

BAB II

TINJAUAN PUSTAKA

II.1. Sistem Pakar

II.1.1. Pengertian Sistem Pakar

Sistem pakar merupakan program yang dapat menggantikan keberadaan seorang pakar. Alasan dasar mengapa ES dikembangkan untuk menggantikan seorang pakar :

1. Dapat menyediakan kepakaran setiap waktu dan diberbagai lokasi.
2. Secara Otomatis mengerjakan tugas-tugas rutin yang membutuhkan seorang pakar.
3. Seorang pakar akan pensiun atau pergi.
4. Menghadirkan/menggunakan jasa seorang pakar memerlukan biaya yang mahal.
5. Kepakaran dibutuhkan juga pada lingkungan yang tidak bersahabat (*hostile environment*). (Kusrini ; 2006 : 14).

Sedangkan menurut Andi Offset,dkk (2009 : 3) kecerdasan buatan adalah sistem pakar yang menggabungkan pengetahuan dan penelusuran data untuk memecahkan masalah yang secara normal memerlukan keahlian manusia. Tujuan pengembangan sistem pakar sebenarnya bukan untuk menggantikan peran manusia, tetapi untuk mendistribusikan pengetahuan manusia ke dalam bentuk sistem, sehingga dapat digunakan oleh orang banyak.

Dari uraian di atas dapat disimpulkan definisi sistem pakar secara umum adalah sebuah perangkat lunak yang merupakan bagian dari kecerdasan buatan yang mengimplementasikan pengetahuan seorang pakar ke dalam sebuah sistem komputer yang berbasis pengetahuan. Sistem pakar menjadi suatu kategori yang lebih luas dari program-program yang dikenal sebagai sistem berbasis pengetahuan dan merupakan suatu program dasar yang menyediakan suatu keahlian tentang suatu masalah dalam suatu bidang khusus.

II.1.2. Ciri-Ciri Sistem Pakar

Secara umum sistem pakar mempunyai beberapa karakteristik, antara lain:

1. Terbatas pada bidang yang spesifik
2. Dapat memberikan penalaran untuk data-data yang tidak lengkap atau tidak pasti.
3. Dapat mengemukakan rangkaian alasan yang di berikannya dengan cara yang dapat dipahami.
4. Berdasarkan pada *rule* atau kaidah tertentu.
5. Dirancang untuk dapat di kembangkan secara bertahap.
6. Outputnya bersifat nasihat atau anjuran.
7. Outputnya tergantung dari dialog dengan *user*.
8. *Knowledge base* dan *inference engine* terpisah. (Kusrini ; 2006 : 14-15)

II.1.3. Komponen-Komponen Sistem Pakar

Sebuah program yang difungsikan untuk menirukan seorang pakar manusia harus bisa melakukan hal-hal yang dapat dikerjakan seorang pakar. Untuk membangun sistem yang seperti itu maka komponen-komponen dasar yang harus dimiliki paling sedikit adalah sebagai berikut :

1. Akuisisi Pengetahuan

Pada proses akuisisi pengetahuan dilakukan pengumpulan data-data pengetahuan yang merupakan masalah dari pakar yang akan dirancang. Bahan pengetahuan dapat didapat dari buku, jurnal yang telah diterbitkan, para pakar dibidangnya, ataupun laporan yang telah diperiksa kebenarannya. Contoh akuisisi pengetahuan dapat dilihat dari penulisan skripsi ini yang dimulai dengan mengumpulkan data tentang macam gangguan psikologi pada anak, gejala-gejalanya, sampai pada penanggulangannya.

2. Basis Pengetahuan dan Basis Aturan

Setelah proses akuisisi pengetahuan, maka pengetahuan tersebut harus dipresentasikan menjadi basis pengetahuan dan basis aturan yang selanjutnya dikumpulkan, dikodekan, dan diorganisasikan serta digambar dalam bentuk rancangan yang sistematis.

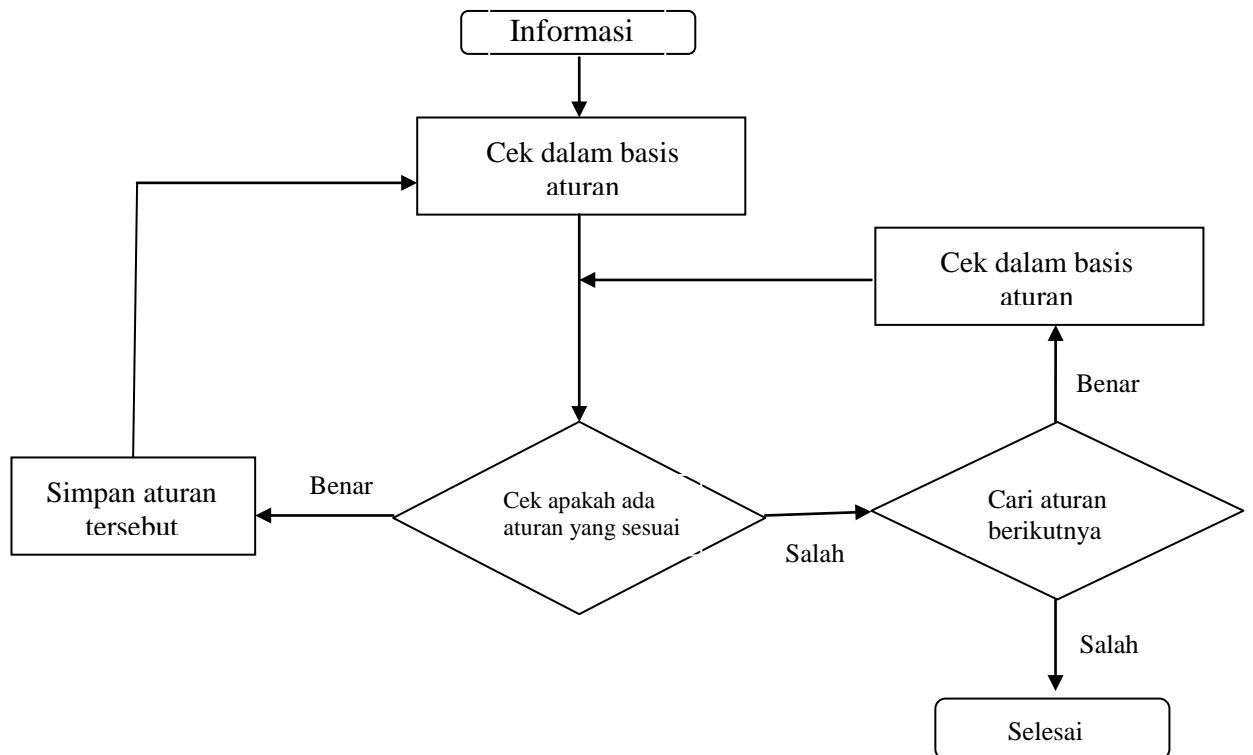
Ada beberapa cara merepresentasikan data menjadi basis pengetahuan, yaitu data dalam bentuk atribut, aturan-aturan, jaringan semantik, frame dan logika. Semua bentuk representasi tersebut bertujuan untuk menyederhanakan data sehingga mudah dipahami untuk pengembangan program.

