BAB 2
TINJAUAN PUSTAKA

2.1 Profil Tempat Penelitian

Profil Koperasi Maju Terus meliputi meliputi sejarah instansi, visi dan misi, struktur organisasi, dan deskripsi jabatan.

2.1.1 Sejarah Instansi

Koperasi Maju Terus adalah salah satu organisasi yang bergerak di bidang penyimpanan dan peminjaman dana. Pada awalnya karyawan PT.Daliatex mengusulkan diadakannya koperasi sebagai tempat untuk penyimpanan dan peminjaman dana. Pada tahun 1993, pemilik perusahaan mempercayakan Bapak Ilyasa Sutopo untuk mendirikan koperasi ini. Setelah melalui berbagai pertimbangan, Koperasi ini resmi berdiri tanggal 5 Januari 1994 dibawah pengawasan PT.Daliatex.

Nama koperasi diambil berdasarkan rapat pengurus koperasi yaitu Koperasi Maju Terus. Melalui nama itupun diharapkan koperasi ini bisa maju terus dalam mengembangkan sistem simpan pinjam yang telah ada, serta diharapkan dapat maju terus dalam meningkatkan pelayanan bagi nasabah.

2.1.2 Visi dan Misi

Visi

Terwujudnya Koperasi Simpan Pinjam yang mandiri dan tangguh dengan berlandaskan amanah dalam membangun ekonomi bersama.

Misi

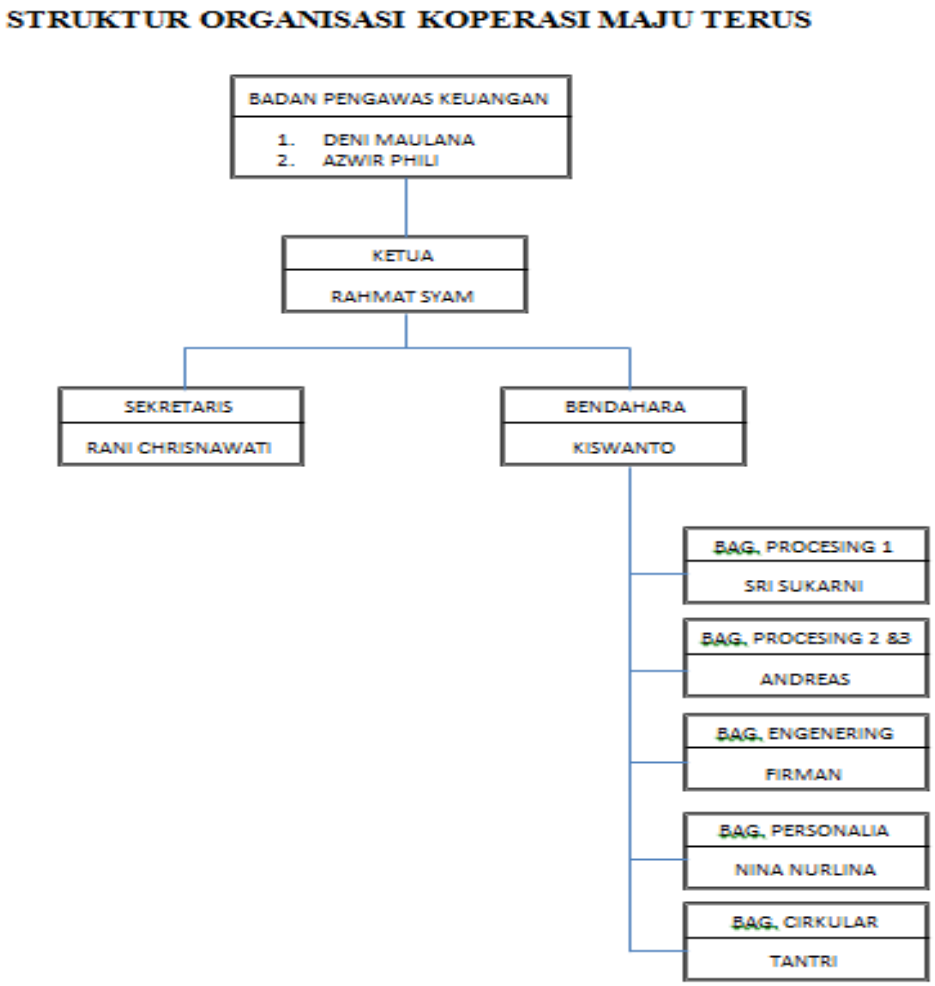
Upaya untuk mewujudkan Visi, Koperasi Simpan Pinjam memiliki Misi sebagai berikut :

