**BAB II**

**LANDASAN TEORI, KERANGKA BERPIKIR DAN HIPOTESIS**

1. **Landasan Teori**

Dalam bab kedua ini, penulis ingin menjelaskan terkait dengan landasan teori dari Rancang Bangun *System Monitoring Inventory* pada PT. Duan Biru Engineering berbasis Java.

1. Rancang Bangun

Menurut penulis, rancang bangun dapat disandingkan dengan kalimat perancangan dalam segi arti. Perancangan adalah kegiatan yang memiliki tujuan untuk mendesain sistem baru yang dapat menyelesaikan masalah-masalah yang dihadapi perusahaan yang diperoleh dari pemilihan alternatif sistem yang terbaik (Bin Ladjamudin, 2005:39).

Dari definisi di atas rancang bangun atau disebut juga dengan perancangan merupakan kegiatan merencanakan, menggambarkan maupun mengimplementasikan sebuah sistem baru dengan tujuan untuk memudahkan dan menyelesaikan berbagai masalah yang muncul dalam suatu organisasi sehingga mendukung tercapainya tujuan dari proses-proses yang lain dengan lebih mudah dan tepat sasaran.

1. *Monitoring*

*Monitoring* atau melakukan monitor menurut penulis merupakan kegiatan pemantauan yang dilakukan secara berkala dan terukur terhadap komponen-komponen atau elemen-elemen yang menjadi bagian dari sebuah sistem dengan tujuan untuk melihat memantau jalannya sistem agar tetap dalam kerangkan pencapaian suatu tujuan.

Pemantauan (*monitoring*) menjadi penting ketika sebuah organisasi yang baik mulai melihat seluruh kegiatannya tidak hanya berpatokan pada hasil produksi atau usaha yang dilakukan, namun juga semua proses yang berjalan dalam mencapai tujuan yang di sasar juga diperhatikan dengan teliti.

1. *Inventory*

Persediaan Barang (*Inventory*) dapat diartikan sebagai kegiatan untuk mengadakan barang-barang sesuai dengan kebutuhan dan keberlangsungan dari sebuah organisasi baik untuk kebutuhan sendiri (*internal*) maupun untuk kebutuhan pelanggan atau pihak luar (*external*).

Dalam kegiatan pengadaan barang ini tentunya terdapat kegiatan-kegiatan pendukung seperti kemana barang yang dibutuhkan harus dicari, bagaimana mendata barang-barang yang telah dating agar tercatat dengan benar , bagaimana penggunaan barang yang tersedia untuk keperluan di dalam maupun keperluan pelanggan atau pihak lain.

1. Sistem

Dalam konsep dasar sistem terdapat dua buah pendekatan dalam mendefiniskan arti dari sistem dengan melakukan pendekatan prosedural maupun pendekatan komponen atau elemen yang ada. Dalam pendekatan secara prosedural, sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran yang tertentu (Jogiyanto, 2005:1).

Sedangkan apabila dilihat dari pendekatan komponen atau elemen, sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu (Jogiyanto, 2005:2).

Kedua definisi diatas menurut penulis bias saling terkait dan berhubungan satu sama lain, dimana penulis simpulkan sistem merupakan kumpulan dari banyak komponen saling mempunyai jaringan satu dengan yang lain dihubungkan dengan prosedur-prosedur dengan tujuan untuk mencapai sasaran yang ditentukan dalam sebuah organisasi baik skala kecil, menengah maupun besar.

1. Java

Bahasa Java, diciptakan oleh Sun Microsystem pada 1995-an dengan dipimpin oleh Patrick Naughton dan James Gosling dimana bahasa java ini tidak tergantung pada *platform* tertentu. Java menurut definisi dari Sun adalah nama untuk sekumpulan teknologi untuk membuat dan menjalankan perangkat lunak pada computer *standalone* ataupun pada lingkukan jaringan (Rosa, 2010:246).