3. Mekanisme Inferensi

Mekanisme inferensi adalah bagian dari sistem pakar yang melakukan penalaran dengan menggunakan isi daftar aturan berdasarkan urutan dan pola tertentu. Selama proses konsultasi antara sistem dan pemakai, mekanisme inferensi menguji aturan satu demi satu sampai kondisi aturan itu benar (Andi Offset ; 2009 : 14-15). Secara Umum, ada dua teknik yang digunakan dalam mekanisme inferensi untuk pengujian aturan, yaitu:

a. Penalaran Maju (*Forward Chaining*)

Forward Chaining merupakan pendekatan yang dimulai dari sekumpulan data menuju kesimpulan. Pada penalaran maju, aturan diuji satu demi satu dalam urutan tertentu. Saat tiap aturan diuji, sistem pakar akan mengevaluasi apakah kondisinya benar atau salah. Jika kondisinya benar, maka aturan itu disimpan kemudian aturan berikutnya diuji. Sebaliknya jika kondisinya salah, maka aturan itu tidak disimpan dan aturan berikutnya diuji. Proses ini kan berulang sampai seluruh basis aturan teruni dengan berbagai kondisi. Pada Gambar II.1 akan dijelaskan alur diagram untuk mekanisme inferensi penalaran maju.



Gambar II.1. Penalaran Maju

(Sumber : Andi Offset,dkk ; 2009 : 15)

Penalaran maju menggunakan himpunan aturan kondisi-aksi. Gambar II.2

berikut ini akan menjelaskan cara kerja inferensi *forward chaining*.

DATA → ATURAN → KESIMPULAN

A=1
B=1

Jika A=1 dan B=2
maka C=3
Jika C=3 maka D=4

D=4

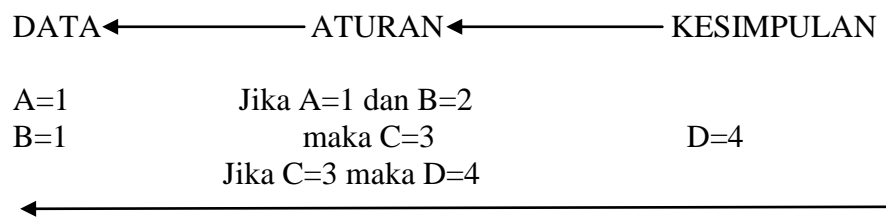
→

Gambar II.2. Cara Kerja Inferensi *Forward Chaining*

(Sumber : Kusrini ; 2006 : 36)

b. Penalaran Mundur (*Backward Chaining*)

Backward chaining memulai penalarannya dari sekumpulan hipotesa menuju fakta-fakta yang mendukung hipotesa tersebut. *Backward chaining* merupakan kebalikan dari *forward chaining*.



Gambar II.3. Cara Kerja Inferensi *Backward Chaining*

(Sumber : Kusrini ; 2006 : 37)

4. Fasilitas Penjelasan Sistem

Fasilitas penjas sistem merupakan bagian dari sistem pakar yang memberikan penjelasan tentang bagaimana proses dijalankan, apa yang harus dijelaskan kepada pemakai tentang suatu masalah, dan memberikan rekomendasi kepada pemakai. Dalam sistem pakar, fasilitas penjelasan sistem sebaiknya diintegrasikan ke dalam tabel basis pengetahuan dan basis aturan karena hal ini akan memudahkan dalam perancangan sistem.