1. Membantu anggota di dalam mobilisasi permodalan demi kelancaran usaha sehingga dapat meningkatkan kesejahteraan mereka.
2. Memberdayakan koperasi.

3. Memberikan kredit berbunga rendah kepada anggota.
4. Membangun dan mengembangkan jaringan distribusi dan jaringan ekonomi sampai kepada masyarakat umum.
5. Mewujudkan iklim usaha yang kondusif bagi perkembangan Koperasi.

2.1.3 Struktur Organisasi

Struktur Organisasi adalah suatu susunan dan hubungan antara tiap bagian serta posisi yang ada pada suatu organisasi atau perusahaan dalam menjalankan kegiatan operasional untuk mencapai tujuan. Struktur organisasi Koperasi Maju Terus dapat dilihat pada Gambar 2.1.



Gambar 2. 1 Struktur Organisasi Koperasi Maju Terus

2.1.4 Deskripsi Jabatan Koperasi Maju Terus

Deskripsi jabatan (*Job Description*) masing-masing bagian dari struktur organisasi koperasi Maju Terus adalah sebagai berikut:

1. Badan Pengawasan Keuangan
    a. Mengawasi peredaran uang dan atau transaksi simpan pinjam yang dilakukan koperasi.
    b. Memeriksa laporan keuangan.
    c. Pemeriksaan tertutup keuangan tahunan.
2. Ketua
    a. Mengawasi kegiatan penyimpanan dana, peminjaman dana, keuangan dan umum keadaan kegiatan dalam rangka pencapaian tujuan perusahaan.
    b. Mengawasi dan memberikan persetujuan pengeluaran dan penerimaan dana agar sesuai dengan anggaran yang sudah ditetapkan.
    c. Secara periodik mengawasi sistem pengawasan internal dan melaksanakan pemeriksaan internal.
    d. Menandatangani surat perjanjian peminjaman.
3. Sekretaris
    a. Membuat laporan transaksi simpan pinjam.
    b. Membuat surat perjanjian bagi nasabah.
    c. Mengklasifikasikan data dan dokumen sumber untuk melakukan persiapan pembukuan laporan.
    d. Mengisi, membukukan, menyiapkan dan mengirimkan laporan kepada katua koperasi.
4. Bendahara Umum
    a. Mengawasi bendahara-bendahara bagian.
    b. Menerima laporan SHU (Sisa Hasil Usaha) dari bendahara bagian.
    c. Menandatangani surat perjanjian peminjaman.

d. Mensetujui peminjaman kepada nasabah berdasarkan syarat dan ketentuan pinjaman koperasi.

5. Bendahara Bagian
   a. Mengawasi transaksi nasabah menurut bagiannya masing-masing.
   b. Mengawasi dan memberikan persetujuan peminjaman dana kepada nasabah sesuai bagiannya masing-masing.

Membuat laporan SHU (Sisa Hasil Usaha) untuk nasabah menurut bagiannya masing-masing.

2.2 Landasan Teori

Landasan Teori bertujuan memberikan gambaran dari teori yang terkait dalam pembangunan aplikasi.

