BAB II
TINJAUAN PUSTAKA

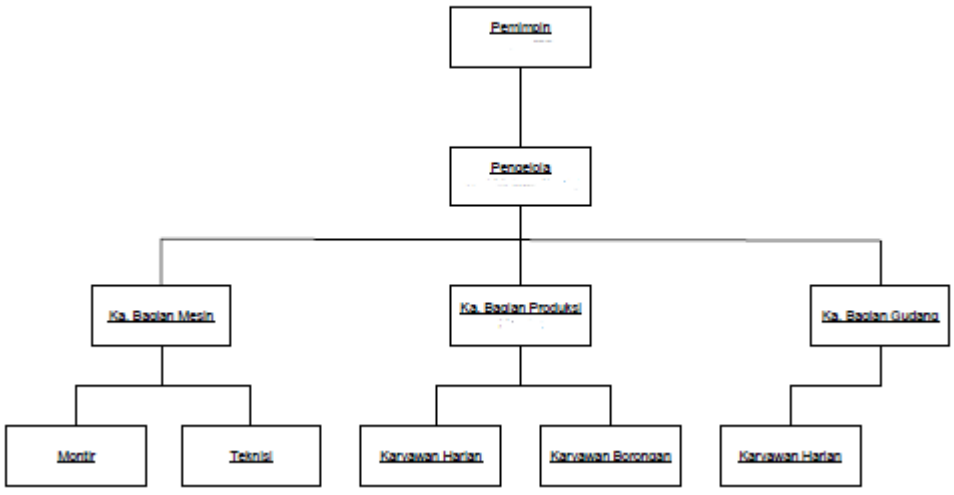
II.1. Tentang PT. Bandung Perkasa Jaya

PT. Bandung Perkasa Jaya merupakan perusahaan yang bergerak dibidang produski benang/s*pinning* dan merupakan perusahaan kecil yang akan berkembang. PT. Bandung Perkasa Jaya berdiri sejak tahun 2003 dengan tanah seluas 4,5 hektare. Pemasaranya hanya utnuk pasar lokal saja. Dan sampe saat ini, PT. Bandung Perkasa Jaya mempunyai sekitar 300 karyawan tetap. Mempunyai berbagai jenis mesin untuk produksi benang diantaranya, mesin *blowing* 2 buah, *carding*, *drawing*, *spinning* dan *winding*.

Nama Perusahaan : PT. Bandung Perkasa Jaya
Lokasi : Jalan Solokan Jeruk, Majalaya, Jawa Barat
Bidang : Produksi Benang/Spinning

II.2. Struktur Organisasi Perusahaan

Berikut ini adalah struktur organisasi PT. Bandung Perkasa Jaya yang dapat dilihat pada Gambar II.1 :



Gambar II.1.Struktur Organisasi PT.Bandung Perkasa Jaya

Uraian tugas dan wewenang masing-masing bagian pada struktur organisasi PT. Bandung Perkasa Jaya adalah sebagai berikut :

1. Pengelola

   Mengawasi kegiatan produksi, penjualan, keuangan, pembelian, administrasi dan kegiatan lain dalam rangka pencapaian tujuan perusahaan
2. Kepala Bagian Mesin

   Mengawasi Seluruh kegiatan produksi yang berhubungan dengan mesin atau alat produksi lainnya.
3. Kepala Bagian Produksi

   Megawasi kegiatan produksi dari awal bahan mentah hingga akhir proses menjadi barang siap jual.
4. Kepala Bagian Gudang

   Mengawasi kegiatan keluar masuk barang yang akan dijual atau dipesan

II.3. Landasan Teori

Landasan teori yang berkaitan dengan materi atau teori yang digunakan sebagai acuan dalam melakukan penelitian. Landasan teori yang diuraikan merupakan hasil dari literatur dan buku-buku.

II.4. Pengertian data mining

Data mining adalah suatu istilah yang digunakan untuk mneguraikan penemuan pengetahuan di dalam basis data. Didalamnya juga terdapar proses yang menggunakan teknik statistik, matematika, kecerdasan buatan, dan *machine learning* untuk mengekstraksi dan mengidentifikasi informasi yang bermanfaat dan pengetahuan yang terkait dari berbagai basis data.