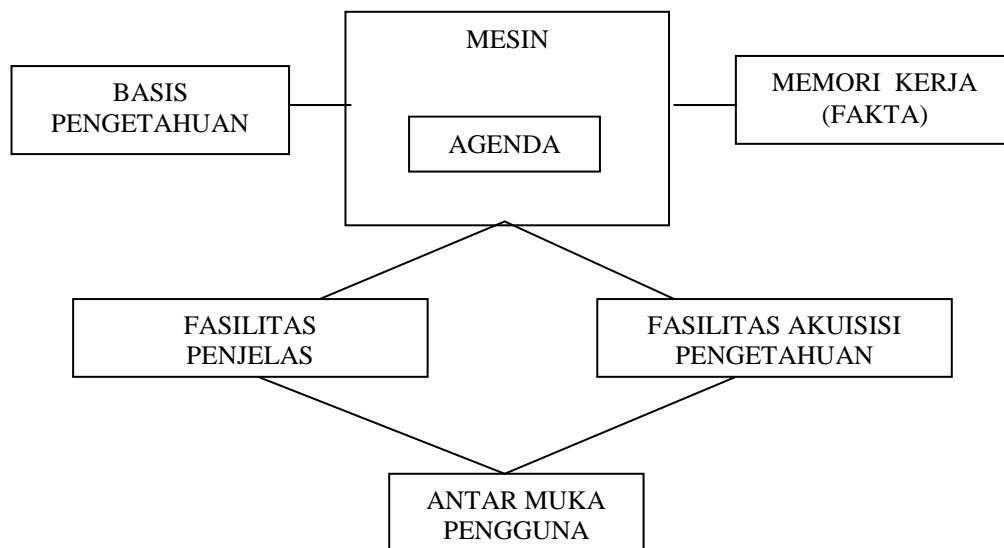
5. Antarmuka Pengguna (*User Interface*).

Antarmuka pemakai memberikan fasilitas komunikasi antara pemakai sistem, memberikan berbagai fasilitas informasi dan berbagai keterangan yang bertujuan untuk membantu mengarahkan alur penelusuran masalah sampai ditemukannya solusi.

Antarmuka pemakai juga memiliki fungsi untuk menginputkan pengetahuan baru ke dalam basis pengetahuan sistem pakar dan memberikan tuntunan langkah demi langkah penggunaan sistem sehingga pemakai mengerti apa yang harus dilakukan terhadap sistem. Oleh sebab itu, kemudahan dalam menjalankan sistem menjadi syarat utama untuk membangun antarmuka pemakai. Sebuah program harus interaktif, komunikatif dan mudah digunakan.

II.1.4. Arsitektur Sistem Pakar

Arsitektur sistem pakar dapat dilihat pada Gambar II.4.



Gambar II.4. Arsitektur Sistem Pakar

(Sumber : Kusrini ; 2006 : 19)

Memori kerja dalam arsitektur sistem pakar merupakan bagian dari sistem pakar yang berisi fakta-fakta masalah yang ditemukan dalam suatu sesi, berisi fakta-fakta tentang suatu masalah yang di temukan dalam proses konsultasi.

II.1.5. Kategori Masalah Sistem Pakar

Masalah – masalah yang dapat diselesaikan dengan sistem pakar, antara lain:

1. *Interpretasi* ialah membuat kesimpulan atau deskripsi dari sekumpulan data mentah.
2. *Prediksi* ialah memproyeksikan akibat-akibat yang di mungkinkan dari situasi–situasi tertentu.
3. *Diagnosis* ialah menentukan sebab malfungsi dalam dalam situasi kompleks yang didasarkan pada gejala–gejala yang teramati.
4. *Desain* ialah menentukan konfigurasi komponen–komponen sistem yang cocok dengan tujuan–tujuan kinerja tertentu yang memenuhikendala–kendala tertentu.
5. *Planning* ialah merencanakan serangkaian tindakan yang akan dapat mencapai sejumlah tujuan dengan kondisi awal tertentu.
6. *Debugging* dan *Repair* ialah menentukan dan interpretasikan cara untuk mengatasi malfungsi.
7. *Intruksi* ialah mendeteksi dan mengoreksi defisiensi dalam pemahaman domain subyek.
8. *Controlling* ialah mengatur tingkah laku suatu *environment* yang kompleks.

9. *Selection* ialah mengidentifikasi pilihan terbaik dari sekumpulan kemungkinan.
10. *Simulation* ialah pemodelan interaksi antara komponen–komponen sistem.
11. *Monitoring* ialah membandingkan hasil pengamatan dengan kondisi yang di harapkan.

II.2. Psikologi Pada Anak

II.2.1. Pengertian Psikologi Anak

Ditinjau dari segi bahasa, perkataan psikologi berasal dari kata *Psyce* yang diartikan sebagai jiwa, dan kata *logos* yang berarti ilmu atau ilmu pengetahuan. Maka psikologi diartikan sebagai ilmu pengetahuan tentang jiwa yang disingkat dengan ilmu jiwa. (Bimo : 2001). Jadi, secara umum psikologi dapat diartikan sebagai ilmu yang mempelajari tentang jiwa ataupun perilaku, baik macam-macam gejalanya, prosesnya maupun latar belakangnya. Sedangkan psikologi anak adalah ilmu yang mempelajari mengenai kejiwaan dan tingkah laku pada anak.

Keadaan psikologi anak menjadi perhatian beberapa pakar ahli anak karena semakin cepatnya perkembangan gangguan anak yang terjadi. Saat ini berbagai metode deteksi dini untuk mengetahui gangguan psikologi anak telah dibuat. Masa tumbuh kembang anak merupakan hal yang sangat penting mengingat berbagai aspek pertumbuhan manusia banyak terjadi dimasa anak-anak.