2.2.1 Pengertian Data Mining

Data mining, sering juga disebut *knowledge discovery in database* (KDD), adalah kegiatan yang meliputi pengumpulan, pemakaian data historis untuk menentukan keteraturan, pola atau hubungan dalam set data berukuran besar. Keluaran dari *data mining* ini bisa dipakai untuk memperbaiki pengambilan keputusan di masa depan. [1]

Data mining adalah serangkaian proses untuk menggali nilai tambah dari suatu kompulan data berupa pengetahuan yang selama ini tidak diketahui secara manual. [2]

Data mining adalah serangkaian proses untuk menggali nilai tambah dari suatu kumpulan data berupa pengetahuan yang selama ini tidak diketahui secara manual, juga diartikan sebagai analisa otomatis dari data yang berjumlah besar atau kompleks dengan tujuan untuk menemukan pola atau kecenderungan yang penting yang biasanya tidak disadari keberadaannya.

2.2.2 Fungsi Data Mining

Mengacu kepada Lorase (2005), terdapat enam fungsi dalam data mining, yaitu: [3]

1. Fungsi deskripsi (*description*)
2. Fungsi estimasi (*estimation*)
3. Fungsi prediksi (*prediction*)
4. Fungsi klasifikasi (*classification*)
5. Fungsi pengelompokan (*clustering*), dan
6. Fungsi asosiasi (*association*)

Mengacu kepada Berry dan Browne (2006), keenam fungsi *data mining* tersebut dapat dipilah menjadi :

1. Fungsi minor atau fungsi tambahan, yang meliputi ketiga fungsi yang pertama, yaitu deskripsi, etimasi, dan prediksi.
2. Fungsi mayor atau fungsi utama, yang meliputi ketiga fungsi berikutnya, yaitu klasifikasi, pengelompokan, dan asosiasi.

(Fayyad, 1996).

2.2.3 Tahapan Data Mining

Istilah *data mining* dan *knowledge discovery in databases* (KDD) sering kali digunakan secara bergantian untuk menjelaskan proses penggalian informasi tersembunyi dalam suatu basis data yang besar. Sebenarnya kedua istilah tersebut memiliki konsep yang berbeda akan tetapi berkaitan satu sama lain. Dan salah satu tahapan dalam keseluruhan proses KDD adalah *data mining*. Proses KDD secara garis besar dapat dijelaskan sebagai berikut : [4]

1. *Data Selection*

   Pemilihan (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 basis data operasional.

2. *Pre-processing/ Cleaning*

   Sebelum proses *data mining* dapat dilaksanakan, perlu dilakukan proses *cleaning* pada data yang menjadi fokus KDD. Proses *cleaning* mencakup antara lain membuang duplikasi data, memeriksa data yang inkonsisten, dan memperbaiki kesalahan pada data, seperti kesalahan cetak (*tipografi*). Juga dilakukan proses *enrichment*, yaitu proses "memperkaya" data yang sudah ada dengan data atau informasi lain yang relevan dan diperlukan untuk KDD, seperti data atau informasi eksternal. Adapun tahapan dari *preprocessing* adalah sebagai berikut:

   a. Ekstrasi data (*data extraction*)

      Ekstraksi data dapat diartikan sebagai proses pengambilan data dari sumber data dalam rangka untuk melanjutkan proses pengolahan data ketingkat selanjutnya ataupun untuk menyimpan data hasil *ekstrak* tersebut.

   b. Pembersihan data (*data cleaning*)

      Proses menghilangkan *noise* dan menghilangkan data yang tidak *relevan* atau *inkonsisten* disebut pembersihan data.

   c. Pemilihan atribut (*atribut selection*)

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

3. *Transformation*

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

Pada tahap ini menggunakan proses normalisasi yaitu Min-max Normalization**.** Jika *minA* dan *maxA* merupakan nilai-nilai minimum dan maximum dari atribut A, minmax normalization memetakan sebuah nilai *v* dari A menjadi *v'* dalam rentang [*new_minA, new_maxA*] dengan rumus:

$$v' = \frac{v - min_A}{max_A - min_A} (newmax_A - newmin_A) + newmin_A$$ …Persamaan (2.1)

4. *Data mining*

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

5. *Interpretation/ Evaluation*

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 dengan *interpretation*. Tahap ini mencakup pemeriksaan apakah pola atau informasi yang ditemukan bertentangan dengan fakta atau hipotesa yang ada sebelumnya.