Java berdiri diatas sebuah mesin *interpreter* yang diberi nama dengan Java Virtual Machine (JVM). JVM inilah yang akan membaca bytecode dalam file *.class* dari suatu program. Oleh karena itu Java disebut sebagai bahasa pemrograman yang *portable* karena dapat dijalankan di berbagai sistem operasi, asalhkan sistem operasi tersebut telah disiapakan JVM di dalamnya.

Java 2 adalah generasi kedua dari Java *platform* yang terdiri dari tiga edisi java 2 untuk keperluan berbeda, yaitu:

* 1. Java 2 Standard Edition (J2SE)
  2. Java 2 Enterprise Edition (J2EE)
  3. Java 2 Micro Edition (J2ME)

Ruang lingkup hubungan dari ketiga versi dari java 2 dapat dilihat dari gambar 1 dibawah ini.



**Gambar 1.** Ruang lingkup keterhubungan J2EE, J2SE, dan J2ME

Untuk pengembangan aplikasi yang akan di buat oleh penulis, penulis lebih menggunakan Java yang berbasis *Standard Edition* yang akan terinstal secara *standalone* pada sebuah komputer*.*

Beberapa kemampuan yang disediakan oleh java diantaranya adalah :

1. Pemrograman berorientasi objek
2. Reflection
3. Interface dan Inner Class
4. Event Listener Model
5. Event listener Model
6. Graphical user interface
7. Mutithreding

Masih banyak lagi fungsi fungsi dari java yang belum. Disebutkan. Beragam kemampuan ini yang mendukung Java untuk dapat di gunakan oleh banyak orang serta berjalan pada banyak *platform*.

1. MySQL

Sebelum mengenal MySQL, penulis penulis mengajak untuk memahami terlebih dahulu *Relational DataBase Menagement System* (RDBMS). RDBMS dapat dianalogikan sebagai rak-rak penyimpanan data yang diberi tanda dan dapat dengan cepat di cari dengan bahasa *query*.

Sedangkan MySQL merupakan salah satu *database relational* (RDBMS) yang awalnya dibangun melalui komunitas, tetapi sekarang MySQL sudah dikuasai oleh Oracle (Agus Kurniawan, 2014:176).

Untuk mencari data yang dinginkan dari beberapa tabel yang terdapat pada MySQL dibutuhkan bahasa *Structure Query Language* (SQL) yang lebih mudah dikenal dengan bahasa *query* yang terstruktur. Selain *query* beberapa fitur yang terdapat dalam MySQL diantaranya :

1. Fungsi manipulasi data (tampikan, tambah, edit dan hapus data)
2. View
3. Trigger
4. Store Procedure
5. Store Function

Fitur-fitur tersebut sangat membatu dalam proses pencarian data yang tersimpan dalam *database*.

Agar bahasa Java dapat berkomunikasi dengan MySQL *database* dibutuhkan sebuah konektor agar bahasa Java dapat membaca isi data dari MySQL yang diberi nama dengan Java DataBase Connection (JDBC).

1. Diagram Arus Data

Untuk memudahkan dalam menggambarkan aliran data dari suatu sistem yang akan didesain, beberapa pakar terdahulu telah menggunakan simbol-simbol dalam perancangan programnya. Hal ini sangat dibuthkan untuk memudahkan dalam memahami perancangan suatu sistem yang cukup kompleks. Diagram yang menggunakan notasi-notasi ini untuk menggambarkan arus dari data sistem sekarang dikenal dengan nama diagram arus data atau DAD (*data flow diagram* *atau DFD*) (Jogiyanto, 2005:700).

Simbol-simbol yang digunakan pada DFD digambarkan mewakili beberapa hal:

1. Batas sistem atau kesatuan luar (*boundary* atau e*xternal entity*)

Setiap sistem mempunyai batas sistem yang memisahkan suatu sistem dengan lingkungan luarnya. Sistem akan menerima input dan menghasilkan output lingkungan luarnya. Kesatuan luar merupakan kesatuan yang berada diluar sistem dapat berupa orang, organisasi maupun sistem lain yang akan memberikan *input* terhadap sistem atau menerima *output* dari sistem.

Kesatuan luar digambarkan dengan notasi seperti pada gambar 2.



