BAB 2
TINJAUAN PUSTAKA

2.1. Gabah

Gabah adalah bulir padi. Biasanya mengacu pada bulir padi yang telah dipisahkan dari tangkainya (jerami). Gabah berasa dari bahasa kawa *gabah*. Dalam perdagangan komoditas gabah merupakan tahapan yang sangat penting dalam pengolahan padi sebelum dikonsumsi karena perdagangan padi dalam partai besar dilakukan dalam bentuk gabah. Terdapat defenisi perdagangan untuk gabah yaitu hasil tanaman padi yang telah dipisahkan dari tangkainya dengan cara perontokan.

2.2. Parameter Prediksi Harga Gabah

Dalam menentukan prediksi harga gabah, dinas pertanian berkerjasama dengan BPS untuk mengelola data – data pendukung yang diambil dari data sensus melalui prosedur yang dinamakan pengisian rekam data nulanan. Rekam data ini mencakup pengolahan data di berbagai sektor, mulai dari tingkat kepadatan penduduk, ekonomi daerah hingga sekotr pertanian. Pada sektor pertanian inilah, data pendukung tersebut dijadikan parameter untuk menentukan prediksi harga gabah. Berdasarkan observasi langsung dan wawancara terhadap pihak terkait, data parameter itu terdiri dari Kadar air dan Kadar hampa/kotoran. Kada air merupakan jumlah kadar air yang terdpat didalam gabah (%), sedangkan kadar hampa/kotoran merupakan banyaknya kotoran yang terdapat pada gabah (%).

2.3. Artificial Intelligence

Kecerdasan buatan sering disebut *Artificial Intelligence* (AI), kecerdasan *artificial*, *intelijensia artificial*, atau *intelijensia* buatan. Para ahli mendefinisikan AI secara berbeda-beda tergantung pada sudut pandang masing-masing. Ada yang fokus pada logika berfikir manusia saja, tetapi ada juga yang mendefinisikan AI secara lebih luas pada tingkah laku manusia. Stuart Russel dan Peter Norvig

mengelompokkan definisi AI yang diperoleh dari beberapa *textbook* berbeda kedalam empat kategori [8], yaitu:

1. *Thingking Humanly*

   Pendekatan ini dilakukan dengan du acara sebagai berikut:

   a. Melalui intropeksi: mencoba menangkap pemikiran-pemikiran pada saat berfikir.
   b. Melalui eksperimen-eksperimen psikologi.

2. *Acting Humanly*

   Pada tahun 1950, Alan Turing merancang suatu kajian bagi komputer berintelijensia untuk menguji apakah komputer tersebut mampu mengelabuhi seorang manusia yang mengintrogasinya melalui telepaty (komunikasi berbasis teks jarak jauh). Jika interrogator tidak dapat membedakan yang diinterogasi adalah manusia atau komputer, maka komputer berintelijensia tersebut lolos dari *Turing test*. Komputer tersebut perlu memiliki kemampuan *Natural Language Processing*, *Knowledge Representation, Automated Reasoning*, *Machine Learning*, *Computer Vision*, dan *Robotic*. Turing Test sengaja menghindari interaksi fisik antara interrogator dan komputer karena simulasi fisik manusia tidak memerlukan *intelijensia.*

3. *Thingking Rationally*

   Terdapat dua masalah dalam pendekatan ini, yaitu:

   a. Tidak mudah untuk membuat pengetahuan informal dan menyatakan pengetahuan tersebut ke dalam formal term yang diperlukan oleh notasi logika, khususnya ketika pengetahuan tersebut memiliki kepastian kurang dari 100%
   b. Terdapat perbedaan besar antara dapat memecahkan masalah “dalam prinsip” dan memecahkannya “dalam dunia nyata”

4. *Acting Rationally*

   Membuat Inferensi yang logis merupakan bagian dari suatu rational agent. Hal ini disebabkan satu-satunya cara untuk melakukan aksi secara rasional adalah dengan menalar secara logis. Dengan menalar

secara logis, maka bisa didapatkan kesimpulan bahwa aksi yang diberikan akan mencapai tujuan atau tidak. Jika mencapai tujuan, maka agent dapat melakukan aksi berdasarkan kesimpulan tersebut.