2.2.4 Tahap Siklus Data Mining

Tahap siklus hidup pengembangan data mining : [5]

1. *Business Understanding*

   Tahap pertama adalah memahami tujuan dan kebutuhan dari sudut pandang bisnis, kemudian menterjemakan pengetahuan ini ke dalam pendefinisian masalah dalam *data mining*. Selanjutnya akan ditentukan rencana dan strategi untuk mencapai tujuan tersebut.

2. *Data Understanding*

   Tahap ini dimulai dengan pengumpulan data yang kemudian akan dilanjutkan dengan proses untuk mendapatkan pemahaman yang mendalam tentang data, mengidentifikasi masalah kualitas data, atau untuk mendeteksi adanya bagian yang menarik dari data yang dapat digunakan untuk hipotesa untuk informasi yang tersembunyi.

3. *Data Preparation*

Tahap ini meliputi semua kegiatan untuk membangun dataset akhir (data yang akan diproses pada tahap pemodelan/*modeling*) dari data mentah. Tahap ini dapat diulang beberapa kali. Pada tahap ini juga mencakup pemilihan tabel, record, dan atribut-atribut data, termasuh proses pembersihan dan transformasi data untuk kemudian dijadikan masukan dalam tahap pemodelan (*modeling*).

4. *Modeling*

Dalam tahap ini akan dilakukan pemilihan dan penerapan berbagai teknik pemodelan dan beberapa parameternya akan disesuaikan untuk mendapatkan nilai yang optimal. Secara khusus, ada beberapa teknik berbeda yang dapat diterapkan untuk masalah data mining yang sama. Di pihak lain ada teknik pemodelan yang membutuhan format data khusus. Sehingga pada tahap ini masih memungkinan kembali ke tahap sebelumnya.

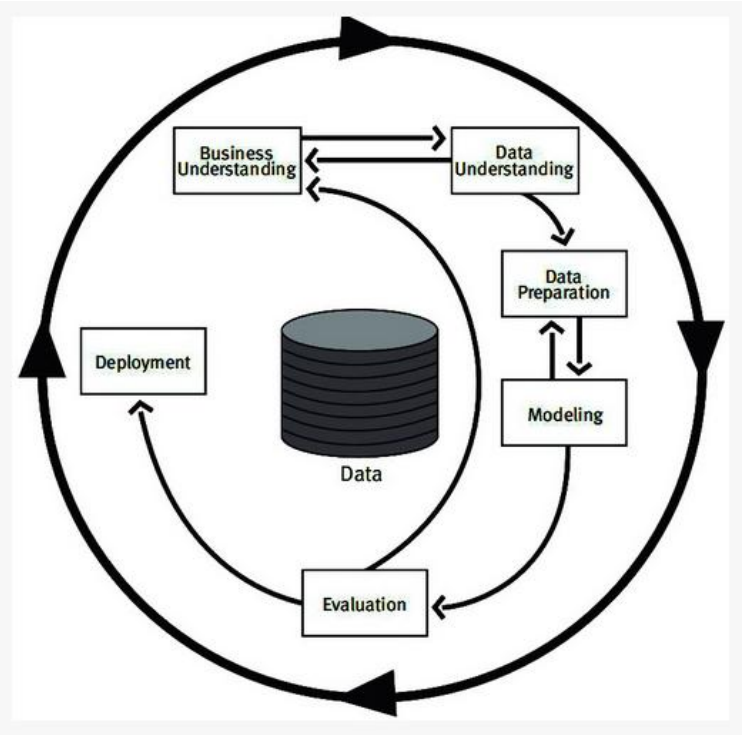
5. *Evaluation*

Pada tahap ini, model sudah terbentuk dan diharapkan memiliki kualitas baik jika dilihat dari sudut pandang analisa data. Pada tahap ini akan dilakukan evaluasi terhadap keefektifan dan kualitas model sebelum digunakan dan menentukan apakah model dapat mencapat tujuan yang ditetapkan pada fase awal (*Business Understanding*). Kunci dari tahap ini adalah menentukan apakah ada masalah bisnis yang belum dipertimbangkan. Di akhir dari tahap ini harus ditentukan penggunaan hasil proses data mining.