Menurut Fajar Astuti Hermawati, "*Data mining* adalah proses yang memperkerjakan satu atau lebih teknik pembelajaran komputer untuk menganalisis dan mengekstrasi pengetahuan secara otomatis."

Menurut Han Jiawei, "*Data mining* adalah sebuag proses berulang bertujuan untuk menganalisa *database* yang besar dengan tujuan penggalian informasi dan

pengetahuan yang berkaitan dengan pengambilan keputusan dan pemecahan masalah."

Dari definisi-definisi yang telah disampaikan, hal-hal penting yang terkait dengan *data mining* adalah :

1. *Data mining* merupakan suatu proses otomatis terhadap data yang sudah ada.
2. Data yang akan diproses berupa data yang sangat besar
3. Tujuan *data mining* adalah mendapatkan hubungan atau pola yang mungkin memberikan indikasi yang bermanfaat.

II.4.1. Tahapan data mining

Proses *data mining* secara garis besar dapat dijelaskan sebagai berikut :

1. *Data Selection*

   Seleksi data dari sekumpulan data operasional perlu dilakukan sebelum tahap penggalian informasi dalam KDD dimulai. Data hasil seleksi yang akan digunakan untuk proses *data mining*, disimpan dalam suatu berkas, terpisah dari baris data operasional.

2. *Pre-Processing/cleaning*

   Sebelum proses *data mining* dapat dilakukan, perlu dilakukan dahulu proses *cleaning* pada data yang menjadi fokus KDD. Proses *cleaning* mencakup antara lain membuang duplikasi data, memeriksa data yang inkonsisten, dan memeperbaiki kesalahan pada data. Juga dilakukan proses *enrichment*, yaitu proses "meperkaya" data yang sudah ada dengan data atau informasi lain yang relevan dan diperlukan untuk KDD. *Pre-processing* data adalah hal yang harus dilakukan dalam proses data mining karena tidak semua data atau atribut data dalam data digunakan dalam proses data mining. Proses ini dilakukan agar data yang akan digunakan sesuai dengan kebutuhan. Adapun langkah-langkah *pre-processing* adalah sebagai berikut :

   a. Pemilihan atribut (*atribut selection*)

      Pemilihan atribut adalah proses pemilihan mana saha atribut data yang akan digunakan sehingga data tersebut dapat kita olah sesuai dengan kebutuhan proses data mining

b. Pembersihan data (*data cleaning*)

Proses menghilangkan noise dan menghilangkan data yang tidak relevan.

3. *Transformation*

Coding adalah proses transformasi pada ata yang telah dipilih, sehingga data tersebut sesuai untuk proses *data mining.* Proses *coding* dalam KDD merupakan proses kreatif dan sangat tergantung pad ajenis atau pola informasi yang akan dicari dalam basis data.

4. *Data mining*

Data mining adalah proses mencari pola atau informasi menarik dalam data terpilij dengan menggunakan teknik atau metode tertentu. Teknik, metode, atau algortima yang tepat sangat bergantung pada tujuan dan proses KDD secara keseluruhan.

5. *Interpretation/evaluation*

Pola informasi yang dihasilkan dari proses *data mining* perlu ditampilkan dalam bentuk yang mudah dimengerti oleh pihak yang berkepentingan. Tahap ini merupakan bagian dari proses KDD yang disebut *interpretation*. Tahap ini mencakup pemeriksaan apakah pola atau informasi yang disebut bertentangan dengan fakta atau hipotesis yang ada sebelumnya.

II.4.2. Arsitektur data mining

Secara garis besar, sistem data mining terdiri dari komponen-komponen berikut:

a. *Database, data warehouse,* atau media penyimpanan informasi

Media penyimpanan dalam hal ini bisa berupa *database, data warehouse spreadsheets,* atau jenis-jenis penampung informasi lainnya.

b. *Database* atau *data warehouse server*

Database atau *data warehouse server* berfungsi untuk menyediakan data yang relevan berdasarkan permintaan dari *user data mining.*

c. *Knowledge base*

Merupakan dasar pengetahuan yang digunakan sebagai panduan dalam pencarian pola.

d. *Data mining engine*

Merupakan bagian dari *software* yang menjalankan program berdasarkan algortima yang ada.