2.4. Sistem Pakar

Sistem pakar adalah sistem berbasis komputer yang menggunakan pengetahuan, fakta, dan teknik penalaran dalam memecahkan masalah yang biasanya hanya dapat dipecahkan oleh seorang pakar dalam bidang tersebut. Sistem pakar memberikan nilai tambah pada teknologi untuk membantu dalam menangani era informasi yang semakin canggih.

Konsep dasar dari suatu sistem pakar mengandung beberapa unsur, diantaranya adalah keahliah, ahli, pengalihan keahlian, inferensi, aturan dan kemampuan menjelaskan. Keahlian merupakan salah satu penguasaan pengetahuan di bidang tertentu yang didapatkan baik secara formal maupun non formal. Ahli adalah seseorang yang mempunyai pengetahuan tertentu dan mampu menjelaskan suatu tanggapan dan mempunyai keinginan untuk belajar yang bertujuan memperbaharui pengetahuan dalam bidangnya. Pengalihan keahlian adalah mengalihkan keahlian dari seorang pakar dan kemudian dialihkan lagi ke orang yang bukan ahli atau orang awam yang membutuhkan. Sedangkan inferensi, merupakan suatu rangkaian proses untuk menghasilkan informasi dari fakta yang diketahui atau diasumsikan.

Tujuan pengembangan sistem pakar sebenarnya tidak untuk menggantikan peran para pakar, namun untuk mengimplementasikan pengetahuan para pakar ke dalam bentuk perangkat lunak, sehingga dapat digunakan oleh orang banyak dan tanpa biaya yang besar. Untuk membangun sistem yang difungsikan untuk menirukan seorang pakar, manusia harus bisa melakukan hal – hal yang dapat dikerjakan oleh para pakar. Untuk membangun sistem yang seperti itu maka komponen – komponen dasar yang harus dimiliki adalah:

1. Antar Muka (*User Interface*)
2. Basis Pengetahuan (*Konwledge Base*)
3. Mesin Inferensi (*Inference Engine*)

Kaidah produksi merupakan salah satu model untuk merepresentasikan pengetahuan. Kaidah produksi menjadi acuan yang sangat sering digunakan oleh sistem inferensi. Kaidah produksi dituliskan dalam bentuk pernyataan ***IF – THEN*** (jika maka). Pernyataan ini menghubungkan bagian premis (***IF***) dan bagian kesimpulan (***THEN***) yang dituliskan dalam bentuk sebagai berikut:

IF [premis] ***THEN*** [konklusi]

Kaidah ini dapat dikatakan sebagai suatu implikasi yang terdiri dari dua bagian, yaitu premis dan bagian konklusi. Apabila bagian premis dipenuhi maka bagian konklusi akan bernilai benar. Bagian premis dalam aturan produksi dapat memiliki lebih dari satu proposisi. Proposisi tersebut dihubungkan dengan menggunakan operator logika ***AND*** atau ***OR***. Sebagai contoh:

IF gejala is Gatal-Gatal

AND Berbintik-bintik/membengkak merah

THEN Dermatitis Alergi

Sistem pakar memiliki manfaat dan kemampuan sebagai berikut:

1. Meningkatkan Produktivitas, karena Sistem Pakar dapat bekerja lebih cepat daripada manusia.
2. Membuat seorang yang awam bekerja layaknya seorang Pakar.
3. Meningkatkan kualitas, dengan memberi nasehat/pengarahan yang konsisten dan mengurangi kesalahan.
4. Mampu menangkap pengetahuan dan kepakaran seseorang.
5. Dapat beroperasi di lingkongan yang berbahaya.
6. Memudahkan akses pengetahuan seorang pakar.
7. Sistem pakar tidak pernah menjadi bosan, kelelahan ataupun sakit
8. Meningkatkan kapabilitas sistem computer. Integrasi Sistem Pakar dengan sistem komputer lain membuat sistem lebih efektif dan mencakup lebih banyak aplikasi
9. Mampu bekerja dengan informasi yang tidak lengkap atau tidak pasti. Berbeda dengan sistem komputer konvensional, sistem pakar dapat bekerja dengan informasi yang tidak lengkap. Pengguna dapat merespn dengan : “tidak tahu” atau “tidak yakin” pada satu atau lebih

pertanyaan selama konsultasi dan Sistem Pakar tetap akan memberikan jawabanya.