6. *Deployment*

Pada tahap ini, pengetahuan atau informasi yang telah diperoleh akan diatur dan dipresentasikan dalam bentuk khusus sehingga dapat digunakan oleh pengguna. Tahap *deployment* dapat berupa pembuatan laporan sederhana atau mengimplementasikan proses *data mining* yang berulang dalam perusahaan. Dalam banyak kasus, tahap deployment melibatkan konsumen, di samping analis data, karena sangat penting bagi konsumen untuk memahami

tindakan apa yang harus dilakukan untuk menggunakan model yang telah dibuat.



Gambar 2. 2 Siklus hidup pengembangan *data mining*

2.2.5. *Clustering* (Pengelompokan)

Clustering melakukan pengelompokan data tanpa adanya kelas data tertentu, sehingga dilakukan pembagian seluruh data menjadi kelompok-kelompok yang memiliki kemiripan atribut. Pengklusteran merupakan pengelompokan *record*, pengamatan, atau memperhatikan dan membentuk kelas objek-objek yang memiliki kemiripan. [1]

Analisa *cluster* yaitu menemukan kumpulan objek hingga objek-objek dalam satu kelompok sama (atau punya hubungan) dengan yang lain dan berbeda (atau tidak berhubungan) dengan objek-objek dalam kelompok lain. Tujuan dari

analisa *cluster* adalah meminimalkan jarak di dalam cluster dan memaksimalkan jarak antar *cluster*. [2]

Tujuan utama dari metode klaster adalah pengelompokan sejumlah data/objek ke dalam klaster(*group*) sehingga dalam setiap klaster akan berisi data yang semirip mungkin. Dalam *clustering* kita berusaha untuk menempatkan obyek yang mirip (jaraknya dekat) dalam satu klaster dan membuat jarak antar klaster sejauh mungkin.

2.2.6. **Klastering Hirarki (Hierarchical Clustering)**

Dalam klastering hirarki kita hitung jarak amsing-masing obyek dengan setiap obyek yang lain. Selanjutnya kita temukan pasangan obyek yang jaraknya terdekat. Sehingga setiap obyek akan berpasangan dengan satu obyek atau kelompok obyek yang lain yang paling dekat jaraknya. Langkah-langkah yang perlu dilakukan untuk melakukan klastering dengan cara klastering hirarki adalah: [2]

1. Kelompokkan setiap obyek ke dalam kelompok/ klasternya sendiri.
2. Temuakan pasangan paling mirip untuk dimasukan ke dalam klaster yang sama dengan melihat data dalam matriks kemiripan (*resemblance*).
3. Gabungkan kedua obyek dalam satu klaster.
4. Ulangi sampai tersisa hannya satu klaster.

2.2.7. **Tipe Clustering**

Clustering merupakan suatu kumpulan dari keseluruhan *cluster*. Beberapa tipe penting dari *clusteing* adalah : [3]

1. *Partitonal vs Hierarchical*

   Partitonal clustering adalah pembagian objek data ke dalam subhimpunan (*cluster*) yang tidak *overlap* sedemikian hingga tiap objek data berada dalam tepat satu sub-himpuanan. *Hierarchical clustering* meruapakan sebuah himpunan *cluster* bersarang yang diatur sebagai suatu pohon hirarki. Tiap simpul (*cluster*) dalam pohon

(kecuali simpul daun) merupakan gabungan anaknya (*subcluster*) dan simpul akar berisi semua objek.

