Analisis dan Perancangan Sistem Basis Data Manajemen Produksi pada PT. X. Kesimpulan dari penelitian ini bahwa melalui sistem basis data ini, penyimpanan data mengenai bahan produksi menjadi lebih terorganisir, kemudian sistem ini akan digunakan oleh PT. X pada bagian produksi saringan knalpot, dan sistem basis data ini juga berperan sebagai perangkat pendukung untuk tahap analisis oleh bagian administrasi. (Indrajani, 2010 : 1).

2.2.2 Desain Sistem Database Order Produksi pada Departemen Loom – Weaving 4, Divisi Denim PT. Apac Inti Corpora

Penjadwalan dalam organisasi industri mempunyai peranan penting terutama pada *level* operasional. Untuk mencapai sebuah penjadwalan yang baik, semua data pendukung harus diatur untuk mendukung proses penjadwalan. Departemen *Loom-Weaving 4*, Divisi Denim, PT. APAC INTI Corpora, khususnya pada Departemen PPC membutuhkan sebuah sistem basis data untuk mendukung proses penjadwalan. Untuk menjalankan fungsinya, Departemen PPC membutuhkan informasi yang cepat, akurat dan aktual untuk dibagi dengan bagian marketing sebagai dasar pengambilan keputusan dalam menerima atau menolak *order* dan menginformasikan status pembuatan *order*. Departemen PPC juga harus menyediakan informasi untuk pelaksanaan pembelian *order* oleh Departemen Produksi. Penelitian ini mengembangkan sebuah sistem basis data untuk memfasilitasi PPC dalam manajemen data dan menyediakan informasi *order*. Sistem basis data didesain untuk menyediakan bantuan dalam penyimpanan data *order*, pelaporan produksi dan merencanakan perbaikan jika terdapat *order* tambahan. (Prastawa, 2009 : 1).

2.3. Metode Analisis dan Perancangan Sistem

2.3.1 Metode Analisa Berorientasi Objek

Object Oriented Analysis (OOA) Yaitu, metode yang digunakan untuk menganalisa sistem, dengan metode ini dapat mempresentasikan sebuah permasalahan dalam dunia nyata kedalam *object-object*, khususnya dalam pengembangan perangkat lunak, agar dalam pelaksanaannya kita mendapatkan berbagai keuntungan dan kelebihan. Biasanya analisa sistem dimulai dengan adanya dokumen permintaan (*requirement*) yang diperoleh dari semua pihak yang berkepentingan.

Hasil analisis berorientasi objek adalah deskripsi dari apa sistem secara fungsional diperlukan untuk melakukan, dalam bentuk sebuah model konseptual. Itu biasanya akan disajikan sebagai seperangkat menggunakan kasus, satu atau lebih *UML* diagram kelas, dan sejumlah diagram interaksi. Tujuan dari analisis berorientasi objek adalah untuk mengembangkan model yang menggambarkan perangkat lunak komputer karena bekerja untuk memenuhi seperangkat persyaratan yang ditentukan pelanggan.

Rangkaian aktivitas dari *Object Oriented Analysis* yaitu :

1. Menganalisis masalah domain
2. Jelaskan sistem proses
3. Identifikasi obyek
4. Tentukan atribut
5. Mendefinisikan operasi
6. Komunikasi antar-obyek

(Whitten, 2001 : 408)

2.3.2. Metode Perancangan Berorientasi Objek

Object Oriented Design (OOD) adalah tahap perancangan dimulai dengan hasil keluaran yang dihasilkan tahap analisis, dan aktifitas yang dilakukan adalah secara perlahan bergeser tekananya dari domain aplikasi atau persoalan menuju domain komputasi. merupakan cara baru dalam memikirkan suatu masalah dengan menggunakan model yang dibuat menurut konsep sekitar dunia nyata. Dasar pembuatan adalah objek, yang merupakan kombinasi antara struktur data dan perilaku dalam satu entitas. (Whitten, 2004 : 647).

2.4. Metode Pengembangan Sistem

Metode pengembangan *Prototype* adalah metode pengembangan sistem dengan membuat model kerja yang bersifat optional dengan tujuan untuk mengidentifikasi sistem yang berjalan dan memberikan penambahan-penambahan fitur baru serta mengevaluasi kelayakan dan kemungkinan yang terjadi dari desain sistem yang dikembangkan (Indrajani, 2002 : 56).

Dalam pengembangan perangkat lunak (*software*) digunakan metode *Prototyping/pemodelan* (Pressman, 2002 : 39) yang terdiri atas :

1. *Requirements gathering*, yaitu analisa terhadap kebutuhan calon pemakai/klien :
 - a. Pengumpulan data : mengumpulkan data-data yang berkaitan dengan sistem yang akan dibangun
 - b. Analisis data : menganalisa data-data yang sudah terkumpul agar dapat dilihat kebutuhan yang diinginkan klien

2. *Quick Design*, yaitu pembuatan desain global untuk membentuk perangkat lunak *prototype* :
 - a. Membuat desain/rancangan secara keseluruhan yang akan digunakan oleh calon pemakai.
 - b. Desain yang dibuat masih hanya berupa *prototype* yang masih dalam bentuk rancangan.
3. *Build Prototype*, yaitu pembuatan perangkat lunak *prototype*, termasuk didalamnya adalah pengujian dan penyempurnaan *prototype* :
 - a. Desain yang sudah dipilih akan dibuat perangkat lunak *prototype*-nya dengan aplikasi yang sesuai dengan keinginan klien
 - b. Perangkat lunak yang sudah dibuat *prototype*-nya akan diuji kebenarannya dan keandalannya, sehingga nantinya akan dibuat *prototype* sebenarnya.
4. *Evaluate and Refine Requirements*, yaitu mengevaluasi *prototype* dan memperhalus analisis kebutuhan calon pemakai/klien :
 - a. *Prototype* yang sudah diuji dan disempurnakan dievaluasi kebenaran dan kemampuannya terhadap sistem.
 - b. Kebutuhan calon pemakai yang dianalisis dilihat kesesuaiannya terhadap perangkat lunak yang dibangun

5. *Engineer Product*, yaitu pembuatan perangkat lunak yang sebenarnya, yang termasuk didalamnya adalah: *design*, *coding*, dan *testing* terhadap perangkat lunak. Aktivitas-aktivitas tersebut dikelompokkan menjadi dua macam aktivitas, yaitu: Pembangunan perangkat lunak *prototype* (aktivitas 1 s.d 3) dan pembangunan perangkat lunak sebenarnya (aktivitas 4 s.d 5).