e. *Pattern evaluation module*

Merupakan bagian dari *software* yang berguna untuk menemukan *pattern* atau pola-pola yang terdapat di dalam *database* yang diolah sehingga nantinya proses data mining dapat menemukan *knowledge* yang sesuai.

f. *Graphical user interface*

Merupakan sarana antara pengguna dan sistem data mining untuk berkomunikasi, dimana pengguna dapat berinteraksi dengan sistem melalui data mining query, untuk menyediakan informasi yang dapat membantu dalam pencarian *knowledge.*

II.5. *Association Rule*

Analisis asosiasi atau *association rule mining* adalah teknik data mining untuk menemukan aturan asosiasi antara kombinasi item. Aturan asosiasi akan menggunakan data latihan, sesuai dengan perngertian data mining untuk menemukan *knowledge*. *Knowledge* untuk mengetahui item-item belanja yang sering dibeli secara bersamaan dalam suatu waktu.

Penting tidaknya suatu aturan asosiatif dapat diketahui dengan dua parameter, yaitu *support* dan *confidence*. *Support* (nilai penunjang) adalah suatu ukuran yang memnunjukkan seberapa besar tingkat dominasi suatu *item/itemset* layak untuk dicari *confidence*-nya, sedangkan *confidence* (nilai kepastian) adalah suatu ukuran yang menunjukkan hubungan antara 2 *item* secara *conditional*.

Metodologi dasar analisis asosiasi terbagi menjadi dua tahap [4] :

1. Analisis pola frekuensi tinggi

Tahap ini mencari kombinasi item yang memenuhi syarat minimum dari nilai support dalam database. Nilai support sebuah item diperoleh dengan rumus berikut.

$$Support(\mathrm{A}) = \frac{Jumlah\ transaksi\ mengandung\ A}{Total\ Transaksi} \quad \text{..... Persamaan (II.1)}$$

Sementara itu, nilai support dari 2 item diperoleh dari rumus berikut.

$$Support\ (A,B) = P(A \cap B)$$ ..... Persamaan (II.2)

$$Support\ (A,B) = \frac{\Sigma\, Transaksi\ mengandung\ A\ dan\ B}{\Sigma Transaksi}$$ ..... Persamaan (II.3)

2. Pembentukan Aturan asosiasi

Setelah semua pola frekuensi tinggi ditemukan, barulah dicari aturan asosiasi yang cukup kuat tingkat ketergantungan antar item dalam *antecedent* (pendahulu) dan *consequent* (pengikut) serta memenuhi syarata minimum untuk confidence dengan menghitung confidence aturan Asosiatif $A \nrightarrow B$

Misalkan D adalah himpunan transaksi, dimana setiap transaksi T dalam D merepresentasikan himpunan item yang berada alam I. I adalah himpunan item yang dijual. Misalkan kita memilih himpunan item A dan himpunan item lain B, kemudian aturna asosiasi akan berbentuk :

$$jika\ A, maka\ B\ (A \rightarrow B)$$

Dimana *antecedent* A dan *consequent* B merupakan subset dari I, dan A dan B merupakan *mutually exclusive* dimana aturan :

$$Jika\ A, maka\ B$$

Tidak berarti

$$Jika\ B, maka\ A$$

Sebuah itemset adalah himpunan item-item yang ada dalam I, dan k-temset adalah itemset yang berisi k item. Frekuensi itemset merupakan itemset yang memiliki frekuensi kemunculan lebih dari nilai minimum yang telah ditentukan (ϕ). Misalkan (ϕ) = 2, maka semua itemset yang frekuensi kemunculannya lebih dari atau sama dengan 2 kali disebut *frequent*. Himpunan dari *frequent* k-itemset dilambangkan dengan Fk.

Nilai *confidence* dari aturan $A \longrightarrow B$ diperoleh dari rumus berikut.

$$Confidence = P(B \mid A) = \frac{\Sigma\ Transaksi\ mengandung\ A\ dan\ B}{\Sigma\ Transaksi\ mengandung\ A}$$ .....Persamaan (II.4)

II.6. Algoritma FP-Growth

Algoritma *FP-Growth* merupakan pengembangan dari algoritma Apriori. Sehingga kekuragan dari algortima Apriori diperbaiki oleh algoritma *FP-Growth*[4].

