

BAB II

LANDASAN TEORI

II.1. Sistem Pendukung Keputusan

Konsep Sistem Pendukung Keputusan pertama kali diperkenalkan pada awal tahun 1970-an oleh Michael S. Scott Morton dengan istilah *Management Decision System* (Sprague, 1982). Konsep pendukung keputusan ditandai dengan sistem interaktif berbasis komputer yang membantu pengambil keputusan memanfaatkan data dan model untuk menyelesaikan masalah-masalah yang tidak terstruktur. Pada dasarnya SPK dirancang untuk mendukung seluruh tahap pengambilan keputusan mulai dari mengidentifikasi masalah, memilih data yang relevan, menentukan pendekatan yang digunakan dalam proses pengambilan keputusan, sampai mengevaluasi pemilihan alternatif. (Hilya Magdalena, 2012 : 50).

Pengertian Sistem Pendukung Keputusan menurut para ahli sebagai berikut (Tri Susilowati, 2015 : 14) :

1. Menurut Turban dkk.. (2005), “*SPK adalah pendekatan berbasis komputer atau metodologi untuk mendukung pengambilan keputusan*”.
2. Menurut Moore and Chang (2011) “*SPK adalah suatu sistem yang dirancang untuk mengkomunikasikan masalah dan menyelesaikan pemecahan masalah yang dilakukan manajer bersifat semi struktur yang spesifik untuk mengambil suatu keputusan*”.

3. Menurut Wibowo (2011) *“SPK adalah proses pengambilan keputusan dibantu menggunakan komputer untuk membantu pengambil keputusan dengan menggunakan beberapa data dan model tertentu untuk menyelesaikan beberapa masalah yang tidak terstruktur”*.

Dari pendapat yang dikemukakan di atas dapat disimpulkan bahwa Sistem Pendukung Keputusan (SPK) adalah suatu sistem informasi berbasis komputer yang menghasilkan berbagai alternatif keputusan untuk membantu manajemen dalam menangani berbagai permasalahan yang terstruktur ataupun tidak terstruktur dengan menggunakan data dan model.

II.2.1. Karakteristik dan Keterbatasan Sistem Pendukung Keputusan

SPK, menurut tinjauan konotatif, merupakan system yang ditujukan kepada tingkatan manajemen yang lebih tinggi, dengan penekanan karakteristik sebagai berikut:

1. Berfokus pada keputusan., ditujukan pada manajer puncak dan pengambil keputusan.
2. Menekankan pada fleksibilitas, adaptabilitas, dan respon yang cepat.
3. Mampu mendukung berbagai gaya pengambilan keputusan dan masing-masing pribadi manajer. (Hilya Magdalena, 2012 : 51).

Adapun keterbatasan sistem pendukung keputusan menurut Hilya Magdalena (2012 : 51) adalah sebagai berikut :

1. Adanya gambaran bahwa SPK seakan-akan hanya dibutuhkan pada tingkat manajemen puncak. Pada kenyataannya, dukungan bagi

pengambilan keputusan dibutuhkan pada semua tingkatan manajemen dalam suatu organisasi.

2. Pengambilan keputusan yang terjadi pada beberapa level harus dikoordinasikan. Jadi, dimensi dan pendukung keputusan adalah komunikasi dan koordinasi diantara pengambil keputusan antar level organisasi yang berbeda maupun pada level organisasi yang sama.

II.2. *Analytical Hierarchy Process (AHP)*

AHP dikembangkan pada tahun 1970an oleh Dr Thomas L. Satty untuk menyediakan pendekatan sistematis untuk menentukan prioritas dan pengambilan keputusan dalam suatu kompleks lingkungan. AHP dirancang untuk mencerminkan cara berpikir orang sebenarnya. Metode ini memungkinkan aspek kuantitatif dan kualitatif keputusan yang akan dipertimbangkan. AHP mengurangi keputusan yang kompleks menjadi sebuah rangkaian satu-satu pada perbandingan yang kemudian memberikan hasil yang akurat. AHP juga menggunakan skala rasio untuk bobot kriteria dan scoring alternatif yang menambahkan untuk pengukuran presisi. (Hilya Magdalena, 2012 : 51).