2. *Exclusive vs non-exclusive*

   Semua bentuk *clustering* merupakan *exclusive clustering*, karena setiap objek berada tepat di satu *cluster*. Sebaliknya dalam *overlapping* atau *non-exclusive clustering*, sebuah objek dapat berada dilebih dari satu *cluster* secara bersamaan.

3. *Fuzzy vs non-fuzzy*

   Dalam *fuzzyclustering*, sebuah titik termasuk dalam setiap *cluster* dengan suatu nilai bobot antara 0 dan 1. Jumlah dari bobot-bobot tersebut sama dengan 1. *Clustering probabilitas* mempunyai karakteristik yang sama.

4. *Partial vs complete*

   Dalam *completeclustering*, setiap objek ditempatkan dalam sebuah *cluster*. Tetapi dalam *partial clustering*, tidak semua objek ditempatkan dalam sebuah *cluster*. Kemungkinan ada objek yang tidak tepat untuk ditempatkan di salah satu *cluster*, misalkan berupa *outlier* atau *noise.*

2.2.8. **Algoritma K-Means *Clustering***

Menggunakan pendekatan *partitional clustering*. Tiap *cluster* dihubungkan dengan sebuah *centroid* (titik pusat). Tiap titik ditempatkan ke dalam *cluster* dengan *centroid* terdekat. Jumlah *cluster*, K, harus ditentukan. Algoritma dasarnya sangat sederhana, yaitu : [2]

1. Menentukan banyaknya *cluster* (k).
2. Menentukan *centroid.*
3. Apakah *centroid* tersebut berubah?
   a. Jika ya, hitung jarak data dari *centroid*.
   b. Jika tidak, selesai.
4. Menghitung jarak dari centroid

Rumus yang digunakan untuk menghitung jarak dari centroid adalah rumus *Euclidean Distance* :

$$d(x_j, c_j) = \sqrt{\sum_{j=1}^{n}(x_j - c_j)^2} \quad \text{… Persamaan (2.2)}$$

Ket :

d = jarak

j = banyaknya data

c = centroid

x = data

5. Mengelompokan data berdasarkan jarak yang terdekat.

   Rumus untuk memperbaharui centroid dapat dilihat pada persamaan 2.3 sebagai berikut :

$$\mu_k = \frac{1}{N_k}\sum_{q=1}^{N_k} x_q \quad \text{……Persamaan (2.3)}$$

   Ket :

   μ_k = Titik centroid dari cluster ke-K

   N_k = Banyaknya data pada cluster ke-K

   x_q = Data ke-q pada cluster ke-K

Algoritma K-Means akan menghasilkan kelompok catatan sebangak *k* buah. Algoritma K-Means pertama kali digegas oleh J. MacQueen. Langkah-langkah pada algoritma K-Means, adalah sebagai berikut :

1. Langkah pertama: tanyakan kepada pemakai *algoritma k-means*, catatan-catatan yang ada akan dibuat menjadi beberapa kelompok, sebutlah sebanyak *k* kelompok.
2. Langkah kedua: secara sembarangan, pilihlah *k* buah catatan (dari sekian catatan yang ada) sebagai pusat-pusat kelompok awal.

3. Langkah ketiga: untuk setiap catatan, tentukan pusat kelompok terdekatnya dan tetapkan catatan tersebut sebagai anggota dari kelompok yang terdekat pusat kelompoknya. Hitung rasio antara besaran *Between Cluster Variation* dengan *Within Cluster Variation*, lalu bandingkan rasio tersebut dengan rasio sebelumnya (bila sudah ada). Jika rasio tersebut membesar, lanjutkan ke langkah keempat. Jika tidak, hentikan proses.
4. Langkah keempat: perbarui pusat-pusat kelompok (berdasarkan kelompok yang didapat dari langkah ketiga) dan kembalilah ke langkah ketiga.

