



BAB II

TINJAUAN PUSTAKA

2.1 Teori Judul

2.1.1 Sistem Pakar

Andriani (2017:9) mendefinisikan sistem pakar sebagai sebuah sistem yang cara kerjanya mengadopsi keahlian dari seorang pakar dalam bidang tertentu ke dalam sistem atau program komputer yang disajikan dengan tampilan yang dapat digunakan oleh pengguna yang bukan seorang pakar sehingga dengan sistem tersebut pengguna dapat membuat sebuah keputusan atau menentukan kebijakan layaknya seorang pakar.

Menurut Sutojo dalam Aji, furqon, dan Widodo (2018:2128) istilah sistem pakar berasal dari istilah *knowledge-based expert system*. Sistem pakar memasukkan pengetahuan seorang pakar ke dalam komputer. Seorang yang bukan pakar/ahli dapat menggunakan sistem pakar untuk meningkatkan kemampuan pemecahan masalah, sedangkan seorang pakar dapat menggunakan sistem pakar untuk *knowledge assistant*.

Sedangkan menurut Arhami dalam Miranda, dkk. (2016:124) salah satu teknik kecerdasan buatan yang menirukan penalaran manusia adalah Sistem Pakar. Secara umum, Sistem Pakar (*expert system*) adalah sistem yang berusaha memasukkan pengetahuan manusia ke komputer, agar komputer dapat menyelesaikan masalah seperti yang biasa dilakukan oleh para ahli. Sistem Pakar yang baik dirancang agar dapat menyelesaikan suatu permasalahan tertentu dengan meniru kerja para ahli.

Dari beberapa pendapat ahli di atas dapat disimpulkan bahwa sistem pakar adalah suatu sistem yang berusaha untuk mengadopsi atau memasukkan keahlian dari seorang pakar ke dalam sistem komputer, agar seseorang yang bukan pakar/ahli dapat menyelesaikan masalah seperti yang dilakukan seorang pakar.

Menurut Andriani (2017:11) ciri-ciri dari sistem pakar adalah sebagai berikut:

1. Memiliki dan memberikan informasi yang handal.
2. Mudah untuk dimodifikasi.



3. Terbatas pada domain keahlian tertentu.
4. Dapat memberikan penalaran untuk data-data yang sifatnya tidak pasti.
5. Sistem berdasarkan pada kaidah/*rule* tertentu.
6. Memiliki kemampuan untuk belajar beradaptasi.
7. Keluarannya bersifat anjuran.

Menurut Andriani (2017:14) sistem pakar mempunyai komponen utama pada strukturnya, antara lain sebagai berikut.

1. Basis pengetahuan (*Knowledge Base*)

Inti dari suatu sistem pakar adalah basis pengetahuan yang merupakan representasi pengetahuan yang dimiliki oleh seorang pakar yang tersusun oleh atas fakta dan kaidah. Basis pengetahuan bisa kita dapatkan langsung dari seorang pakar maupun dari data histori yang berisi data-data pengetahuan dari seorang pakar.

2. Mesin Inferensi (*Inference Engine*)

Otak dari sebuah sistem pakar adalah mesin inferensi yang berfungsi untuk memandu proses penalaran terhadap suatu kondisi berdasarkan pada basis pengetahuan yang tersedia. Terdapat dua penalaran yang dapat dilakukan dalam melakukan inferensi, yaitu:

- a. *Forward Chaining*

Merupakan cara penalaran dengan memulai dari fakta terlebih dahulu untuk menguji kebenaran hipotesis atau mencocokkan fakta atau pernyataan dimulai dari bagian sebelah kiri dulu (*IF* dulu). *Forward Chaining* merupakan grup dari multiple inferensi yang melakukan pencarian dari suatu masalah kepada solusinya. Jika klausa premis sesuai dengan situasi (bernilai *TRUE*), maka proses akan meng-*assert* konklusi.

- b. *Backward Chaining*

Merupakan cara penalaran dengan memulai dari hipotesis (ekspektasi apa yang diinginkan terjadi) terlebih dahulu, dan untuk menguji kebenaran hipotesis tersebut harus dicari fakta-fakta yang ada dalam basis pengetahuan.