Gambar 2. Notasi kesatuan luar DAD

1. Arus data (*Data flow*)

Arus data (*data flow*) diberi symbol dengan panah. Arus data ini mengalir diantara proses (process), penyimpanan data (*data store*) serta kesatuan luar (*external entity*). Arus dapat menunjukkan aliran data berupa masukan untuk sistem atau hasil keluaran dari sistem.

Arus data sebaiknya diberi nama yang jelas dan memiliki arti yang dituliskan di samping pada gambar panah arus data.



Gambar 3. Arus data yang mengalir dari kesatuan luar langganan ke proses order dengan nama order langganan

1. Proses (*Process*)

Suatu proses adalah kegiatan atau kerja yang dilakukan oleh orang, mesin atau computer berdasarkan arus data yang masuk ke proses yang akan menghasilkan keluaran berupa arus data ke penympanan data (*data store*), ke proses lain sebagai masukan arus data, atau arus data menuju kesatuan luar yang lain. Symbol proses menggunakan lingkaran yang memiliki nama proses didalamnya, seperti ditunjukkan pada gambar 4.



Gambar 4. Notasi proses pada DAD

Setidaknya dalam membuat notifikasi proses diagram arus data harus ada :

1. Identifikasi proses

Merupakan nomor acuan dari proses berupa angka yang dituliskan pada bagian atas symbol.

1. Nama Proses

Menunjukan nama dari proses berupa apa yang dikerjakan oleh proses tersebut. Nama proses harus jelas dan lengkap, biasanya berbentuk suatu kalimat yang diawali dengan kata kerja (misal: mengjitung, membuat, mebandingkan, memverifikasi dan lain sebagainya) dan diletakkan dibawah identifikasi proses.

1. Penyimpanan data (*Data store*)

Penyimpanan data (*data store*) merupakan tempat penyimpanan data yang dapat berupa:

* 1. Suatu file atau database di sistem computer
  2. Suatu arsip atau catatan manual
  3. Suat kotak tempat data di meja seseorang
  4. Suatu agenda atau buku

Symbol notifikasi untuk penyimpanan data di DAD digambarkan pada gambar 5.



Gambar 5. Notifikasi pemrosesan data pada DAD

Nama dari penyimpanan data (*data store*) menunjukkan nama filenya atau apabila menggunakan *database* nama dapat menunjukkan nama tabel penyimpanan datanya.

Dalam pembuatan DAD terdapat 2 macam bentuk diagram arus data yaitu diagram arus data fisik atau DADF (*physical data flow diagram* atau PDFD) dan diagram arus data logika atau DADL (*logical data flow diagram* atau LDFD).

Sedangkah langkah langkah untuk menggambarkan sebuah DAD mengikut langkah-langkah berikut.

1. Identifikasi terlebih dahulu semua kesatuan luar (*external entities*) yang terlibat dalam sistem yang di desain. Kesatuan luar ini merupakan sumber data serta bias juga menjadi penerima informasi dari hasil proses sebuat sistem.
2. Identifikasi semua input dan output yang melibatkan kesatuan luar (*external entities*).

|  |  |  |
| --- | --- | --- |
| Kegiatan luar | Input | Output |
| Langganan | Order Langganan | - |
| Bagian gudang | - | Tembusan permintaan persediaan |
| Bagian pengiriman | Tembusan jurnal | Faktur,  tembusan kredit dan tembusan jurnal |
| Manajer kredit | - | Status piutang |

Table 1. contoh pembuatan DAD

1. Setelah semua teridentifikasi, gambarlah DAD mulai dari tingkat dasar atau di sebut dengan istilah diagram konteks (*context diagram*). Karena DAD merupakan alat yang digunakan untuk menggambarkan aliran data pada sistem informasi secara terstruktur (structured analysis), maka dari diagram konteks dapat dijabarkan mulai dari level terendah (*low level*) di DAD level 0. Dari DAD level 0 dapat di jabarkan ke level 1, dan seterusnya sampai sistem terkecil dari DAD memilki hanya satu masukan dan keluaran.

Penggambarannya dapat di lihat seperti contoh dalam gambar 6.