2.2.9. **Basis Data (*Database*)**

Basis data (*Database*) adalah sekumpulan informasi bermanfaat yang diorganisasikan kedalam tata cara yang khusus. [6]

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: [6]

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*).

2.2.10. **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". [6]

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*

   Hardware merupakan sistem komputer aktual yang digunakan untuk menyimpan dan mengakses *databse*. 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. Prosedur

   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

   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. *User* (Pengguna)

   Ada sejumlah u*ser* 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 *workstatio*n dan berinteraksi secara langsung dengan sistem.
   c. *Programmer aplikasi*, orang yang berinteraksi dengan *database* melalui cara yang berbeda.

2.2.11. **ERD (Entity Relationship Diagram)**

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: [6]

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, biasanya 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).*

2.2.12. **DFD (Data Flow Diagram)**

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. [6]

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 :

1. Kebebasan dari menjalankan implementasi teknis sistem.

2. Pemahaman lebih jauh mengenai keterkaitan satu sama lain dalam sistem dan subsistem.
3. Mengkomunikasikan pengetahuan sistem yang ada dengan
4. pengguna melalui diagram aliran data.
5. Menganalisis sistem yang diajukan untuk menentukan apakah datadata dan proses yang diperlukan sudah ditetapkan.

Disamping itu terdapat kelebihan tambahan, yaitu :

1. Dapat digunakan sebagai latihan yang bermanfaat bagi penganalisis, sehingga bisa memahami dengan lebih baik
2. keterkaitan satu sama lain dalam sistem dan subsistem.
3. Membedakan sistem dari lingkungannya dengan menempatkan
4. batas-batasnya.
5. Dapat digunakan sebagai suatu perangkat untuk berinteraksi dengan pengguna.
6. 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.

2.2.13. **MySQL**

MySQL merupakan aplikasi Relation Database Management Server (RDBMS) berbasis Open Source yang paling popular karena beberapa keunggulannya seperti kecepatan yang konsisten, keandalan tinggi, dapat berjalan diberbagai system operasi, kemudahan dan kenyamanan dalam menggunakannya serta dapat dipergunakan oleh banyak pemakai secara bersamaan dan dapat dibatasi pemakaiannya berdasarkan previlage (hak akses). [4]

MySQL menggunakan bahasa SQL *(Structured Query language)* yaitu bahasa standart internasional dalam penggunaan pemrograman database. Berdasarkan riset dinyatakan bahwa di platform Web, dan baik untuk kategori open source maupun umum,

MySQL adalah database yang paling banyak dipakai. Dan akan perkembangannya, MySQL telah terpasang di sekitar 3 juta komputer. Puluhan hingga ratusan ribu situs mengandalkan MySQL bekerja siang malam memompa data bagi para pengunjungnya. MySQL Kepopuleran MySQL disebabkan karena menggunakan SQL sebagai bahasa dasar untuk mengakses databasenya. Pada MySQL, sebuah database mengandung satu atau sejumlah tabel. Tabel terdiri atas sejumlah batas sejumlah baris dan setiap baris mengandung satu atau beberapa kolom. [4]

2.2.14. **C Sharp (C#)**

Bahasa C# (dibaca C sharp) adalah suatu asel yang terdapat dalam .NET Framework. Anda dapat menggunakan C# untuk membuat file yang berdiri sendiri atau untuk membuat isi alau content dinamis dalam sebuah web form.

C# dikenal sebagai visual C# dalam Visual Studio .Net. Dukungan untuk Visual C# termasuk proyek template, desainer, halaman poperti, kode, model

objek dan fitur lain dari lingkungan pengembangan. Library untuk pemrograman visual C# adalah .NET Framework. Visual C# dibuat berdasarkan pemrograman C# yang merupakan bahasa pemrograman berorientasi objek dan mempunyai banyak kesamaan dengan C++, Java, dan VB. C# pada faktanya merupakan kombinasi antara efisiensi pemrograman C++, kesederhanaan pemrograman Java, dan penyederhanaan dari pemrograman Visual Basic. [5]