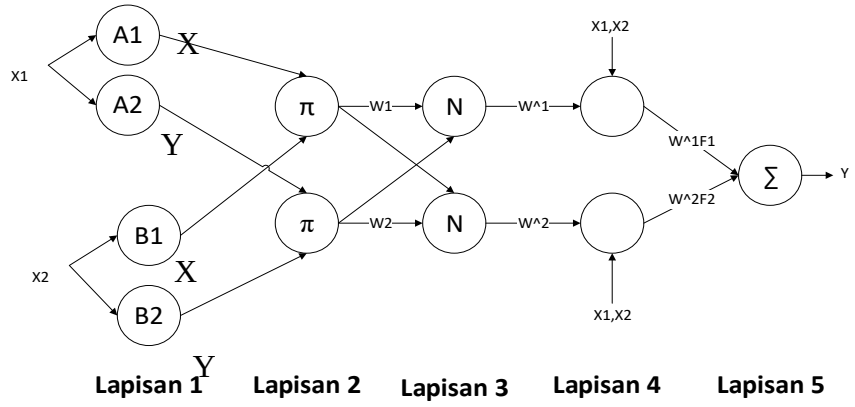
10. Sistem pakar dapat digunakan sebagai media pelangkap pelatihan. Penggunapemula yang bekerja dengan sistem pakar akan menjadi lebih berpengalaman karena adanya fasilitas penjelas yang berfungsi sebagai guru.
11. Meningkatkan kemampuan untuk menyelesaikan masalah karena Sistem Pakar mengambil sumber pengetahuan dari banyak pakar.

2.5. *Adaptive Neuro Fuzzy Inference System* (ANFIS)

Adaptive Neuro Fuzzy Inference System (ANFIS) adalah arsitektur yang secara fungsional sama dengan *fuzzy rule base* model Sugeno. Arsitektur ANFIS juga sama dengan jaringan syaraf dengan fungsi radial dengan sedikit batasan tertentu. Bisa dikatakan bahwa ANFIS adalah suatu metode yang mana dalam melakukan penyetelan aturan digunakan algoritma pembelajaran terhadap sekumpulan data. Pada ANFIS juga memungkinkan aturan-aturan untuk beradaptasi.

2.5.1. Langkah-langkah ANFIS

System inferensi fuzzy model Tagaki-Sugeno-Kang (TSK) direpresentasikan sebagai jaringan syaraf *feed forward* lima lapis khusus pada struktur ANFIS yang digambarkan dalam arsitektur jaringan syaraf tiruan [9] seperti pada gambar 2.1 berikut:

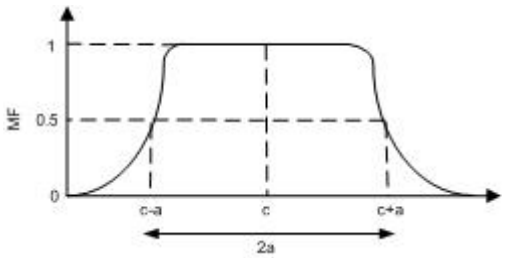


Gambar 2. 1 Struktur ANFIS

Pada gambar 2.1 terlihat sistem *inferensi fuzzy* yang digambarkan pada arsiterktur jaringan syaraf tiruan terdiri atas lima lapisan dengan fungsi yang berbeda-beda. Berikut merupakan penjelasan dari tiap lapisan yang ada dalam struktur ANFIS:

1. Lapisan 1

Semua simpul pada lapisan ini adalah simpul adaptif (parameternya dapat berubah). Parameter a,b,c pada fungsi keanggotaan gbell dinamakan parameter premis yang adaptif, dengan b yang biasa nya bernilai 1 untuk menentukan nilai keanggotaan bell tidak terbalik. Berikut gambar kurva keanggotaan *Generalized Bell* yang dapat dilihat pada gambar 2.2, sementara untuk persamaan gbellnya dapat dilihat pada persamaan 2.1.