Sumber: <http://juwita.staff.gunadarma.ac.id/Downloads/files/3440/Pengenalan+DAD.doc>

Gambar 6. Contoh penggambaran DAD

1. Kamus Data

Kamus data (KD) atau *data dictionary* (DD) atau disebut juga dengan istilah *system data dictionary* adalah katalog fakta tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi (Jogiyanto, 2005:725). Fungsi dari kamus data adalah agar analis sistem dapat mendefinisikan data yang mengalir di sistem dengan lengkap seperti tergambar pada gambar 7.

Gambar 7. Kamus Data

Isi dari kamus data harus dapat mencerminkan keterangan yang jelas mengenai data yang dicatat. Untuk itu kamus data harus berisi beberapa hal berikut.

1. Nama Arus Data

Karena kamus data berisi data-data yang dibawa pada arus data yang mengalir, maka tentunya nama arus data harus disertakan, sehingga arus data dapat dijelaskan lebih detail pada kamus data nanti.

1. Alias

Dalam kamus data, alias atau nama lain di perlukan apabila nama lain dari kamus data ini ada. Nama lain dibutuhkan karena terdapat perbedaan penyebutan nama di beberapa departemen, sehingga penggunaan alias sangat membantu dalam penamaan kamus data.

1. Bentuk data

Bentuk data dari arus data juga harus disebutkan dalam kamus data ini. Bentuk data ini bisa dalam bentuk :

* Dokumen dasar atau formulir
* Dokumen hasil cetakan computer
* Laporan tercetak
* Tampilan di layar monitor
* Variable
* Parameter
* Field

Pencatatan bentuk data pada kamus data ini berguna untuk pengelompokan kamus data sesuai dengan kegunaan sewaktu perancangan sistem serta nantinya untuk merancang *database*.

1. Arus data

Dalam kamus data, arus data merupakan pencatatan mengalirnya data dari mana menuju ke mana. Tujuan dari pencatatan arus data adalah nantinya memudahkan mencari arus data di DAD yang bersesuaian.

1. Penjelasan

Dalam kamus data, untuk memperjelas tentang arti dari arus data yang dicatat, bagian penjelasan dapat disi dengan keterangan-keterangan detail tentang arus data tersebut.

1. Periode

Periode menunjukkan kapan terjadinya arus data ini untuk mengindikasikan kapan masukan data harus dimasukkan ke sistem, kapan proses dari program harus dilakukan dan kapan laporan dihasilkan.

1. Volume

Volume yang akan dicatat dalam kamus data adalah volume rata-rata atau volume puncak dari arus data. Volume rata-rata menunjukkan banyaknya rata-rata arus data yang menunjukkan arus data mengalir pada periode tertentu, sedangkan volume puncak menujukkan volume terbesar yang mengalir pada arus data tersebut.

1. Struktur data

Struktur data menunjukkan arus data yang dicatat kamus data yang terdiri dari item-item data yang dibutuhkan sesuai dengan aturan.

Gambar 8. Contoh sebuah kamus data

1. Diagram Relasi Entitas

Diagram Relasi Entitas atau *Entity Relationship Diagram* (ERD) adalah suatu model jaringan yang menggunakan susunan data yang disimpan dalam sistem secara abstrak (Bin Ladjamudin, 2005:142).

Dari definisi yang dijelaskan di atas dapat disimpulkan bahwa diagram relasi entitas merupakan suatu cara penggambaran model jaringan data yang tersusun dalam *database* yang memiliki komponen-komponen atribut yang masing-masing merepresentasikan fakta serta hubungan antar sebuah komponen dengan komponen yang lainnya.