Backward Chaining juga merupakan penalaran dengan mencocokkan fakta atau pernyataan yang dimulai dari bagian sebelah kanan (*THEN* dulu).

3. Basis Data (*Database*)

Merupakan kumpulan data yang terdiri dari semua fakta yang diperlukan, dimana fakta-fakta tersebut digunakan untuk memenuhi kondisi dari kaidah-kaidah dalam sistem.

4. Antarmuka Pemakai (*User Interface*)

Merupakan fasilitas yang dapat digunakan sebagai perantara komunikasi antara pemakai dengan komputer dalam menggunakan sistem pakar. Antarmuka ini memudahkan pengguna sistem pakar yang bukan merupakan seorang pakar dapat bekerja dan bertindak atau membuat keputusan layaknya seorang pakar.

2.1.2 Metode *Certainty Factor* (CF)

Metode *certainty factor* atau faktor kepastian adalah salah satu teori yang dapat digunakan untuk menyelesaikan masalah ketidakpastian. *Certainty factor* merupakan nilai untuk mengukur keyakinan pakar. *Certainty factor* diperkenalkan oleh Shortliffe Buchanan dalam pembuatan sistem pakar MYICN yang merupakan nilai parameter klinis yang diberikan MYCIN untuk menunjukkan besarnya kepercayaan. *Certainty factor* menunjukkan ukuran kepastian terhadap suatu fakta atau aturan. Nilai tertinggi dalam *certainty factor* adalah +1,0 (pasti benar atau *Definitely*), dan nilai terendah *certainty factor* adalah -1,0 (pasti salah atau *Definitely not*). Nilai positif merepresentasikan derajat keyakinan, sedangkan nilai negatif merepresentasikan derajat ketidakpercayaan. *Certainty factor* didefinisikan sebagai berikut (Andriani, 2017:18):

$$CF [h,e] = MB[h,e] - MD[h,e]$$

Dimana:

CF[h,e] = Faktor Kepastian

MB[h,e] = Uuran kepercayaan / tingkat keyakinan terhadap hipotesis h, jika Diberikan / dipengaruhi *evidence* e (antara 0 dan 1)



MD[h,e] = Ukuran ketidakpercayaan / tingkat ketidakyakinan terhadap Hipotesis h, jika diberikan / dipengaruhi *evidence* e (antara 0 dan 1)

Adapun beberapa kombinasi certainty factor terhadap premis tertentu:

1. *Certainty factor* dengan satu premis.

$$\begin{aligned} CF[h,e] &= CF[e] * CF[rule] \\ &= CF[user] * CF[pakar] \end{aligned}$$

2. *Certainty factor* dengan lebih dari satu premis.

$$\begin{aligned} CF[A \wedge B] &= \text{Min}(CF[a], CF[b]) * CF[rule] \\ CF[A \vee B] &= \text{Max}(CF[a], CF[b]) * CF[rule] \end{aligned}$$

3. *Certainty factor* dengan kesimpulan yang serupa.

$$CF_{gabungan}[CF1, CF2] = CF1 + CF2 * (1 - CF1)$$

Kelebihan dari metode ini adalah cocok digunakan pada sistem pakar yang mengukur sesuatu yang pasti atau tidak pasti seperti mendiagnosis penyakit dan perhitungan dari metode ini hanya berlaku untuk sekali hitung, serta hanya dapat mengolah dua data sehingga keakuratannya terjaga (Turban, 2001).

2.1.3 Excavator

Lindawati (2013) mendefinisikan *Excavator* (ekskavator) adalah alat berat yang terdiri dari lengan (*arm*), *boom* (bahu) serta *bucket* (alat keruk) dan digerakkan oleh tenaga hidrolik yang dimotori dengan mesin diesel dan berada di atas roda rantai (*trackshoe*). *Excavator* merupakan alat berat paling serbaguna karena bisa menangani berbagai macam pekerjaan alat berat lain. Sesuai dengan namanya (*excavation*), alat berat ini memiliki fungsi utama untuk pekerjaan penggalian. Namun tidak terbatas itu saja, *excavator* juga bisa melakukan pekerjaan konstruksi seperti membuat kemiringan (*sloping*), memuat *dumptruck* (*loading*), pemecah batu (*breaker*), dan sebagainya.