Gambar 2. 2 Kurva *Generalized-Bell*

$$gbell(x, a, b, c) = \frac{1}{1 + \left|\frac{x - c}{a}\right|^{2b}} \qquad \ldots (2.1)$$

Koefisien b: koefisien adalah angka positif, dan biasanya bernilai 1. Jumlah tersebut terkait dengan bagian atas kurva dan menandakan bahwa kurva berbentuk lonceng dan penyebab naik ke bagian datar. Koefisien a menyatakan lebar dari titik pusat hingga batas keanggotaan dan koefisien c menyatakan titik pusat *membership function*. Nilai a,b,c dan jumlah himpunan masukan fuzzy (x) ditentukan pada awal proses dan akan berubah melalui proses pembelajaran dinamakan parameter premis yang adaptif.

2. Lapisan 2

Semua simpul pada lapisan ini adalah nonadaptif (parameter tetap). Fungsi simpul ini adalah mengalikan setiap sinyal masukan yang datang. Fungsi simpul:

$$O_{2.i} = w_i = \mu_{A_i}(x).\mu_{B_i}(y), untuk\ i = 1,2 \quad \ldots(2.2)$$

Keterangan :

x	=	Nilai masukan
y	=	Nilai masukan
w_i	=	Bobot ke-i
μ_i	=	Nilai keanggotaan kelas-i
$O2_{,i}$	=	Nilai keluaran lapisan 2 ke-i

Tiap keluaran simpul menyatakan derajat pengaktifan (*firing strength*) tiap aturan *fuzzy*. Fungsi ini dapat diperluas apabila bagian premis memiliki lebih dari dua himpunan *fuzzy*. Banyaknya simpul pada lapisan ini menunjukkan banyaknya aturan yang dibentuk. Fungsi perkalian yang digunakan adalah interpretasi kata hubung *and* dengan menggunakan operator *t-norm*.

3. Lapisan 3

Lapisan Setiap simpul pada lapisan ini adalah simpul nonadaptif yang menampilkan fungsi derajat pengaktifan ternomalisasi (*normalized firing strength*) yaitu rasio keluaran simpul ke-i pada lapisan sebelumnya terhadap seluruh keluaran lapisan sebelumnya, dengan bentuk fungsi simpul:

$$O_{3.i} = \bar{w}_i = \frac{w_i}{w_1 + w_2}, untuk\ i = 1,2 \quad \ldots(2.3)$$

Keterangan :

w_i	=	Bobot ke-i
$\bar{w}_i$	=	derajat pengaktifan ternomalisasi ke-i
$O_{3,i}$	=	Nilai keluaran lapisan 3 ke-i

Apabila dibentuk lebih dari dua aturan, fungsi dapat diperluas dengan membagi w_i dengan jumlah total w untuk semua aturan.

4. Lapisan 4

Setiap simpul pada lapisan ini adalah simpul adaptif dengan fungsi simpul:

$$O_{4,i} = \bar{w}_i f_i = \bar{w}_i (p_i x + q_i y + r_i) \quad \ldots (2.4)$$

Dengan $\bar{w}_i$ adalah derajat perngaktifan ternormalisasi dari lapisan 3 dan parameter p, q, r menyatakan parameter konsekuen yang adaptif.