ERD digunakan untuk memodelkan struktur data dan hubungan antar data. Dengan ERD, model dapat diuji dengan mengabaikan proses yang dilakukan. ERD pertama kali dideskripsikan oleh Peter Chen yang dibuat sebagai bagian dari perangkat lunak CASE. Notasi yang digunakan dalam ERD dapat dilihat pada Tabel di bawah ini :

|  |  |
| --- | --- |
| Notasi | Keterangan |
|  | Entitas : adalah suatu objek yang dapat diidentifikasi dalam lingkungan pemakai |
|  | Relasi :menunjukkan adanya hubungan diantar sejumlah entitas yang berbeda |
|  | Atribut : mendiskripsikan karakter entitas (atribut yang berfungsi sebagai kunci (*key*) diberi garis bawah) |
|  | Garis : sebagai penghubung antar relasi dengan entitas, relasi dan entitas dengan atribut |

Table 2. Notasi ERD oleh Peter Chen

1. Kardinalitas Relasi

Dalam ERD, kardinalitas dapat menunjukkan jumlah maksimal hubungan antara suatu entitas dengan entitas yang lainnya. Banyaknya hubungan yang terjadi antara entitas satu dengan lain disebut dengan derajad relasi.

1. *One to One Relationship*

Hubungan satu ke satu (*one to one relationship*) menujukkan adanya hubungan antar dua entitas satu dengan yang lain dimana satu entitas berhubungan dengan satu entitas lain saja. Contohnya seorang pegawai memiliki jabatan sebagai teknisi, dalam hal ini satu pegawai hanya memiliki satu jabatan.

Gambar 9. Hubungan satu ke satu (*one to one relationship*)

1. One to Many Relationship

Hubungan satu ke banyak (*one to many relationship*) menujukkan adanya hubungan antar dua entitas satu dengan yang lain dimana satu entitas dapat memiliki banyak atribut dari entitas yang lain. Contohnya seorang anggota perpustakaan dapat meminjam lebih dari satu buku.

 Gambar 10. Hubungan satu ke banyak (*one to many relationship*)

1. *Many to many relationship*

Hubungan banyak ke banyak (*many to many relationship*) menujukkan adanya hubungan antar dua entitas satu dengan yang lain dimana satu entitas dapat memiliki banyak atribut dari entitas yang lain dan juga sebaliknya entitas yang lain dapat memiliki banyak atribut dari entitas sebelumnya. Contohnya seorang mahasiswa dapat mengambiil beberapa mata kuliah, dan mata kuliah dapat di ambil oleh beberapa mahasiswa.



Gambar 11. Hubungan banyak ke banyak (*many to many relationship*)

Untuk membuat suatu ERD, sumber utama yang digunakan adalah kamus data yang telah di buat pada proses sebelumnya. Adapun teknik pembuatannya dapat mengikuti langkah-langkah berikut.

1. Memilih kelolmpok atribut yang sama yang akan dijadikan sebuah entitas dan menuntukan atribut kunci (*primary key*) yang merupakan sebuah acuan unik dari sebuah entitas.