Frequent Pattern Growth (FP-Growth) adalah salsah satu alternatif algoritma yang digunakan untuk menentuka himpunan data yang paling sering muncul dalam sebuah kumplan data[8].

Pada algortima Apriori diperlukan *generate candidate* untuk mendapatkan *frequent itemset*. Akan tetapi, pada algortima *FP-Growth generate candidate* tidak dilakukan karena *FP-Growth* menggunakan konsep pembangunan *tree* dalam pencarian *frequent itemset.* Hal ini yang menyebabkan algortima *FP-Growth* lebih cepat dari algortima Apriori.

Karakteristik algoritma *FP-Growth* adalah struktur data yang digunakan adalah *tree* yang disebut dengan *FP-Tree.* Dengan menggunakan *FP-Tree*, algortima *FP-Growth* dapat langsung mengekstrak *frequent itemset* dari *FP-Tree.*

Penggalian *itemset* yang *frequent* dengan menggunakan algortima *FP-Growth* akan dilakukan dengan cara membangkitkan struktur data *FP-Tree.* Metode *FP-Growth* dapat dibagi menjadi 3 tahapan utama yaitu[6]:

1. Tahap pembangkitan *conditional pattern base.*
2. Tahap pembangkitan *conditional FP-Tree*
3. Tahap pencarian *frequent itemset*

Tahap-tahap tersebut merupakan tahap yang akan dilakukan untuk mendapat *frequent itemset* yang dapat dilihat pada algortima berikut :

Input : FP-Tree Tree Output : Rt Sekumpulan lengkap pola frequent Method : FP-Ggrowth (Tree, null) Procedure : FP-Growth (Tree, _)

{
01 : if Tree mengandung single path P;
02 : then untuk tiap kombinasi (dinotasikan _) dari node node dalam path do
03 : bangkitkan pola _ _ dengan support daro node-node dalam _;
04 : else untuk tiap a_1 dalam header dari Tree
do
{
05 : bangkitkan pola
06 : bangun _ = a_1 _ dengan support = a_1.
Support
07 : if Tree _ = _
08 then panggil FP-Growth (Tree, _)
}

II.7. *FP-Tree*

FP-Tree merupakan struktur penyimpanan data yang dimampatkan dengan memetakan setiap data transaksi ke dalam setiap lintssan tertentu dalam *fp-tree*. Karena dalam setiap transaksi yang dipetakan, mungkin ada transaksi yang memiliki item yang sama, maka lintasannya memungkinkan untuk saling menimpa. Semakin banyak data transaksi yang memiliki item yang sama, maka proses pemampatan dengan struktur data *fp-tree* semakin efektif. Kelebihan dari *fp-tree* yaitu hanya memerlukan dua kali pemindaian data transaksi. Berikut merupakan definisi lain dari *fp-tree :*

1. *Fp-tree* dibentuk oleh sebuah akar yang diberi label *null*, sekumpulan pohon yang beranggotakan item-item tertentu, dan sebuah label *frequent header.*
2. Setiap simpul dalam *fp-tree* mengandung tiga informasi penting, yaitu *label item*, menginformasikan jenis item yang dipresentasikan simpul tersebut, *support count* mempresentasikan jumlah lintasan transaksi yang melalui simpul tersebut, dan *pointer* penghubung yang menghubungkan simpul-simpul dengan label item sama antar lintasan, ditandai dengan garis putus-putus.

II.8. Basis Data (*Database)*

Basis Data terdiri dari atas dua kata, yaitu Basis dan Data. Basis kurang lebih dapat diartikan sebagai markas atau gudang, tempat bersarang atau berkumpul. Sedangkan Data adalah representasi fakta dunia nyata yang mewakili objek seperti manusia, barang, hewan, peristiwa, konsep, keadaan, dan sebagainya yang direkam dalam bentuk angka, huruf, simbol, teks, gambar, bunyi, atau kombinasinya.

Database adalah kumplan data yang saling berkaitan, berhubungan yang disimpan secara bersama-sama sedemikian rupa tanpa pengulangan yang tidak perlu, untuk memenuhi berbagai kebutuhan. Data-data ini harus mengandung semua informasi untuk mendukung semua kebutuhan sistem. Proses dasar yang dimiliki oleh database ada empat, yaitu:

1. Pembuatan data-data baru (*create database*)
2. Penambahan data (*insert*)
3. Mengubah data *(update*)
4. Menghapus data (*delete*).