Karena sulitnya menentukan bobot-bobot ataupun prioritas-prioritas yang sering berubah-ubah, digunakan perbandingan berpasangan yang menggunakan data, pengetahuan, dan pengalaman membuat penilaian berkenaan dengan pertimbangan relatif pentingnya satu elemen terhadap yang lain. Untuk itu diperlukan suatu skala perbandingan antar dua elemen, baik secara kualitatif maupun kuantitatif. (Hilya Magdalena, 2012 : 52).

Untuk kegiatan perbandingan antar sepasang objek, metode AHP memberikan sebuah standar nilai perbandingan antar dua objek seperti dituangkan pada tabel II.1.

Tabel II.1. Nilai Perbandingan

Pembandingan	Nilai
Sangat diutamakan	9
Lebih diutamakan menuju sangat diutamakan	8
Lebih diutamakan	7
Diutamakan menuju lebih diutamakan	6
Diutamakan	5
Cukup diutamakan menuju diutamakan	4
Cukup diutamakan	3
Setara menuju cukup diutamakan	2
Setara	1

(Sumber : Hilya Magdalena, 2012)

II.3.1. Proses-proses dalam AHP

Adapun proses-proses yang harus dilakukan pada metode AHP adalah sebagai berikut (I Nyoman Yudha Astana, 2013) :

1. Mendefinisikan masalah dan menentukan solusi yang diinginkan.
2. Membuat struktur hirarki yang diawali tujuan umum dilanjutkan dengan kriteria dan kemungkinan alternatif pada tingkatan kriteria paling bawah.

3. Membuat matrik perbandingan berpasangan yang menggambarkan kontribusi relatif atau pengaruh setiap elemen terhadap kriteria yang setingkat di atasnya.
4. Melakukan perbandingan berpasangan sehingga diperoleh *judgment* (keputusan) sebanyak $n \times ((n-1)/2)$ kali, dengan n adalah banyaknya elemen yang dibandingkan.
5. Menghitung nilai *eigen* dan menguji konsistensinya jika tidak konsisten maka pengambilan data diulangi lagi.
6. Mengulangi langkah 3,4 dan 5 untuk setiap tingkatan hirarki.
7. Menghitung *vector eigen* dari setiap matrik perbandingan berpasangan.
8. Memeriksa konsistensi hirarki. Jika nilainya lebih dari 10 persen maka penilaian data *judgment* harus diperbaiki.

II.4. Basis Data

Basis data sendiri dapat didefinisikan dalam sejumlah sudut pandang seperti (Akhmad Sholikhin, 2013 : 51) :

1. Himpunan kelompok data (arsip) yang saling lberhubungan yang diorganisasikan sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.
2. Kumpulan data yang saling berhubungan disimpan secara bersama sedemikian rupa dan tanpa pengulangan yang yang perlu untuk memenuhi berbagai kebutuhan.

3. Kumpulan *file* atau tabel atau arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronis.

Berdasarkan tingkat kompleksitas nilai data, tingkatan data dapat disusun dalam sebuah hierarki, mulai dari yang paling sederhana hingga paling sederhana hingga paling kompleks.

1. Sistem basis data, merupakan sekumpulan subsistem yang terdiri atas basis data dengan para pemakai yang menggunakan basis data secara bersama-sama, personal-personal yang merancang dan mengelola basis data, teknik-teknik untuk merancang dan mengelola basis data, serta sistem komputer untuk mendukungnya.
2. Basis data, merupakan sekumpulan dari bermacam-macam tipe *record* yang memiliki hubungan antar-*record* dan rincian data terhadap obyek tertentu.
3. File, merupakan sekumpulan *record* sejenis secara relasi yang tersimpan dalam media penyimpanan sekunder.
4. *Record*, merupakan *field*/atribut/data item yang saling berhubungan terhadap obyek tertentu.
5. *Data item/field*/atribut, merupakan unit terkecil yang disebut data, sekumpulan *byte* yang mempunyai makna.
6. *Data aggregate*, merupakan sekumpulan data item/*field*/atribut dengan ciri tertentu dan diberi nama.
7. *Byte*, adalah bagian terkecil yang dialamatkan dalam memori. *Byte* merupakan sekumpulan *bit* yang secara konvensional terdiri atas

kombinasi 8 *bit* biner yang menyatakan sebuah karakter dalam memori (1 *byte* = 1 karakter).