Keterangan:

$\bar{w}_i$	=	derajat pengaktifan ternomalisasi ke-I
f_i	=	sinyal kontrol ke –i
$O_{4,i}$	=	Nilai keluaran lapisan 4 ke-i

5. Lapisan 5

Pada lapisan ini hanya ada satu simpul tetap yang fungsinya untuk menjumlahkan semua masukan. Fungsi simpul:

$$O_{5,i} = \sum_i \bar{w}_i f_i \quad \ldots (2.5)$$

Keterangan:

$\bar{w}_i$	=	derajat pengaktifan ternomalisasi ke-I
f_i	=	sinyal kontrol ke –i
$O_{5,i}$	=	Nilai keluaran lapisan 5 ke-i

Jaringan adaptif dengan lima lapisan tersebut ekuivalen dengan system inferensi fuzzy TSK.

2.5.2 Proses Belajar ANFIS

Proses belajar yang ada digunakan untuk memperbaharui parameter premis dan parameter konsukuen yang ada pada jaringan ANFIS. Pembelajaran *hybrid* yang dilakukan terdiri atas dua bagian yaitu arah maju (*forward pass*) dan arah mundur (*backward pass*).

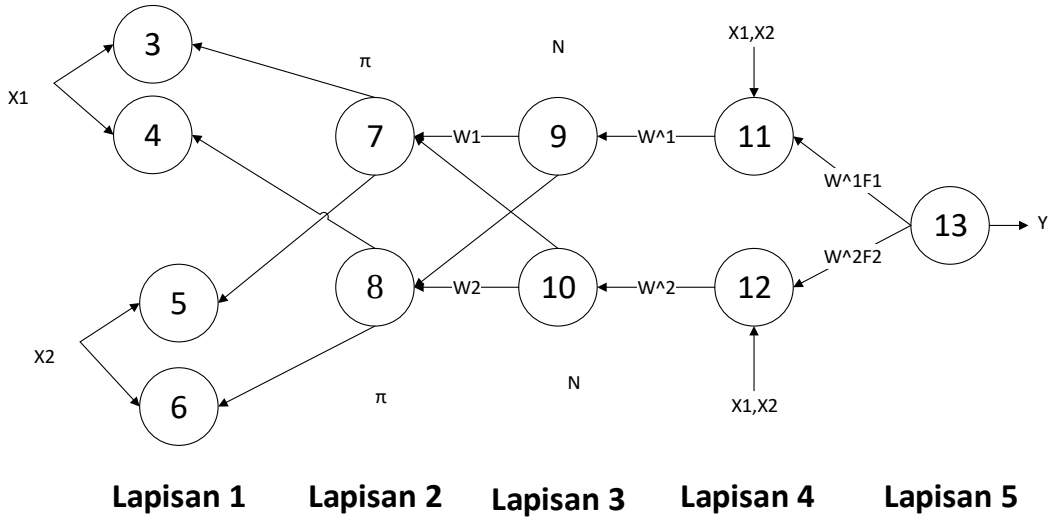
Pada arah maju, parameter premis dibuat tetap. Dengan menggunakan metode *Recursive Least Square Estimator* (RLSE), parameter konsekuen diperbaiki berdasarkan pasangan data masukan – keluaran. Metode RLSE dapat diterapkan karena parameter konsekuen yang diperbaiki adalah parameter linier. Metode RLSE akan mempercepat proses belajar *hybrid*. Kemudian setelah parameter konsekuen didapatkan, data masukan dilewatkan jaringan adaptif

kembali dan hasil keluaran jaringan adaptif ini dibandingkan dengan keluaran yang sebenarnya [7]. Karena ANFIS menggunakan salah satu metode training yaitu propagasi-balik maka istilah yang digunakan pada training adalah epoch. Epoch berbeda dari iterasi, epoch terdiri dari 2 tahap yaitu arah maju dan arah mundur, sedangkan untuk iterasi itu hanya terjadi pada 1 tahapan. Pada arah mundur, parameter konsekuen dibuat tetap. Proses pembelajaran hibrid ANFIS dapat dilihat pada tabel 2.1.