II.2.2. Gangguan Psikologi Pada Anak

Manusia dalam hidupnya selalu mengalami perkembangan. Dari mulai dilahirkan sebagai seorang bayi, berkembang menjadi anak-anak, remaja, dewasa, tua dan akhirnya meninggal dunia. Dalam perjalanannya tersebut tidak sedikit yang mengalami berbagai gangguan dan permasalahan yang kemudian disebut sebagai hambatan atau gangguan psikologi. Anak-anak merupakan fase paling rentan sebagai deteksi awal gangguan psikologi. Adapun macam-macam gangguan psikologi anak antara lain:

a. Autis

Kata autisme, diambil dari kata Yunani "autos" yang berarti "aku". Dalam pengertian non ilmiah mudah menimbulkan interpretasi bahwa semua anak yang bersikap mengarah pada dirinya sendiri karna sebab apapun disebut autistik. (Eko Suryani, Hesty Widyasih ; 2010 : 82). Ciri khas autisme adalah bahwa mereka sejak lahir memiliki kontak sosial yang sangat terbatas. Perhatian mereka hampir tidak tertuju pada orang lain, melainkan hanya pada benda-benda mati. Faktor genetika diperkirakan menjadi penyebab utama dari kelainan autis, walaupun bukti-bukti yang konkrit masih sulit ditemukan. Diprediksikan bahwa ada kelainan kromosom pada anak autis, namun kelainan itu tidak berada pada kromosom yang sama. Hanya dengan penanganan dan pendidikan yang penuh kasih sayang, konsekuen, tidak kenal jemu dan dalam jangka waktu yang sangat lama, dapat terjadi perbaikan pada anak autis.

b. ADHD (*Attention Deficit Hyperactivity Disorder*)

ADHD dapat diterjemahkan dengan Gangguan Pemusatan Perhatian dan Hiperaktivitas atau GPPH. Orang awan sering menyebutnya dengan anak hiperaktif saja. Sebenarnya hiperaktif bukan nama penyakit, tetapi hanya salah satu gejalanya. Istilah hiperaktif dipakai untuk anak dengan kelainan perilaku. Sebenarnya anak normalpun pada tahap perkembangan tertentu, juga mengalami semacam hiperaktifitas, tetapi istilah yang dipakai untuk anak normal

Gejala anak dengan ADHD sekilas mirip autisme, tetapi memiliki kemampuan komunikasi dan interaksi sosial yang jauh lebih baik. Pada tahap tertentu setelah mengalami terapi yang efektif, perilaku anak autisme mirip sekali dengan ADHD.

c. Sindrom Asperger

Sindrom Asperger mirip dengan autisme infantil dalam hal kurangnya interaksi sosial. Tetapi mereka masih mampu berkomunikasi dengan cukup baik. Anak sering memperlihatkan perilaku yang kurang wajar dan minat yang terbatas. Anak mampu mengikuti kegiatan sekolah dengan prestasi rata-rata atau diatas rata-rata.

Seseorang penyandang SA dapat memperlihatkan bermacam-macam karakter dan gangguan tersebut. Seseorang penyandang SA dapat memperlihatkan kekurangan dalam bersosialisasi, mengalami kesulitan jika terjadi perubahan, dan selalu melakukan hal-hal yang sama berulang ulang. Sering mereka terobsesi oleh rutinitas dan menyibukkan diri dengan sesuatu

aktivitas yang menarik perhatian mereka. Mereka selalu mengalami kesulitan dalam membaca aba-aba (bahasa tubuh) dan seringkali seseorang penyandang SA mengalami kesulitan dalam menentukan dengan baik posisi badan dalam ruang (orientasi ruang dan bentuk).

II.3. Pengertian Basis Data

Database merupakan komponen terpenting dalam pembangunan sistem informasi, karena menjadi tempat untuk menampung dan mengorganisasikan seluruh data yang ada dalam sistem, sehingga dapat dieksplorasi untuk menyusun informasi-informasi dalam berbagai bentuk. *Database* merupakan himpunan kelompok data yang saling berkaitan. Data tersebut diorganisasikan sedemikian rupa agar tidak terjadi duplikasi yang tidak perlu, sehingga dapat diolah atau dieksplorasi secara cepat dan mudah untuk dihasilkan informasi (Budi Sutedjo Dharma Oetomo ; 2006 : 101).

Menurut Budi Sutedjo Dharma Oetomo (2006 : 102), hirarki atau tingkatan data dalam database yaitu :

1. Database

Suatu *database* menggambarkan data yang saling berhubungan antara satu dengan yang lainnya.

2. File, yaitu kumpulan dari record yang saling terkait dan memiliki format *field* yang sama dan sejenis.

3. Record, yaitu kumpulan *field* yang menggambarkan suatu unit data individu tertentu.

4. Field, yaitu atribut dari *record* yang menunjukkan suatu item dari data seperti nama, alamat, dan lain sebagainya.
5. Byte, atribut dari *field* yang berupa huruf yang membentuk nilai dari sebuah *field*. Huruf tersebut dapat berupa numerik maupun abjad atau karakter khusus.
6. Bit, yaitu bagian terkecil dari data secara keseluruhan yaitu berupa karakter ASCII (*American Standard Code Form Information Interchange*) nol atau satu yang merupakan komponen pembentuk *byte*.

II.4. Kamus Data

Menurut Jogiyanto (2005: 725), 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. Dengan menggunakan KD, analis sistem dapat mendefinisikan data yang mengalir di sistem dengan lengkap.

Menurut Budi Sutedjo Dharma Oetomo (2006 : 118) kamus data ikut berperan dalam perancangan dan pembangunan SI karena peralatan ini berfungsi untuk :

1. Menjelaskan arti aliran data dan penyimpanan dalam penggambaran dalam *data flow diagram*.
2. Mendeskripsikan komposisi paket data yang bergerak melalui aliran, misalnya data alamat diurai menjadi nama jalan, nomor, kota, Negara dan kode pos.

3. Menjelaskan spesifikasi nilai dan satuan yang relevan terhadap data yang mengalir dalam sistem tersebut.

II.5. Normalisasi

Normalisasi adalah transformasi tinjauan pemakai yang kompleks dan data yang tersimpan ke sekumpulan bagian – bagian struktur data yang kecil dan stabil. Disamping menjadi lebih sederhana dan lebih stabil, struktur data yang dinormalisasikan lebih mudah diatur daripada struktur data yang lain. Adapun bentuk-bentuk Normalisasi menurut (Kusrini : 2007 ; 41-43) adalah :

1. Bentuk Tidak Normal

Bentuk ini merupakan kumpulan data yang akan direkam, tidak ada keharusan mengikuti format tertentu, dapat saja tidak lengkap dan terduplikasi. Data dikumpulkan apa adanya sesuai keadaannya.