8. *Bit*, adalah sistem biner yang terdiri atas dua macam nilai, yaitu 0 dan 1. Sistem biner merupakan dasar yang dapat digunakan untuk komunikasi antara manusia dan mesin (komputer). (Edy Sutanta, 2011 : 35-36).

II.5. Kamus Data

Kamus Data adalah katalog fakta tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi agar pengguna sistem dapat mendefinisikan data yang mengalir di sistem dengan lengkap. (Sudibyo, 2014 :71).

Menurut Roger. S. Pressman kamus data adalah sebuah daftar yang terorganisasi dari elemen data yang berhubungan dengan system, dengan definisi yang tegas dan teliti, sehingga pemakai dan analisis system akan memiliki pemahaman yang umum mengenai input, output, dan komponen penyimpanan dan bahkan kalkulasi intermediate. (Syahrani Dhimas Prabowo, 2013 : 30).

II.6. Normalisasi

Normalisasi diartikan sebagai suatu teknik yang menstrukturkan/mendekomposisi data dalam cara-cara tertentu untuk mencegah timbulnya permasalahan pengolahan data dalam basis data. Permasalahan yang dimaksud adalah berkaitan dengan penyimpangan-penyimpangan (*anomalies*)

yang terjadi akibat adanya kerangkapan data dalam relasi dan in-efisiensi pengolahan. (Edy Sutanta ; 2011 : 174)

Proses normalisasi menghasilkan relasi yang optimal, yaitu (Martin, 1975)
: (Edy Sutanta ; 2011 : 175)

1. Memiliki struktur *record* yang konsisten secara logik;
2. Memiliki struktur *record* yang mudah untuk dimengerti;
3. Memiliki struktur *record* yang sederhana dalam pemeliharaan;
4. Memiliki struktur *record* yang mudah ditampilkan kembali untuk memenuhi kebutuhan pengguna;
5. Minimalisasi kerangkapan data guna meningkatkan kinerja sistem.

Secara berturut-turut masing-masing level normal tersebut dibahas berikut ini, dimulai dari bentuk tidak normal. (Edy Sutanta ; 2011 : 176-179)

1. Relasi bentuk tidak normal (*Un Normalized Form* / UNF)

Relasi-relasi yang dirancang tanpa mengindahkan batasan dalam defisi basis data dan karakteristik *Relational Database Management System* (RDBM) menghasilkan relasi *Un Normalized Form* (UNF). Bentuk ini harus di hindari dalam perancangan relasi dalam basis data. Relasi *Un Normalized Form* (UNF) mempunyai kriteria sebagai berikut.

- a. Jika relasi mempunyai bentuk *non flat file* (dapat terjadi akibat data disimpan sesuai dengan kedatangannya, tidak memiliki struktur tertentu, terjadi duplikasi atau tidak lengkap)
- b. Jika relasi membuat *set atribut* berulang (*non single values*)

c. Jika relasi membuat *atribut non atomic value*

2. Relasi bentuk normal pertama (*First Norm Form / 1NF*)

Relasi disebut juga *First Norm Form (1NF)* jika memenuhi kriteria sebagai berikut.

- a. Jika seluruh atribut dalam relasi bernilai *atomic (atomic value)*
- b. Jika seluruh atribut dalam relasi bernilai tunggal (*single value*)
- c. Jika relasi tidak memuat set atribut berulang
- d. Jika semua record mempunyai sejumlah atribut yang sama.

Permasalahan dalam *First Norm Form (1NF)* adalah sebagai berikut.

- a. Tidak dapat menyisipkan informasi parsial
- b. Terhapusnya informasi ketika menghapus sebuah *record*

3. Bentuk normal kedua (*Second Normal Form / 2NF*)

Relasi disebut sebagai *Second Normal Form (2NF)* jika memenuhi kriteria sebagai berikut

- a. Jika memenuhi kriteria *First Norm Form (1NF)*
- b. Jika semua atribut nonkunci *Functional Dependence (FD)* pada *Primary Key (PK)*

Permasalahan dalam *Second Normal Form / 2NF* adalah sebagai berikut

- a. Kerangkapan data (*data redundancy*)
- b. Pembaharuan yang tidak benar dapat menimbulkan inkonsistensi data (*data inconsistency*)
- c. Proses pembaharuan data tidak efisien