Sedangkan menurut Pratatis (2016:535), “*Excavator* adalah alat yang serba guna yang dapat untuk menggali tanah, membuat parit, memuat material ke *dump truck* atau kayu ke *trailer*.”



Dari beberapa pengertian diatas dapat disimpulkan bahwa Excavator adalah alat berat yang paling serbaguna yang dapat digunakan untuk beberapa pekerjaan seperti penggalian, memuat material ke *dump truck*, pemecah batu dan sebagainya.

2.1.4 Android

Menurut Juhara (2016:1), “Android adalah sistem operasi berbasis Linux yang dimodifikasi untuk perangkat bergerak (*mobile device*) yang terdiri dari sistem operasi, *middleware*, dan aplikasi-aplikasi utama.”

Menurut Safaat (2015:1), “Android adalah sebuah sistem operasi untuk perangkat *mobile* berbasis linux yang mencakup sistem operasi, *middleware* dan aplikasi.”

Menurut Wahana (2013:2), “Android adalah sebuah sistem operasi *mobile* yang berbasiskan pada versi modifikasi dari linux.”

Dari beberapa pendapat para ahli diatas dapat disimpulkan bahwa Android adalah sebuah sistem operasi hasil modifikasi dari linux yang diperuntukkan untuk perangkat bergerak (*mobile*) yang terdiri dari sistem operasi, *middleware* dan aplikasi.

2.2 Teori Khusus

2.2.1 Unified Modelling Language (UML)

Menurut Salahuddin (2014:137), “UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung”.

Pada perkembangan teknologi perangkat lunak, diperlukan adanya bahasa yang digunakan untuk memodelkan perangkat lunak yang akan dibuat dan perlu adanya standarisasi agar orang di berbagai negara dapat mengerti pemodelan perangkat lunak. UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak.



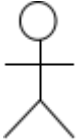

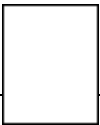

UML hanya berfungsi untuk melakukan pemodelan. Sehingga penggunaan UML tidak terbatas pada metodologi tertentu, meskipun pada kenyataannya UML paling banyak digunakan untuk metodologi berorientasi objek.

2.2.2 Use Case Diagram

Menurut Rosa dan Shalahuddin (2018:155), *use case diagram* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Dapat dikatakan *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.

Berikut adalah simbol-simbol yang ada pada diagram *use case*:

Tabel 2.1 Simbol *Use Case Diagram*

No	Gambar	Nama	Keterangan
1.		Aktor/ Actor	Aktor menggambarkan pengguna sistem, dapat berupa manusia atau sistem <i>terotomatisasi</i> lain yang berinteraksi dengan sistem lain untuk berbagi, mengirim, dan menerima informasi.
2.		Use Case	Simbol ini menggambarkan interaksi antara actor dengan software aplikasi tersebut.
3.		System Boundary	Menggambarkan batasan antara sistem dengan actor.
4.		Asosiasi/ Association	Menggambarkan hubungan antar aktor dan <i>use case</i> .

(Sumber : Indrajani, 2015:46)

2.2.3 Activity Diagram (Diagram Aktivitas)



Menurut Rosa dan Shalahuddin (2018:161), *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Berikut adalah simbol-simbol yang ada pada diagram aktivitas:

Tabel 2.2 Simbol Diagram Aktivitas

No	Simbol	Nama	Deskripsi
1.		Status awal	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
2.		Aktivitas	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3.		Percabangan / <i>decision</i>	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
4.		Penggabungan / <i>join</i>	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
5.		Status akhir	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
6.		<i>Swimlane</i>	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

(Sumber : Rosa dan Shalahuddin, 2018:162)

2.2.4 Class Diagram (Diagram Kelas)



Menurut Rosa dan Shalahuddin (2018:141), diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan *method* atau operasi. Berikut penjelasan atribut dan operasi:

1. Atribut merupakan variable-variabel yang dimiliki oleh suatu kelas.
2. Operasi atau *method* adalah fungsi-fungsi yang dimiliki oleh suatu kelas.