2. Bentuk Normal Tahap Pertama (1st *Normal Form*)

Defenisi :

Sebuah tabel disebut 1NF jika :

- Tidak ada baris yang duplikat dalam tabel tersebut
- Masing-masing *cell* bernilai tunggal

3. Bentuk Normal Tahap Kedua (2nd *Normal Form*)

Bentuk Normal Kedua (2NF) terpenuhi jika pada sebuah table semua atribut yang tidak termasuk dalam *primary key* memiliki ketergantungan fungsional pada *primary key* secara utuh. Sebuah table dikatakan tidak memenuhi 2NF, jika ketergantungannya hanya bersifat parsial (hanya tergantung pada sebagian dari *primary key*).

4. Bentuk Normal Normal Tahap Ketiga (2rd Normal Form)

Sebuah tabel dikatakan memenuhi bentuk normal ketiga (3NF), jika setiap ketergantungan fungsional dengan notasi $X \rightarrow A$, dimana A mewakili semua atribut tunggal di dalam tabel yang tidak ada didalam X, maka :

- X haruslah *superkey* pada tabel tersebut
- Atau A merupakan bagian dari *primary key* pada tabel tersebut.

5. Bentuk Normal Tahap Keempat dan Kelima

Penerapan aturan normalisasi sampai bentuk normal ketiga sudah memadai untuk menghasilkan tabel berkualitas baik. Namun demikian, terdapat pula bentuk normal keempat (4NF) dan kelima (5NF). Bentuk normal keempat berkaitan dengan sifat ketergantungan banyak nilai (*multivalued dependency*) pada suatu table yang merupakan pengembangan dari ketergantungan fungsional. Adapun bentuk normal tahap kelima merupakan nama lain dari *Project Join Normal Form* (PJNF).

6. Boyce Code Normal Form (BCNF)

- Memenuhi 1st NF
- Relasi harus bergantung fungsi pada atribut *superkey*.

II.6. Entity Relationship Diagram (ERD)

Entity Relational Diagram merupakan salah satu pemodelan data konseptual yang paling sering digunakan dalam proses pengembangan basis data bertipe relasional. Model E-R adalah rincian yang merupakan representasi logika

dari data pada suatu organisasi atau area bisnis tertentu. Model E-R terdiri dari beberapa komponen dasar yaitu sebagai berikut:

1. Entitas

Entitas adalah sesuatu atau objek di dunia nyata yang dapat dibedakan dari sesuatu atau objek yang lainnya. Sebagai contoh, setiap mahasiswa dalam suatu universitas adalah suatu entitas. Setiap fakultas dalam suatu universitas adalah juga suatu entitas. Dapat dikatakan bahwa entitas bisa bersifat konseptual/abstrak atau nyata hadir di dunia nyata.

2. Atribut

Atribut adalah properti deskriptif yang dimiliki oleh setiap anggota dari himpunan entitas. Sebagai contoh entitas mahasiswa, atribut-atribut yang dimiliki adalah nim, nama mahasiswa, alamat dan lain-lain.

3. Hubungan antar relasi (Relationship)

Hubungan antar relasi adalah hubungan antara suatu himpunan entitas dengan himpunan entitas yang lainnya. Misalnya, entitas mahasiswa memiliki hubungan tertentu dengan entitas matakuliah (mahasiswa mengambil matakuliah). Pada penggambaran model E-R, relasi adalah perekat yang menghubungkan suatu entitas dengan entitas yang lainnya.

4. Kardinalitas/Derajat Relasi

Kardinalitas relasi menunjukkan jumlah maksimum entitas yang dapat berelasi dengan entitas pada himpunan entitas yang lain. Sebagai contoh: entitas-entitas pada himpunan entitas mahasiswa dapat berelasi dengan satu entitas, banyak

entitas atau tidak satupun entitas dari himpunan entitas kuliah. Kardinalitas relasi yang terjadi di antara dua himpunan entitas dapat berupa:

- Satu ke Satu(One to One)

Setiap entitas pada himpunan entitas A berhubungan dengan paling banyak dengan satu entitas pada himpunan entitas B, begitupun sebaliknya.

- Satu ke Banyak(One to Many)

Setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B, tetapi tidak sebaliknya, dimana setiap entitas pada himpunan entitas B berhubungan dengan paling banyak dengan satu entitas pada himpunan entitas A.

- Banyak ke Satu(Many to One)

Setiap entitas pada himpunan entitas A berhubungan dengan paling banyak dengan satu entitas pada himpunan entitas B, tetapi tidak sebaliknya dengan entitas B.

- Banyak ke Banyak (Many to Many)

Setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B, dan demikian sebaliknya, dimana setiap entitas pada himpunan entitas B dapat berhubungan dengan banyak entitas pada himpunan entitas A. (Radiant Victor Imbar dan Dewanto Adi Putra : Vol.2 ; 2007 : 169-170).