Tabel 2.1 Tabel Notasi ERD 8 2.3.2

. a. One to One Relationship Hubungan antara file pertama dan file kedua adalah satu berbanding satu. Contoh : · pada pengajaran private satu guru satu siswa · “seorang guru mengajar seorang siswa, seorang siswa diajar oleh seorang guru” Entitas 1 1 Relationship 1 Entitas 2 Gambar 2.1 Gambar One-toOne Relationship b. One to Many atau Many to One Relationship Hubungan antara file pertama dan file kedua adalah satu berbanding banyak atau banyak berbanding satu. Contoh : · Dalam suatu perusahan satu bagian mempekerjakan banyak pegawai. · “Satu bagian mempekerjakan banyak pegawai, satu pegawai kerja dalam satu bagian” 9 Entitas 1 1 Relationship M Entitas 2 Gambar 2.2 One-to-Many Relationship c. Many to Many Relationship Hubungan file pertama dan file kedua adalah banyak berbanding banyak. Contoh : · Dalam universitas seorang mahasiswa dapat mengambil banyak matakuliah · “Satu mahasiswa mengambil banyak matakulih dan satu matakuliah diambil banyak mahasiswa.” Entitas 1 M Relationship M Entitas 2 Gambar 2.3 Many-to-Many Relationship 2.3.3 LANGKAH-LANGKAH PERANCANGAN TEKNIK E-R Sumber awal data teknik perencanaan database dengan ER adalah data dictionary (kumpulan data). Langkah-langkah perancangan ER: 1. Memilih kelompok atribut yang sama untuk dijadikan sebuah entitas dan menentukan primary key dengan syarat unik dan mewakili entitas 2. Menggambarkan Cardinality dari ER diagram berdasarkan analisa relasi yang didapat. Relasi yang terjadi dapat One to One, One to Many dan Many to Many relationship 3. Membentuk SKEMA DATABASE atau LRS (Logical Record Structure) berdasarkan ER diagram 10 Keterangan : § Bila relasi One to One maka foreign key diletakkan pada salah satu dari 2 entitas yang ada atau menyatukan ke dua entitas tersebut. § Bila relasi One to Many maka foreign key diletakkan di entitas yang Many § Bila relasi many to many maka dibuat “file konektor” yang berisi 2 foreign key yang berasal dari kedua entitas § Membentuk tabel-tabel berdasarkan primary key yang terpilih dengan syarat sudah mencapai aturan normalisasi sekurang-kurangnya 3NF dari Skema DB/LRS yang ada. 2.4 Normalisasi Teknik normalisasi banyak digunakan terutama pemula karena mudah dipahami dan diaplikasikan. 2.4.1 Dasar-dasar normalisasi Normal form (bentuk normal) adalah suatu klas dari skema database relasi yang didefinisikan untuk memenuhi tujuan dari tingginya integritas dan maintainability. Kreasi dari suatu bentuk normal disebut normalisasi. Normalisasi dicapai dengan penganalisaan ketergantungan diantara setiap individu attribut yang diassosiasikan dengan relasinya. 1. First normal form Suatu relasi ada dalam kondisi First Normal Form (1NF) jika dan hanya jika semua domain yang tercakup terdiri hanya atomic value, misalnya tidak ada pengulangan group (domain-domain) dalam suatu tuple. Keuntungan dari 1NF dibanding Unnormalized relation (UNRs) adalah pada bentuk penyederhanaan representasi dan kemudahan dalam pengembangan menggunakan suatu query language. Kekurangannya adalah kebutuhan terhadap duplikasi data. Sebagian besar sistem relasi (tidak semua) membutuhkan suatu relasi dalam bentuk 1NF. 11 2. Second Normal Form Suatu superkey adalah suatu himpunan dari satu atau lebih attribute, yang mana, dimana diambil secara khusus yang memungkinkan kita untuk mengidentifikasikan secara unik satu entitas atau relasi. Suatu Candidate key adalah suatu subset dari attribut-attribut pada superkey yang juga merupakan superkey dan tidak reducible ke superkey yang lain. Suatu primary key dipilih dari himpunan candidate key untuk digunakan pada suatu index untuk relasi yang bersangkutan. Kepemilikan dari satu atau beberapa attribute yang dapat didefinisikan secara unik dari nilai satu atau beberapa attribute disebut functional dependency. Suatu relasi adalah dalam posisi second normal form (2NF) jika dan hanya jika relasi tersebut juga dalam 1NF dan setiap nonkey attribute tergantung penuh pada primary key-nya. 2NF membutuhkan bahwa FD apapun didalam relasi harus berisi semua komponen dari primary key sebagai determinant, baik secara langsung atau transitif. 3. Third Normal Form Suatu relasi dalam Third Normal Form (3NF) jika dan hanya jika relasi tersebut dalam 2NF dan setiap non key attribute adalah nontransitive dependent pada primary key. 4. Boyce-Codd Normal Form BCNF adalah suatu bentuk yang lebih kuat dari normalisasi ke tiga. 3NF equivalent dengan perkataan bahwa untuk setiap nontrivial FD X A, dimana X dan A merupakan simple atau composite attribut, satu dari dua kondisi harus dipenuhi.X adalah superkey, atau A adalah prime attribute. BCNF mengelimisasi kondisi kedua dari 3NF.

Berikut conoth atas maka dapat diberikan kesimpulan bahwa

judul Rancang **Bangun Monitoring Inventory System pada PT. Daun Biru Engineering berbasis Java**.

1. **Penelitian yang Relevan**
2. **Kerangka Berpikir**
3. **Hipotesis Penelitian**