Berikut adalah simbol-simbol yang ada pada diagram kelas:

Tabel 2.3 Simbol Diagram Kelas

No	Simbol	Nama	Keterangan			
1.	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td>Class</td></tr> <tr><td>Attribute</td></tr> <tr><td>Operation</td></tr> </table>	Class	Attribute	Operation	Kelas	Kelas pada struktur sistem
Class						
Attribute						
Operation						
2.		Antarmuka/ <i>Interface</i>	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi obyek			
3.		Asosiasi/ <i>Association</i>	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>			
4.		Asosiasi berarah/ <i>directed</i>	Asosiasi antarkelas dengan makna kelas yang satu digunakan oleh kelas lain, asosiasi biasanya juga disertain dengan <i>multiplicity</i>			
5.		Generalisasi	Asosiasi antar kelas dengan makna generalisasi spesialisasi (umum – khusus)			
6.		Kebergantungan/ <i>dependency</i>	Relasi antar kelas dengan makna kebergantungan antarkelas			
7.		Agregasi / <i>aggregation</i>	Relasi antar kelas dengan makna semua bagian (<i>whole-part</i>)			



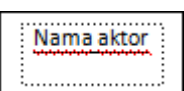

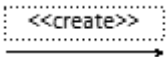
(Sumber : Rosa dan Shalahuddin, 2018:146)

2.2.5 Sequence Diagram (Diagram Sekuen)



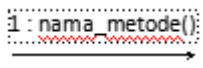
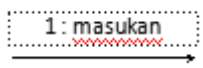
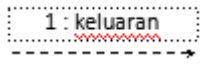
Menurut Rosa dan Shalahuddin (2018:165), *sequence diagram* menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek tersebut. Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada *use case*. Simbol-simbol yang digunakan dalam *sequence diagram*, yaitu :

Tabel 2.4 Simbol Diagram Sekuen

No	Simbol	Nama	Keterangan
1.		Garis hidup / <i>lifeline</i>	Menyatakan kehidupan suatu objek
2.	Atau  	Aktor	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor
3.		Waktu aktif	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya
4.		Pesan tipe <i>create</i>	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat

Lanjutan **Tabel 2.4** Simbol Diagram Sekuen



No	Simbol	Nama	Keterangan
5.		Pesan tipe <i>call</i>	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri, arah panah mengarah pada objek yang memiliki operasi/metode
6.		Pesan tipe <i>send</i>	Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim
7.		Pesan tipe keluaran	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian

(Sumber : Rosa dan Shalahuddin, 2018:165)

2.3 Teori Program

2.3.1 PHP Hypertext Preprocessor (PHP)

2.3.1.1 Pengertian PHP Hypertext Preprocessor (PHP)

Menurut Sidik dalam buku *Pemrograman Web dengan PHP7* (2017:4) PHP secara umum dikenal sebagai bahasa pemrograman *script-script* yang membuat dokumen HTML secara *on the fly* yang dieksekusi di *server web*, dokumen HTML yang dihasilkan dari suatu aplikasi bukan dokumen HTML yang dibuat dengan menggunakan editor teks atau editor HTML, dikenal juga sebagai bahasa pemrograman *server side*.

Menurut Winarno, dkk. dalam buku *Pemrograman Web Berbasis HTML5, PHP, dan JavaScript* (2014:49) PHP adalah sebuah bahasa pemrograman *web* berbasis *server (server-side)* yang mampu memarsing kode PHP dari kode *web* dengan ekstensi *.php*, sehingga menghasilkan tampilan *website* yang dinamis di sisi *client (browser)*.