Menurut Kusri (2007 : 33), adapun langkah-langkah untuk mengimplementasikan *Entity Relationship Diagram* adalah sebagai berikut:

- Transformasi dari model data yang telah dibuat ke skema basis data sesuai dengan DBMS yang dipilih.
- Diagram ER → basis data
- *Entity* → tabel-tabel/file-file data.
- *Atribut* → *field*

II.7. Unified Modelling Language (UML)

Pengembangan sistem adalah aktivitas manusia. Tanpa adanya kemudahan untuk memahami sistem notasi, proses pengembangan kemungkinan besar akan mengalami kesalahan. UML adalah sistem notasi yang sudah dibakukan di dunia pengembangan sistem, hasil kerjasama dari Grady Booch, James Rumbaugh dan Ivar Jacobson. UML yang terdiri dari serangkaian diagram memungkinkan bagi sistem analis untuk membuat cetak biru sistem yang komperhesif kepada klien, programmer dan tiap orang yang terlibat dalam proses pengembangan tersebut. Dengan UML akan dapat menceritakan apa yang seharusnya dilakukan oleh sistem bukan bagaimana yang seharusnya dilakukan oleh sebuah sistem. (Munawar ; 2005 : 25).

Menurut Prabowo Pudjo Widodo dan Herlawati (2011 : 6-7) UML diaplikasikan untuk maksud tertentu, biasanya antara lain untuk:

1. Merancang perangkat lunak.

2. Sarana komunikasi antara perangkat lunak dengan proses bisnis.
3. Menjabarkan sistem secara rinci untuk analisa dan mencari apa yang diperlukan sistem.
4. Mendokumentasikan sistem yang ada, proses-proses dan organisasinya.

UML adalah bahasa untuk menspesifikasi, memvisualisasi, membangun dan mendokumentasikan *artefacts* (bagian dari informasi yang digunakan atau dihasilkan oleh proses pembuatan perangkat lunak, seperti pada pemodelan bisnis dan sistem non perangkat lunak lainnya. UML tidak hanya merupakan sebuah bahasa pemrograman visual saja, namun juga dapat secara langsung duhubungkan keberbagai bahasa pemrograman seperti JAVA, C++, Visual Basic atau bahkan dihubungkan secara langsung kedalam sebuah *object oriented database*. Begitu juga mengenai pendokumentasian dapat dilakukan seperti *requirements*, arsitektur, *design*, *source*, *project plan*, *tests* dan *prototypes*. UML memiliki 8 tipe diagram, namun pada penulisan skripsi ini penulis akan menggunakan 4 tipe diagram UML yaitu *use case diagram*, *sequence diagram*, *activity diagram* dan *class diagram*.

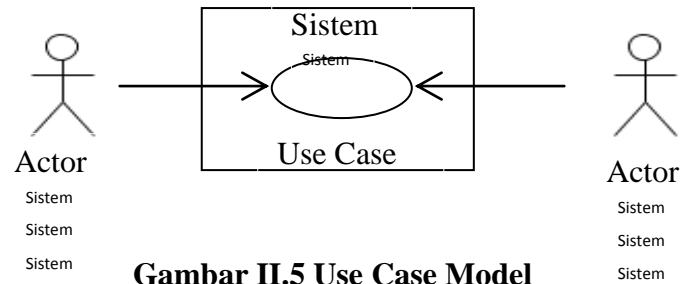
II.7.1. Tipe Diagram UML

Adapun 8 tipe diagram UML adalah :

1. *Use Case Diagram*

Diagram use case menunjukkan 3 aspek dari sistem, yaitu actor, use case, dan sistem/ sub sistem boundary. Actor mewakili peran orang, sistem yang lain

atau alat ketika berkomunikasi dengan use case. Gambar II.5 mengilustrasikan actor, use case dan boundary.



Gambar II.5 Use Case Model

(Sumber : Munawar ; 2005 : 64)

2. *Class Diagram*

Class diagram menggambarkan struktur dan deskripsi class, package dan objek beserta hubungan satu sama lain seperti containment, pewarisan, asosiasi dan lain-lain.

3. *Statechart Diagram*

Diagram Statechart menggambarkan semua state yang dimiliki oleh suatu objek dari suatu class dan keadaan yang menyebabkan state berubah.



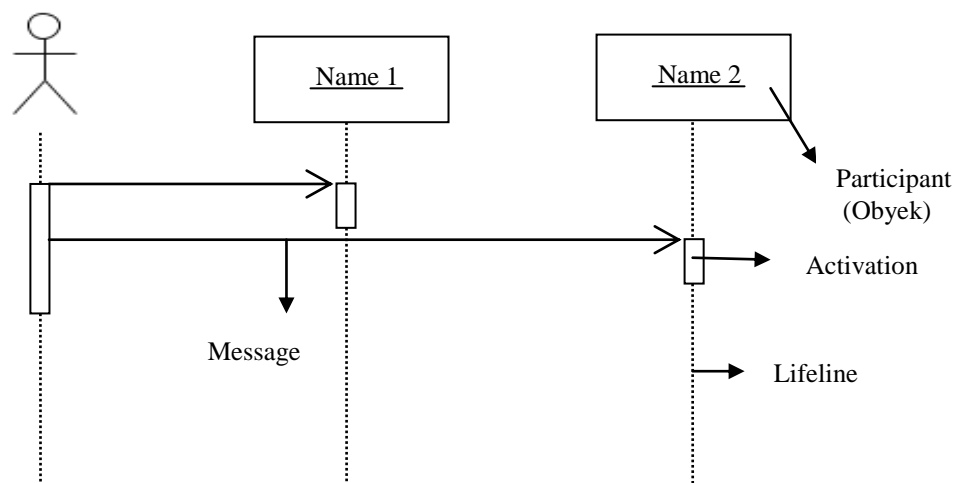
Gambar II.6. Simbol Diagram Statechart

(Sumber : Munawar ; 2005 : 74)

Pada gambar II.6. bentuk lingkaran solid memaparkan bagaimana objek dibentuk atau diawali, tanda panah (*transition*) menjelaskan bagaimana sebuah kejadian yang memicu sebuah state objek dengan cara memperbaharui satu atau lebih nilai atributnya, dan *double round* memaparkan bagaimana sebuah objek dibentuk dan dihancurkan.