Database merupakan salah satu komponen yang penting dalam sistem informasi, karena merupakan basis dalam menyediakan informasi pada para pengguna. Basis data (database) menjadi penting karena munculnya beberapa masalah bila tidak menggunakan data yang terpusat, seperti adanya duplikasi data, hubungan antar data tidak jelas, organisasi data dan update menjadi rumit. Jadi tujuan dari pengaturan data dengan menggunakan basis data adalah :

a. Menyediakan penyimpanan data untuk dapat digunakan oleh organisasi saat sekarang dan masa yang akan datang.
b. Cara pemasukan data sehingga memudahkan tugas operator dan menyangkut pula waktu yang diperlukan oleh pemakai untuk mendapatkan data serta hak-hak yang dimiliki terhadap data yang ditangani.
c. Pengendalian data untuk setiap siklus agar data selalu up-to-date dan dapat mencerminakan perubahan spesifik yang terjadi di setiap sistem.

d. Pengamanan data terhadap kemungkinan penambahan, modifikasi, pencurian dan gangguan-gangguan lain.

Dalam basis data sistem informasi digambarkan dalam model entity relationship (E-R). Bahasa yang digunakan dalam basis data (database) yaitu :

a. DDL (*Data Definition Language*)

Merupakan bahasa definisi data yang digunakan untuk membuat dan mengelola objek database seperti database, tabel dan view.

b. DML (*Data Manipulation Language*)

Merupakan bahasa manipulasi data yang digunakan untuk memanipulasi data pada objek database seperti tabel.

c. DCL (*Data Control Language*)

Merupakan bahasa yang digunakan untuk mengendalikan pengaksesan data.Penyusunan basis data meliputi proses memasukkan data kedalam media penyimpanan data, dan diatur dengan menggunakan perangkat Sistem Manajemen Basis Data (*Database Management System* / DBMS).

II.9. Database Management System (DBMS)

Managemen Sistem Basis Data (*Database Management System* /DBMS) adalah perangkat lunak yang di desain untuk membantu dalam hal pemeliharaan dan utilitas kumpulan data dalam jumlah besar.

Sistem Manajemen Basis data (Database Management System) merupakan sistem pengoperasian dan sejumlah data pada komputer. Dengan sistem ini dapat merubah data, memperbaiki data yang salah dan menghapus data yang tidak dapat dipakai. Sistem manajemen database merupakan suatu perluasan software sebelumnya mengenai software pada generasi komputer yang pertama. Dalam hal ini data dan informasi merupakan kesatuan yang saling berhubungan dan berkerja sama yang terdiri dari: peralatan, tenaga pelaksana dan prosedur data. Sehingga pengolahan data ini membentuk sistem pengolahan data. Peralatan dalam hal ini berupa perangakat keras (*hardware*) yang digunakan, dan prosedur data yaitu berupa perangakat lunak (*software*) yang digunakan dan dipakai untuk mengalokasikan dalam pembuatan sistem informasi pengolahan database.

Manipulasi basis data meliputi pembuatan pernyataan (*query*) untuk mendapatkan informasi tertentu, melakukan pembaharuan atau penggantian (*update*) data, serta pembuatan *report* dari data. Tujuan utama DBMS adalah untuk menyediakan tinjauan abstrak dari data bagi user. Jadi sistem menyembunyikan informasi mengenai bagaimana data disimpan dan dirawat, tetapi data tetap dapat diambil dengan efisien.