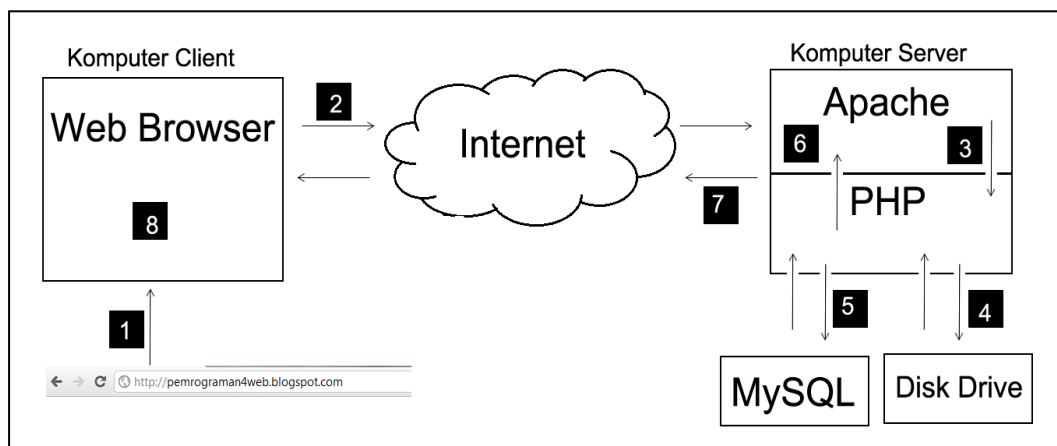


Menurut Raharjo dalam buku *Pemrograman WEB (HTML, PHP, & MySQL* (2016:38) PHP adalah salah satu bahasa pemrograman skrip yang dirancang untuk membangun aplikasi *web*.

Dari definisi diatas dapat disimpulkan bahwa PHP *Hypertext Preprocessor* dikenal sebagai bahasa pemrograman *server side* yang membuat dokumen HTML secara *on the fly* yang dieksekusi dari *server web* yang difungsikan untuk membangun suatu *web* dinamis atau aplikasi berbasis *web*.

2.3.1.2 Cara Kerja PHP

Menurut Raharjo dalam buku *Pemrograman WEB (HTML, PHP, & MySQL* (2016:39) menyatakan bahwa cara kerja aplikasi *web* yang ditulis dengan PHP dapat diilustrasikan dengan gambar di bawah ini.



Gambar 2.1 Cara Kerja PHP

Berikut adalah keterangan dari gambar di atas:

1. *User* menulis *http://pemrograman4web.blogspot.com* ke dalam *address bar* dari *web browser* (IE, Mozilla Firefox, Opera, dll)
2. *Web browser* mengirimkan pesan di atas ke komputer *server* (*http://pemrograman4web.blogspot.com*) melalui internet, meminta halaman *index.php* (halaman default dari *web*).
3. *Web server* (misalnya *Apache*), program yang berjalan di komputer *server*, akan menangkap pesan tersebut, lalu meminta *interpreter* PHP (program lain



yang juga berjalan di komputer *server*) untuk mencari file `index.php` dalam *disk drive*.

4. *Interpreter* PHP membaca file `index.php` dari *disk drive*.
5. *Interpreter* PHP akan menjalankan perintah-perintah atau kode PHP yang ada dalam file `index.php`. jika kode dalam file `index.php` melibatkan akses terhadap *database* (misalnya MySQL) maka *interpreter* PHP juga akan berhubungan dengan MySQL untuk melaksanakan perintah-perintah yang berkaitan dengan *database*.
6. *Interpreter* PHP mengirimkan halaman dalam bentuk HTML ke *Apache*.
7. Melalui internet, *Apache* mengirimkan halaman yang diperoleh dari *interpreter* PHP ke komputer user sebagai respon atas permintaan yang diberikan.
8. *Web browser* dalam komputer *user* akan menampilkan halaman dikirm oleh *Apache*.

2.3.2 Sublime Text

Sublime Text adalah aplikasi editor untuk kode dan teks yang dapat berjalan diberbagai platform *operating system* dengan menggunakan teknologi Phyton API. Terciptanya aplikasi ini terinspirasi dari aplikasi Vim, aplikasi ini sangatlah fleksibel dan *powerfull*. Fungsionalitas dari aplikasi ini dapat dikembangkan dengan menggunakan *sublime-packages*.

Sublime Text bukanlah aplikasi *open source* dan juga aplikasi yang dapat digunakan dan didapatkan secara gratis, akan tetapi beberapa fitur pengembangan fungsionalitas (*packages*) dari aplikasi ini merupakan hasil dari temuan dan mendapat dukungan penuh dari komunitas serta memiliki linsensi aplikasi gratis.