4. *Sequence Diagram*

Sequence diagram digunakan untuk menggambarkan perilaku pada sebuah skenario. Diagram ini menunjukkan sejumlah contoh obyek dan pesan yang diletakkan diantara obyek-obyek dalam *use case*.



Gambar II.7. Simbol-simbol *Sequence Diagram*

(Sumber : Munawar ; 2005 : 89)




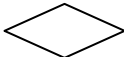


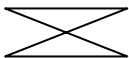
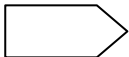
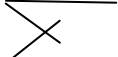

5. *Collaboration Diagram*

Menggambaran kolaborasi dinamis seperti *sequence diagram*. Dalam menunjukkan pertukaran pesan, *collaboration diagram* dengan menggunakan objek dan hubungannya.

6. 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.

Tabel II.1 Simbol Activity Diagram

Gambar	Nama
	Titik awal
	Titik akhir
	<i>Activity</i>
	Pilihan untuk pengambilan keputusan
	<i>Fork</i> , digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan parallel menjadi satu
	<i>Rake</i> , menunjukkan adanya dekomposisi
	Tanda waktu
	Tanda pengiriman
	Tanda penerimaan
	<i>Flow final</i>

(Sumber : Munawar ; 2005 : 109-110)

7. *Component Diagram*

Component diagram menggambarkan struktur dan hubungan antara komponen piranti lunak, termasuk ketergantungan. Komponen dapat juga berupa interface, yaitu kumpulan layanan yang disediakan sebuah komponen untuk komponen lainnya.

8. *Deployment Diagram*

Deployment diagram menggambarkan arsitektur fisik dari perangkat keras dan perangkat lunak system, menunjukkan hubungan computer dengan perangkat satu sama lain dan jenis hubungannya.

II.8. PHP

PHP sebagai alternatif lain memberikan solusi sangat murah karena gratis digunakan dan dapat berjalan di berbagai jenis platform. Pada awalnya PHP berjalan di sistem UNIX dan *variant*-nya, namun kini dapat berjalan dengan di lingkungan sistem operasi Windows. Rasmus Lerdorf adalah bapak penemu awal bahasa PHP ini, yang bermula dari keinginan sederhana Lerdorf untuk mempunyai alat bantu (*tools*) dalam memonitor pengunjung yang melihat situs web pribadinya.

PHP merupakan singkatan dari *Personal Home Page tools*, sebelum akhirnya menjadi singkatan rekursif dari *PHP: Hypertext Preprocessor*. Pertengahan tahun 1995 dirilis PHP/FI (FI adalah singkatan dari *Form Interpreter*) yang memiliki kemampuan dasar membangun aplikasi web,

memproses form, dan mendukung database MySQL. Interpreter PHP mengeksekusi kode PHP pada sisi server (disebut *server-side*) dan berbeda dengan mesin maya Java yang mengeksekusi program pada sisi client (*client-side*). Proses eksekusi kode PHP disisipkan pada halaman HTML. Beberapa keunggulan yang dimiliki program PHP adalah:

1. PHP memiliki akses yang lebih cepat dan bersifat *open source*.
2. PHP memiliki tingkat *lifecycle* yang cepat sehingga selalu mengikuti perkembangan teknologi internet.
3. PHP memiliki tingkat keamanan yang tinggi.
4. PHP mampu berjalan di beberapa server yang ada, misalnya Apache, Microsoft IIS, PWS, AOLServer, phttpd, fhttpd, dan Xitami.
5. PHP mampu berjalan di Linux sebagai platform sistem operasi yang utama bagi PHP, namun juga dapat berjalan di FreeBSD, Unix, Solaris, Windows dan yang lainnya.
6. PHP juga mendukung akses ke beberapa database yang sudah ada, baik yang bersifat gratis ataupun komersial. Database itu antara lain MySQL, PostgreSQL, mSQL, Informix, dan MicrosoftSQL Server. (Andi offset dan Madcoms,dkk ; 2009 : 2).

II.9. MySQL

Basis data merupakan sekumpulan tabel atau objek lain (*view*, *indeks*, dan lain-lain). Tujuan utama pembuatan basis data adalah untuk memudahkan dalam mengakses data. Data dapat ditambahkan, diubah, dihapus atau dibaca dengan

relative mudah dan cepat. Saat ini tersedia banyak perangkat lunak yang ditujukan untuk mengelola *database* seperti Access, MicrosoftSQL Server,MySQL dan masih banyak lagi. Pada penulisan skripsi ini penulis menggunakan MySQL sebagai *database*-nya.

MySQL (baca:mai-se-kyu-el) merupakan software yang tergolong database server bersifat *Open Source*. *Open Source* menyatakan bahwa software ini dilengkapi dengan *source code* (kode yang dipakai untuk membuat MySQL), selain itu tentu saja bentuk *executable*-nya atau kode yang dapat dijalankan secara langsung dalam sistem operasi, dan bisa diperoleh dengan cara mengunduh di internet secara gratis. Hal menarik lainnya adalah MySQL juga bersifat multiplatform. MySQL dapat dijalankan pada berbagai sistem operasi. (Abdul Kadir ; 2009 : 15)

MySQL adalah salah jenis database server yang sangat terkenal. MySQL termasuk jenis RDMS (*Relation Database Manajement Sistem*). MySQL ini mendukung bahasa pemograman PHP. MySQL juga mempunyai query atau bahasa SQL (*Structured Query Language*) yang simple dan menggunakan *escape character* yang sama dengan PHP.

II.10. Aplikasi Web

Menurut Abdul Kadir (2009 : 5), aplikasi web adalah jenis aplikasi yang diakses melalui *browser*, misalnya Internet Explorer dan Mozilla Explorer sedangkan web server adalah server yang melayani permintaan aplikasi web. Aplikasi web paling dasar ditulis dengan menggunakan HTML (*Hypertext Markp*

Language) yang merupakan bahasa standart untuk membuat halaman-halaman web.