Pertimbangan efisiensi yang digunakan adalah bagaimana merancang struktur data yang kompleks, tetapi tetap dapat digunakan oleh pengguna yang masih awam, tanpa mengetahui kompleksitas stuktur data. Sistem manajemen database atau *database management system* (DBMS) adalah merupakan suatu sistem *software* yang memungkinkan seorang user dapat mendefinisikan, membuat, dan memelihara serta menyediakan akses terkontrol terhadap data. Database sendiri adalah sekumpulan data yang berhubungan dengan secara logika dan memiliki beberapa arti yang saling berpautan. DBMS yang utuh biasanya terdiri dari :

1. *Hardware* merupakan sistem komputer aktual yang digunakan untuk menyimpan dan mengakses database. Dalam sebuah organisasi berskala besar, *hardware* terdiri : jaringan dengan sebuah *server* pusat dan beberapa program client yang berjalan di komputer *desktop*.
2. *Software* beserta *utility Software* adalah DBMS yang aktual. DBMS memungkinkan para user untuk berkomunikasi dengan *database*. Dengan kata lain DBMS merupakan mediator antara database dengan user. Sebuah *database* harus memuat seluruh data yang diperlukan oleh sebuah organisasi.
3. Bagian integral dari setiap sistem adalah sekumpulan prosedur yang mengontrol jalannya sistem, yaitu praktik-praktik nyata yang harus diikuti user untuk mendapatkan, memasukkan, menjaga, dan mengambil data.
4. Data adalah jantung dari DBMS. Ada dua jenis data. Pertama, adalah kumpulan informasi yang diperlukan oleh suatu organisasi. Jenis data kedua adalah metadata, yaitu informasi mengenai database.

5. Ada sejumlah user yang dapat mengakses atau mengambil data sesuai dengan kebutuhan penggunaan aplikasi-aplikasi dan interface yang disediakan oleh DBMS, antara lain adalah :
   a. *Database administrator* adalah orang atau group yang bertanggungjawab mengimplementasikan sistem database di dalam suatu organisasi.
   b. Enduser adalah orang yang berada di depan workstation dan berinteraksi secara langsung dengan sistem.
   c. Programmer aplikasi, orang yang berinteraksi dengan database melalui cara yang berbeda.

II.10. Entity Relationship Diagram (ERD)

ERD adalah suatu model jaringan yang menggunakan susunan data yang disimpan dalam sistem secara abstrak[6]. Basis data Relasional adalah kumpulan dari relasi-relasi yang mengandung seluruh informasi berkenaan suatu entitas/ objek yang akan disimpan di dalam database. Tiap relasi disimpan sebagai sebuah file tersendiri. Perancangan basis data merupakan suatu kegiaatan yang setidaknya bertujuan sebagai berikut:

a. Menghilangkan redundansi data.
b. Meminimumkan jumlah relasi di dalam basis data.
c. Membuat relasi berada dalam bentuk normal, sehingga dapat meminimumkan permasalahan berkenaan dengan penambahan, pembaharuan dan penghapusan.

ERD adalah suatu pemodelan dari basis data relasional yang didasarkan atas persepsi di dalam dunia nyata, dunia ini senantiasa terdiri dari sekumpulan objek yang saling berhubungan antara satu dengan yang lainnya. Suatu objek disebut entity dan hubungan yang dimilikinya disebut relationship. Suatu entity bersifat unik dan memiliki atribut sebagai pembeda dengan entity lainnya.

Contoh : entity Mahasiswa, mempunyai atribut nama, umur, alamat, dan nim.

Diagram E-R terdiri dari:

a. Kotak persegi panjang, menggambarkan himpunan entitas.
b. Elips, menggambarkan atribut-atribut entitas.
c. Diamon, menggambarkan hubungan antara himpunan entitas
d. Garis, yang menghubungkan antar objek dalam diagram E-R

E-R Diagram merupakan suatu bahasa pemodelan dimana posisinya dapat dianalogikan dengan story board dalam industri film, blue print arsitektur suatu bangunan, miniatur, dan lain-lain. Dalam praktiknya, membangun suatu sistem terlebih dahulu dilakukannya suatu perencanaan. Pemodelan merupakan suatu sub bagian dari perencanaan secara keseluruhan sebagai salah satu upaya feedback evaluasi perampungan suatu perencanaan. E-R Diagram sebagai suatu pemodelan setidaknya memiliki beberapa karakteristik dan manfaat sebagai berikut:

a. Memudahkan untuk dilakukannya analisis dan perubahan sistem sejak dini, bersifat murah dan cepat
b. Memberikan gambaran umum akan sistem yang akan di buat sehingga memudahkan developer.
c. Menghasilkan dokumentasi yang baik untuk client sebagai bahan diskusi dengan bentuk E-R Diagram itu sendiri, dan
d. Kamus data bagi bagi para pengembang database.