Sublime Text mendukung berbagai bahasa pemrograman dan mampu menyajikan fitur *syntax highlight* hampir di semua bahasa pemrogramman yang didukung ataupun dikembangkan oleh komunitas seperti; *C*, *C++*, *C#*, *CSS*, *D*, *Dylan*, *Erlang*, *HTML*, *Groovy*, *Haskell*, *Java*, *JavaScript*, *LaTeX*, *Lisp*, *Lua*, *Markdown*, *MATLAB*, *OCaml*, *Perl*, *PHP*, *Python*, *R*, *Ruby*, *SQL*, *TCL*, *Textile* and *XML*. Biasanya bagi bahasa pemrograman yang didukung ataupun belum



terdukung secara *default* dapat lebih dimaksimalkan atau didukung dengan menggunakan add-ons yang bisa di-*download* sesuai kebutuhan user.



Gambar 2.2 Logo *Sublime Text*

Berikut beberapa fitur yang diunggulkan dari aplikasi *Sublime Text*:

a. *Goto Anything*

Fitur yang sangat membantu dalam membuka file ataupun menjelajahi isi dari file hanya dengan beberapa keystrokes.

b. *Multiple Selections*

Fitur ini memungkinkan user untuk mengubah secara interaktif banyak baris sekaligus, mengubah nama variabel dengan mudah, dan memanipulasi file lebih cepat dari sebelumnya.

c. *Command Palette*

Dengan hanya beberapa keystrokes, user dapat dengan cepat mencari fungsi yang diinginkan, tanpa harus menavigasi melalui menu.

d. *Distraction Free Mode*

Bila user memerlukan fokus penuh pada aplikasi ini, fitur ini dapat membantu user dengan memberikan tampilan layar penuh.

e. *Split Editing*

Dapatkan hasil yang maksimal dari monitor layar lebar dengan dukungan editing perpecahan. Mengedit sisi file dengan sisi, atau mengedit dua lokasi di satu file. Anda dapat mengedit dengan banyak baris dan kolom yang user inginkan.



f. *Instant Project Switch*

Menangkap semua file yang dimasukkan kedalam project pada aplikasi ini. Terintegrasi dengan fitur *Goto Anything* untuk menjelajahi semua file yang ada ataupun untuk beralih ke file dalam *project* lainnya dengan cepat.

g. *Plugin API*

Dilengkapi dengan plugin API berbasis Python sehingga membuat aplikasi ini sangat tangguh.

h. *Customize Anything*

Aplikasi ini memberikan user fleksibilitas dalam hal pengaturan fungsional.

i. *Cross Platform*

Aplikasi ini dapat berjalan hampir disemua *operating system modern* seperti *Windows, OS X, dan Linux based operating system*.

2.3.2 MySQL

Menurut Gunawan (2010:15), MySQL adalah aplikasi atau sistem untuk mengelola *database* atau manajemen data. MySQL bertugas mengatur dan mengelola data-data pada *database*. MySQL menggunakan struktur atau kerangka yang berbentuk tabel. Dalam tabel-tabel itulah data diatur dan dikelompokkan.

Menurut Winarno, dkk. (2014:49), “MySQL merupakan tipe data relasional yang artinya MySQL menyimpan datanya dalam bentuk table-tabel yang saling berhubungan”.

Dari pengertian di atas dapat disimpulkan bahwa MySQL merupakan suatu aplikasi atau sistem yang digunakan untuk menyimpan dan memanajemen data dalam tabel-tabel yang saling berhubungan.

2.3.3 HTML

Menurut Enterprise dalam buku HTML, PHP, dan MySQL untuk pemula (2018:21), “HTML adalah bahasa markup (*markup language*) seperti yang ada di dalam singkatan HTML itu sendiri. Yang artinya HTML adalah bahasa struktur untuk menandai bagian-bagian dari sebuah halaman.”