Kriteria tersebut mengidentifikasi bahwa antara atribut dalam *Second Normal Form* masih mungkin mengalami *Third Norm Form*. Selain itu, relasi *Second Normal Form* (2NF) menuntut telah didefinisikan atribut *Primary Key* (PK) dalam relasi. Mengubah relasi *First Norm Form* (1NF) menjadi bentuk *Second Normal Form* (2NF) dapat dilakukan dengan mengubah struktur relasi dengan cara :

- a. Identifikasikan *Functional Dependence* (FD) relasi *First Norm Form* (1NF)
 - b. Berdasarkan informasi tersebut, dekomposisi relasi *First Norm Form* (1NF) menjadi relasi-relasi baru sesuai *Functional Dependence* nya. Jika menggunakan diagram maka simpul-simpul yang berada pada puncak diagram ketergantungan data bertindak *Primary Key* (PK) pada relasi baru
4. Bentuk normal ketiga (*Third Norm Form* / 3NF)

Suatu relasi disebut sebagai *Third Norm Form* jika memenuhi kriteria sebagai berikut.

- a. Jika memenuhi kriteria *Second Normal Form* (2NF)
- b. Jika setiap atribut nonkunci tidak (TDF) (*Non Transitive Dependency*) terhadap *Primary Key* (PK)

Permasalahan dalam *Third Norm Form* (3NF) adalah keberadaan penentu yang tidak merupakan bagian dari *Primary Key* (PK) menghasilkan duplikasi rinci data pada atribut yang berfungsi sebagai *Foreign Key* (FK) (duplikasi berbeda dengan keterangan data).

Mengubah relasi *Second Normal Form* (2NF) menjadi bentuk *Third Norm Form* (3NF) dapat dilakukan dengan mengubah struktur relasi dengan cara :

- a. Identifikasi TDF relasi *Second Normal Form* (2NF)
- b. Berdasarkan informasi tersebut, dekomposisi relasi *Second Normal Form* (2NF) menjadi relasi-relasi baru sesuai TDF-nya.

5. Bentuk normal *Boyce-Cood*(*Boyce-Codd Norm Form* / BCNF)

Bentuk normal *Boyce-Codd Norm Form* (BCNF) dikemukakan oleh R.F. Boyce dan E.F. Codd. Suatu relasi disebut sebagai *Boyce-Codd Norm Form* (BCNF) jika memenuhi kriteria sebagai berikut.

- a. Jika memenuhi kriteria *Third Norm Form* (3NF)
- b. Jika semua atribut penentu (determinan) merupakan CK

6. Bentuk normal keempat (*Forth Norm Form* / 4NF)

Relasi disebut sebagi *Forth Norm Form* (4NF) jika memenuhi kriteria sebagai berikut.

- a. Jika memenuhi kriteria *Boyce-Codd Norm Form*.
- b. Jika setiap atribut didalamnya tidak mengalami ketergantungan pada banyak nilai.

7. Bentuk normal kelima (*Fifth Norm Form* / 5NF)

Suatu relasi memenuhi kriteria *Fifth Norm Form* (5NF) jika kerelasian antar data dalam relasi tersebut tidak dapat direkonstruksi dari struktur relasi yang sederhana.

8. Bentuk normal kunci domain (*Domain Key Norm Form* / DKNF)

Relasi disebut sebagai *Domain Key Norm Form* (DKNF) jika setiap batasan dapat disimpulkan secara sederhana dengan mengetahui sekumpulan nama atribut dan domainnya selama menggunakan sekumpulan atribut pada kuncinya.

II.7. *Unified Modeling Language* (UML)