Struktur dari E-R Diagram secara umum ialah terdiri dari:

a. Entitas merupakan objek utama yang informasi akan disimpan, iasanya berupa kata benda, ex; Mahasiswa, Dosen, Nasabah, Mata Kuliah, Ruangan, dan lain-lain. Objek dapat berupa benda nyata maupun abstrak.
b. Atribut merupakan deskripsi dari objek yang bersangkutan.
c. Relationship merupakan suatu hubungan yang terjalin antara dua entitas yang ada.

Kardinalitas Relasi ERD yang mempersentasikan suatu basis data relasional senantiasa memiliki relasi-relasi dari sejumlah entitas yang dapat ditentukan banyaknya. Banyaknya suatu relasi yang dimiliki oleh suatu relasi entitas disebut derajat relasi. Derajat relasi maksimum disebut dengan

kardinalitas, sedangkan derajat minimum disebut dengan modalitas. Kardinalitas yang terjadi diantara dua himpunan entitas (misal A dan B) dapat berupa :

a. *One to One*, satu record dipetakan dengan satu record di entitas lain. Contoh: satu nasabah punya satu account.
b. *One to Many*, Satu record dapat dipetakan menjadi beberapa record di entitas lain. Contoh: satu nasabah dapat punya lebih dari satu account.
c. *Many to Many*, Beberapa record dapat dipetakan menjadi beberapa record di entitas lain. Contoh: satu nasabah dapat memiliki lebih dari satu account. Satu account dapat dimiliki lebih dari satu nasabah (join account).

II.11. Data Flow Diagram (DFD)

DFD adalah suatu model logika data atau proses yang dibuat untuk menggambarkan dari mana asal data dan kemana tujuan data yang keluar dari sistem, dimana data disimpan, proses apa yang menghasilkan data tersebut dan interaksi antara data yang tersimpan dan proses yang dikenakan pada data tersebut.

DFD sering digunakan untuk menggambarkan suatu sistem yang telah ada atau sistem baru yang akan dikembangkan secara logika tanpa mempertimbangkan lingkungan fisik dimana data tersebut mengalir atau dimana data tersebut akan disimpan.

DFD merupakan alat yang digunakan pada metodologi pengembangan sistem yang terstruktur. Kelebihan utama pendekatan aliran data, yaitu :

a. Kebebasan dari menjalankan implementasi teknis sistem.
b. Pemahaman lebih jauh mengenai keterkaitan satu sama lain dalam sistem dan subsistem.
c. Mengkomunikasikan pengetahuan sistem yang ada dengan pengguna melalui diagram aliran data.
d. Menganalisis sistem yang diajukan untuk menentukan apakah data-data dan proses yang diperlukan sudah ditetapkan.

Disamping itu terdapat kelebihan tambahan, yaitu :

a. Dapat digunakan sebagai latihan yang bermanfaat bagi penganalisis, sehingga bisa memahami dengan lebih baik keterkaitan satu sama lain dalam sistem dan subsistem.
b. Membedakan sistem dari lingkungannya dengan menempatkan batas-batasnya.
c. Dapat digunakan sebagai suatu perangkat untuk berinteraksi dengan pengguna.
d. Memungkinkan penganalisis menggambarkan setiap komponen yang digunakan dalam diagram.

DFD terdiri dari context diagram dan diagram rinci (DFD Levelled). Context diagram berfungsi memetakan model lingkungan (menggambarkan hubungan antara entitas luar, masukan dan keluaran sistem), yang direpresentasikan dengan lingkaran tunggal yang mewakili keseluruhan sistem. DFD levelled menggambarkan sistem sebagai jaringan kerja antara fungsi yang berhubungan satu sama lain dengan aliran dan penyimpanan data, model ini hanya memodelkan sistem dari sudut pandang fungsi.

Dalam DFD levelled akan terjadi penurunan level dimana dalam penurunan level yang lebih rendah harus mampu merepresentasikan proses tersebut ke dalam spesifikasi proses yang jelas. Jadi dalam DFD levelled bisa dimulai dari DFD level 0 kemudian turun ke DFD level 1 dan seterusnya. Setiap penurunan hanya dilakukan bila perlu. Aliran data yang masuk dan keluar pada suatu proses di level x harus berhubungan dengan aliran data yang masuk dan keluar pada level x+1 yang mendefinisikan proses pada level x tersebut. Proses yang tidak dapat diturunkan/dirinci lagi dikatakan primitif secara fungsional dan disebut sebagai proses primitif.