Menurut Fauziah dalam buku Konsep Dasar Perancangan Sistem (2014:3), “HTML adalah jenis bahasa yang digunakan untuk membuat halaman *website* yang dengan menggunakan tag-tag yang telah dideklarasikan pada halaman notepad yang dapat saling berhubungan dengan dokumen HTML yang lainnya yang sering kita kenal dengan istilah link.”

Jadi dapat di simpulkan bahwa HTML adalah suatu bahasa *markup* yang digunakan untuk membuat suatu halaman *web*.

2.3.4 Pengertian Basis Data (*Database*)

Rusdiana dan Irfan (2014:302) menyatakan bahwa, *Database* adalah kumpulan informasi yang disimpan dalam komputer secara sistematis untuk memperoleh informasi dari basis data.

Adi dan Setiadi (2014:1250), menjelaskan bahwa Basis data (*database*) merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan di perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya. *Database* merupakan salah satu komponen yang penting dalam sistem informasi, karena merupakan basis dalam menyediakan informasi bagi pemakai. Penerapan *database* dalam sistem informasi disebut dengan *database* sistem.

Berdasarkan pendapat para ahli diatas dapat disimpulkan bahwa, basis data adalah kumpulan data yang disimpan di dalam sebuah tabel dan saling berhubungan sehingga data tersebut mudah untuk dicari dan didapatkan.

Menurut Pamungkas (2017:4) kriteria basis data meliputi:

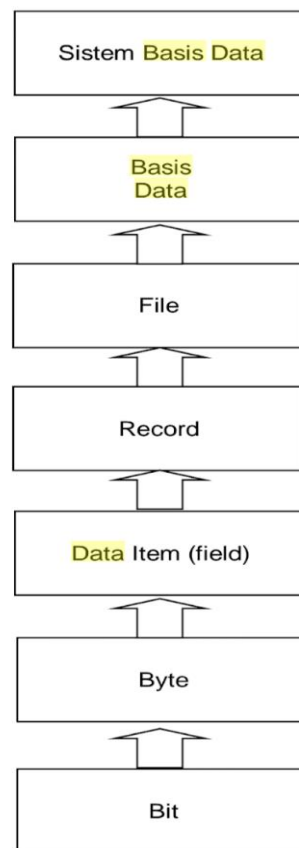
1. Berorientasi pada data dan bukan berorientasi pada program
2. Dapat digunakan oleh beberapa program aplikasi tanpa mengubah basis datanya.
3. Dapat berkembang dengan mudah, baik volume maupun strukturnya.
4. Dapat digunakan dengan cara berbeda-beda.
5. Kelengkapan data minimal.

Menurut Pamungkas (2017:6) komponen-komponen utama penyusun sistem basis data adalah:



1. Perangkat keras
2. Sistem operasi
3. Basis data
4. Sistem pengelola basis data (DBMS)
5. Pemakai (Programmer, user mahir, user umum, user khusus)

Menurut Pamungkas (2017:6) sistem basis data merupakan lingkup terbesar dalam organisasi data. Sistem basis data mencakup semua bentuk komponen data yang ada dalam suatu sistem. Sedangkan basis data merupakan komponen utama yang menyusun sistem basis data.



Gambar 2.3 Komponen Penyusun Sistem Basis Data

Keterangan:

1. *bit*, merupakan sistem angka biner yang terdiri atas angka 0 dan 1
2. *Byte*, merupakan bagian terkecil, dapat berupa karakter numerik, huruf, ataupun karakter khusus yang membentuk suatu item data / *field*. 1 *Byte* digunakan untuk mengkodekan 1 karakter.



3. Data item (*field*), merepresentasikan suatu atribut dari suatu *record* yang menunjukkan suatu item dari data, misalnya nama, alamat, kumpulan dari *field* membentuk suatu *record*.
4. *Record*, menggambarkan suatu unit data individu yang tertentu. Kumpulan dari *record* membentuk suatu file.
5. File, terdiri dari *record-record* yang menggambarkan satu kesatuan data yang sejenis.
6. Basis data, sekumpulan dari berbagai macam tipe *record* yang mempunyai hubungan terhadap suatu objek tertentu.
7. Sistem basis data, merupakan sekumpulan basis data, yang tersusun dari beberapa file.