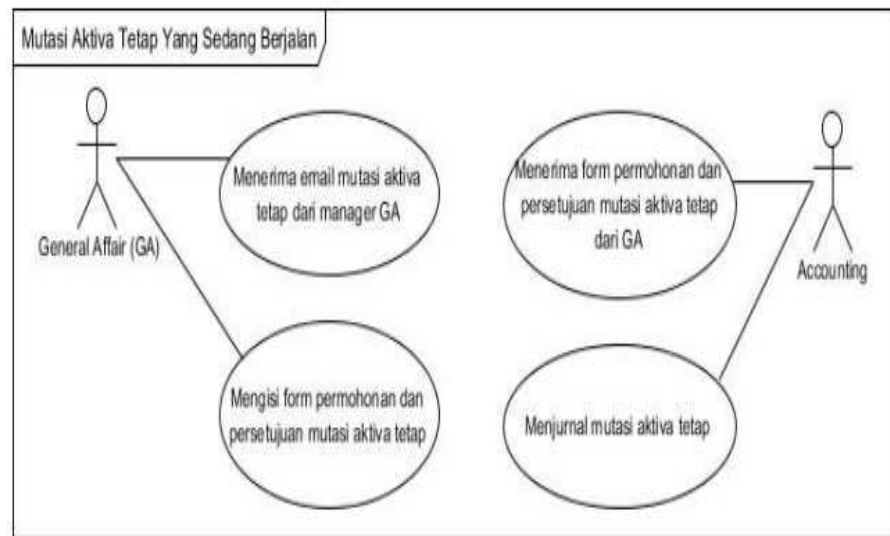
UML adalah salah satu alat bantu yang sangat handal di dunia pengembangan sistem yang berorientasi objek. UML dapat menyediakan bahasa pemodelan yang mudah dimengerti oleh pengembang dan dapat dikomunikasikan dengan pemakai. Metode perancangan yang digunakan pada penelitian ini adalah metode berorientasi objek menggunakan UML. (Nyimas Sopiah, 2012 : 188).

UML (*Unified Modelling Language*) juga merupakan sebuah bahasa yang sudah menjadi standar di dunia industri untuk visualisasi, merancang dan mendokumentasikan system piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah system. (Sentosa Pohan, 2015 : 14).

II.7.1. *Artifact* dan Notasi UML

1. Diagram Use Case

Use case menjelaskan urutan kegiatan yang dilakukan actor dan sistem untuk mencapai suatu tujuan tertentu. Walaupun menjelaskan kegiatan, namun use case hanya menjelaskan apa yang dilakukan oleh actor dan sistem bukan bagaimana actor dan sistem melakukan kegiatan tersebut.

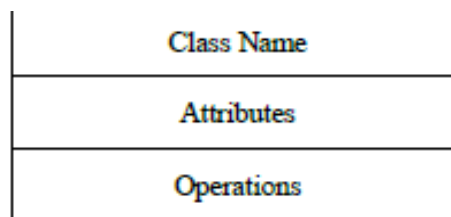


Gambar II.1. Use Case Diagram

(Sumber : Rosana Junita Sirait, et al., 2015)

2. Class Diagram

Menggambarakan struktur statis *class* di dalam sistem. *Class* merepresentasikan sesuatu yang ditangani oleh sistem. *Class* dapat berhubungan dengan yang lain melalui berbagai cara: *associated* (terhubung satu sama lain), *dependent* (satu *class* tergantung/menggunakan *class* yang lain), *specialized* (satu *class* merupakan spesialisasi dari *class* lainnya), atau *package* (grup bersama sebagai satu unit). Sebuah sistem biasanya mempunyai beberapa *class diagram*. Bentuk umum dari *class diagram* dapat dilihat pada gambar berikut :

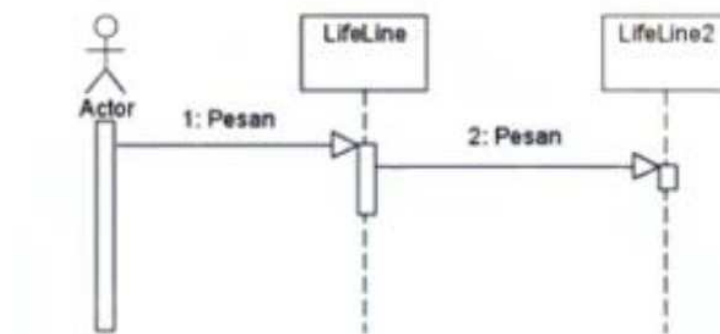


Gambar II.2. Bentuk Umum Class Diagram

(Sumber : Sentosa Pohan, 2015 : 14)