II.12. Kamus Data (Data Dictionary)

Merupakan katalog (tempat penyimpanan) dari elemen-elemen yang berada dalam satu sistem. Kamus data mempunyai fungsi yang sama dalam pemodelan sistem dan juga berfungsi membantu pelaku sistem untuk mengerti aplikasi secara detail, dan me-reorganisasi semua elemen data yang digunakan

dalam sistem sehingga pemakai dan penganalisa sistem punya dasar pengertian yang sama tentang masukan, keluaran, penyimpanan dan proses.

Kamus data mendefinisikan elemen data dengan fungsi sebagai berikut :

a. Menjelaskan arti aliran data dan penyimpanan dalam DFD.
b. Mendeskripsikan komposisi paket data yang bergerak melalui aliran misalnya alamat diuraikan menjadi kota, negara dan kode pos.
c. Mendeskripsikan komposisi penyimpanan data.
d. Menspesifikasikan nilai dan satuan yang relevan bagi penyimpanan dan aliran data.
e. Mendeskripsikan hubungan detail antar penyimpanan yang akan menjadi titik perhatian dalam Diagram Keterhubungan Entitas (E-R).

II.13. MySQL

MySQL adalah sebuah program database server yang mampu menerima dan mengirimkan datanya dengan sangat cepat, multi user serta menggunakan perintah standar SQL (structure Query Language). MySQL merupakan sebuah database server yang free, artinya kita bebas menggunakan database ini untuk keperluan pribadi atau usaha tanpa harus membeli atau membayar lesensinya.

MySQL sendiri adalah sebuah database server yang mampu menangani beberapa user didalamnya. Dengan demikian, MySQL juga mampu menangani beberapa instruksi sekaligus dalam setiap waktu akses. Di dalam sistemnya, MySQL merekam semua data user dalam sebuah tabel user yang berada pada database yang bernama mysql. Dalam tabel user tersebut demua akses dan hak akses user mampu ditangani dengan baik.

Adapun beberapa kelebihan MySQL dalam teknik mengakses data maupun mengelola servernya :

1. Local Client

   MySQL sebagai sebuah Server database, juga dapat berfungsi sebagai client yang dijalankan pada komputer lokal dimana MySQL itu berjalan.

2. Remote Client

   Dapat diakses melalui komputer jaringan dengan cara remote.

3. Remote Login

   Dapat diakses dengan menggunakan program MySQL yang berada pada komputer lain.

4. Web Browser

   Dapat diakses menggunakan program yang dibuat dengan sebuah program yang berbasis server site yang berjalan di bawah Web Browser.

5. Scripting Language

   Dapat diakses melalui programaplikasi client yang kita ciptakan sendiri dengan menggunakan pemrograman visual maupun non visual yang berjalan di dalam jaringan.

II.14. Microsoft Visual Studio .Net dan C#

Visual studio .Net adalah sekumpulan alat pengembangan software yang diperuntukkan bagi .Net platform[8]. Microsoft Visual Studio merupakan sebuah perangkat lunak lengkap (suite) yang dapat digunakan untuk melakukan pengembangan aplikasi, baik itu aplikasi bisnis, aplikasi personal, ataupun komponen aplikasinya, dalam bentuk aplikasi console, aplikasi Windows, ataupun aplikasi Web. Microsoft Visual Studio dapat digunakan untuk mengembangkan aplikasi dalam native code (dalam bentuk bahasa mesin yang berjalan di atas Windows) ataupun managed code (dalam bentuk Microsoft Intermediate Language di atas .NET Framework).

C# sering dianggap sebagai bahasa penerus C++ atau versi canggih dari C++. C# adalah sebuah bahasa pemrograman yang dikembangkan oleh Microsoft dan menjadi salah satu bahasa pemrograman yang mendukung .Net programming melalui Visual Studio. C# didasarkan pada bahasa pemrograman C++, C# juga memiliki kemiripan dengan beberapa bahasa pemrograman seperti Visual Basic, Java, Delphi dan tentu saja C++.