World Wide Web merupakan server pada internet yang didukung oleh bahasa script pemrograman HTML. Didalam Word Wide Web ada beberapa pengertian yang harus dipahami seperti Web Page, Home Page, dan Web Site. Web Page merupakan sebuah dokumen yang berada pada WWW. Setiap halaman web diidentifikasi dengan sebuah alamat unik URL (*Uniform Resouce Locator*). Home Page adalah halaman web yang utama dan biasanya merupakan halaman web yang muncul pertama kali saat kita mengunjungi suatu situs. Web Site merupakan sebuah *site* (lokasi) pada WWW dimana situs terdiri dari sebuah home page, dimana kitadapay melihat halaman pertama yang ditampilkan dan silihat oleh penggunanya dan dapat melanjutkan ke situs *link* selanjutnya.

II.11. Macromedia Dreamweaver

Macromedia Dreamweaver adalah sebuah software HTML editor professional yang digunakan untuk mendesain secara visual dan mengelola situs web maupun halaman web. Menurut Ali Akbar (2006 : 12), Apapun jenis halaman web yang akan dibuat, Dreamweaver mampu digunakan untuk mendesainnya. Dreamweaver tercatat mampu digunakan untuk menangani tipe-tipe halaman web berikut:

- HTML
- Cold Fusion

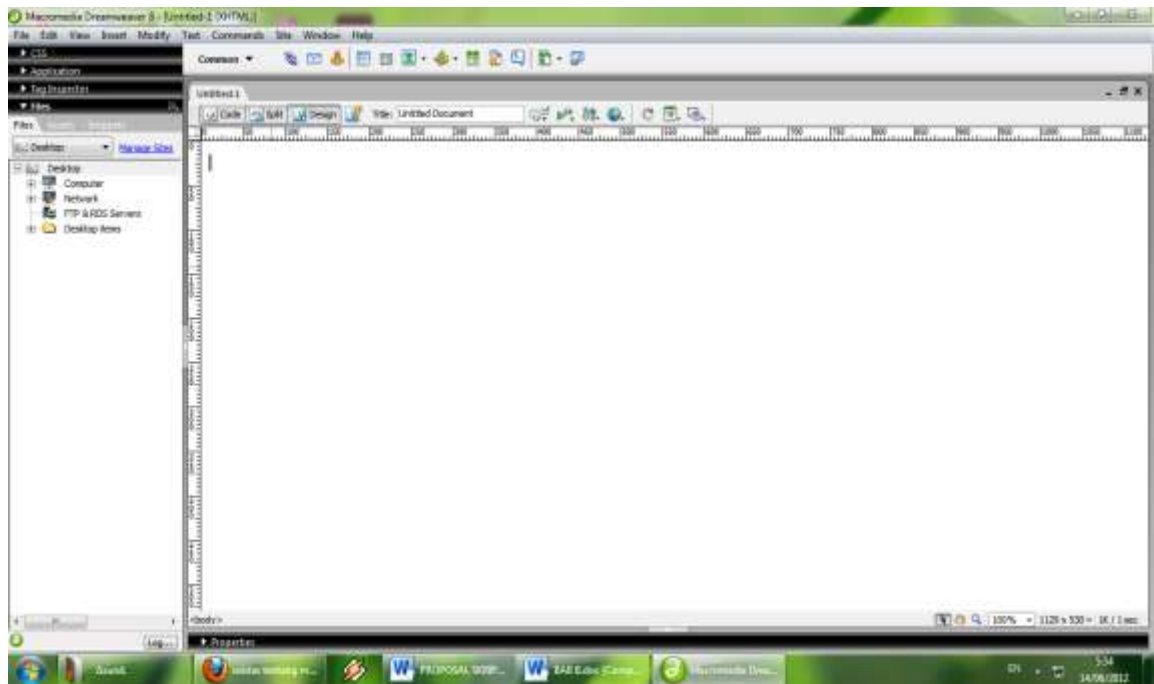
- PHP
- ASP VBScript
- ASPNET C#
- JavaScript
- XML
- XSLT
- CSS
- JSP

Versi dreamweaver yang penulis gunakan adalah versi 8. Pada versi ini, tampilan mengalami perubahan yang kaya akan warna dan area kerjanya menjadi lebih ringkas namun efisien. Dreamweaver 8 memiliki banyak tools untuk kode-kode di halaman web beserta fasilitas-fasilitasnya, antara lain : Referensi HTML, CSS dan java script, java script debugger dan editor kode (tampilan kode dan kode *inspector*) yang memperbolehkan kita mengedit kode java script, XML dan dokumen teks lain secara langsung. Berikut ini akan ditampilkan halaman awal dan area kerja pada Dreamweaver 8:



Gambar II.8. Tampilan Awal Macromedia Dreamweaver 8

Pada saat pertama kali membuka dreamweaver akan muncul tampilan seperti pada Gambar II.8. Pengguna bebas memilih area kerja yang dibutuhkan sesuai dengan bahasa script pemrograman web yang dibutuhkan seperti HTML, PHP, ASP.Net, JavaScript, CSS, dan sebagainya. Selanjutnya pengguna akan masuk ke area kerja dreamweaver 8 yang dapat dilihat pada Gambar II.9.



Gambar II.9. Area Kerja Macromedia Dreaweaver 8

Di area kerja ini pengguna bebas memilih penulisan kode ataupun pembuatan desain website melalui *Code*, *Split*, atau *Design*. Serta menggunakan *tools-tools* yang interaktif untuk mempercantik website yang akan dibuat.