3. *Sequence diagram*

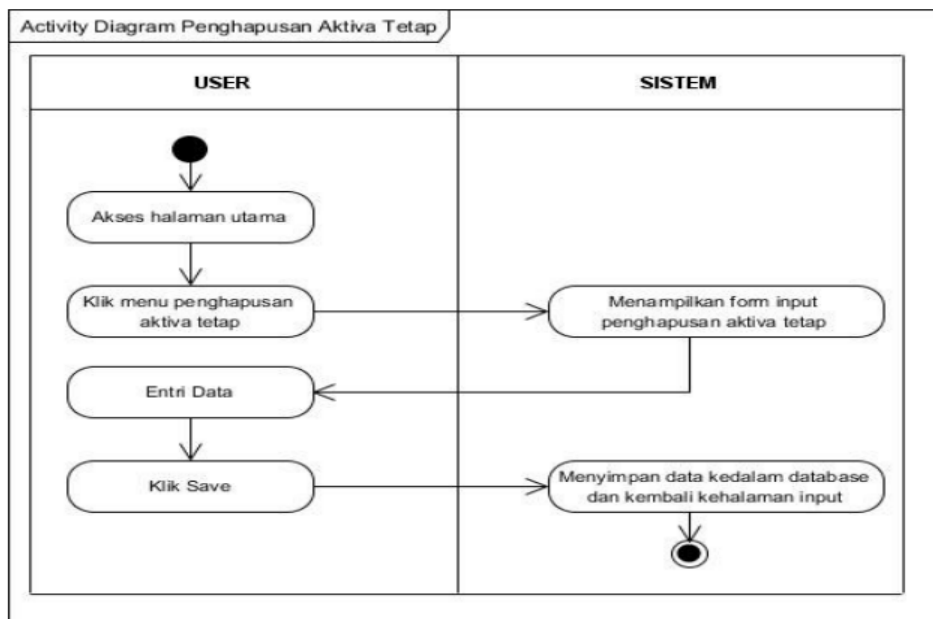
Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait).



Gambar II.3. Bentuk Umum *Sequence diagram*
(Sumber : Sentosa Pohan, 2015 : 14)

4. *Activity Diagram*

Menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktifitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktifitas lainnya seperti *use case* atau interaksi.



Gambar II.4. Activity Diagram
(Sumber : Rosana Junita Sirait, et al., 2015)

II.8. PHP

PHP singkatan dari PHP: *Hypertext Preprocessor* yang digunakan sebagai bahasa *script serverside* dalam pengembangan web yang disisipkan dalam dokumen HTML. Penggunaan PHP memungkinkan web dapat dinamis sehingga maintenance situs web tersebut menjadi lebih mudah dan efisien. PHP merupakan software Open-source yang disebar dan dilisensikan secara gratis serta dapat di download secara bebas dari situs resminya <http://www.php.net>. (Aan Tri Wibowo, 2013 : 63).

PHP adalah suatu bahasa pemrograman *Open Source* yang digunakan secara luas terutama untuk pengembangan web dan dapat disimpan dalam bentuk HTML. Keuntungan utama menggunakan PHP adalah script PHP tidak benarbenar sederhana bagi pemula, tetapi menyediakan banyak fitur tambahan untuk

programmer professional. Meskipun PHP lebih difokuskan sebagai script Server Side. Script PHP dapat digunakan dalam 3 hal, yaitu:

1. Penulisan program *Server Side*. Hal ini adalah target utama PHP. Diperlukan tiga hal agar script PHP dapat bekerja antara lain, PHP *Parser* (CGI atau *Server module*), server web dan browser web. menjalankan server web terlebih dahulu, kemudian mengakses keluaran program PHP melalui browser web dan melihat halaman web.
2. Penulisan program *Command Line*. Script PHP dapat berjalan tanpa server atau browser. Hanya diperlukan PHP *Parser* dalam bentuk *Command Line*.
3. Penulisan program aplikasi desktop. (Aan Tri Wibowo, 2013 : 63-64)

II.9. *MYSQL*

MySQL merupakan *software* RDBMS (atau *server database*) yang dapat mengelola *database* dengan sangat cepat, dapat menampung data dalam jumlah sangat besar, dapat diakses oleh banyak user (*multi-user*), dan dapat melakukan suatu proses secara sinkron atau bebarengan (*multi-threaded*).

Lisensi MySQL terbagi menjadi dua yaitu produk *open source* di bawah GNU (*General Public License*) atau dapat membeli lisensi dari versi komersialnya. MySQL versi komersial tentu memiliki nilai lebih atau kemampuan-kemampuan yang tidak disertakan pada versi gratis. Pada kenyataannya, untuk

keperluan industri menengah kebawah, versi gratis masih dapat digunakan dengan baik. (Budi Raharjo, 2012 : 21 - 22).