Tabel 2. 1 Proses Pembelajaran Hybrid ANFIS

	Arah Maju	**Arah Mundur**
Parameter Premis	Tetap	*Backpropagation-Error*
Parameter Konsekuen	LSE	Tetap

Kesalahan yang terjadi antara keluaran jaringan *adaptif* dan keluaran sebenarnya dipropagasikan balik dengan menggunakan *gradient descent* untuk memperbaiki parameter premis. Pembelajaran ini dikenal sebagai algoritma *Backpropagation-Error*. Berikut arsitektur jaringan pada saat menghitung *Backpropagation-Error* dapat dilihat pada gambar 2.3:



Gambar 2. 3 Arsitektur Jaringan *Backpropagation-error*

1. *Error* pada lapisan ke-5

Apabila kita memiliki jaringan seperti pada Gambar 2.3, yang hanya memiliki 1 simpul pada lapisan *output* (simpul 13), maka propagasi *error* yang menuju pada lapisan ke-5 dapat dirumuskan sebagai berikut:

$$\varepsilon_{13} = \frac{\partial E_p}{\partial X_{13}} = -2(y_p - y_p^*) \quad \dots (2.6)$$

Dengan y_p adalah target *output* data pelatihan ke-p, dan y_p^* adalah output jaringan pada data pelatihan ke-p

2. *Error* pada lapisan ke-4

Pada lapisan 4 tidak dilakukan perhitungan *error* ini dikarenakan pada alur mundur tidak terjadi update nilai parameter konsekuen yang terdapat pada lapisan 4 atau bisa dikatakan tetap.

3. *Error* pada lapisan ke-3

Dengan melihat gambar 2.3, propagasi *error* yang menuju pada lapisan 3, yaitu simpul 9 dan simpul 10, sehingga dapat dirumuskan sebagai berikut:

$$\varepsilon_9 = \left(\frac{\partial E_p}{\partial X_{13}}\right)\left(\frac{\partial f_{13}}{\partial X_{11}}\right)\left(\frac{\partial f_{11}}{\partial X_9}\right) = \varepsilon_{11}\left(\frac{\partial f_{11}}{\partial X_9}\right) \quad \dots (2.9)$$

$$\varepsilon_{10} = \left(\frac{\partial E_p}{\partial X_{13}}\right)\left(\frac{\partial f_{13}}{\partial X_{12}}\right)\left(\frac{\partial f_{12}}{\partial X_{10}}\right) = \varepsilon_{12}\left(\frac{\partial f_{12}}{\partial X_{10}}\right) \quad \dots (2.10)$$

4. *Error* pada lapisan ke-2

Dengan melihat gambar 2.3, propagasi *error* yang menuju pada lapisan ke-2 yaitu simpul 7 dan simpul 8, sehingga dapat dirumuskan sebagai berikut:

$$\varepsilon_7 = \left(\frac{\partial E_p}{\partial X_{13}}\right)\left(\frac{\partial f_{13}}{\partial X_{11}}\right)\left(\frac{\partial f_{11}}{\partial X_9}\right)\left(\frac{\partial f_9}{\partial X_7}\right) + \left(\frac{\partial E_p}{\partial X_{13}}\right)\left(\frac{\partial f_{13}}{\partial X_{12}}\right)\left(\frac{\partial f_{12}}{\partial X_{10}}\right)\left(\frac{\partial f_{10}}{\partial X_7}\right) \quad (2.11)$$

$$\varepsilon_8 = \left(\frac{\partial E_p}{\partial X_{13}}\right)\left(\frac{\partial f_{13}}{\partial X_{12}}\right)\left(\frac{\partial f_{12}}{\partial X_{10}}\right)\left(\frac{\partial f_{10}}{\partial X_8}\right) + \left(\frac{\partial E_p}{\partial X_{13}}\right)\left(\frac{\partial f_{13}}{\partial X_{11}}\right)\left(\frac{\partial f_{11}}{\partial X_9}\right)\left(\frac{\partial f_9}{\partial X_8}\right) \quad \dots (2.12)$$

5. *Error* pada lapisan ke-1

Dengan melihat gambar 2.3, propagasi *eror* yang menuju pada lapisan 1, yaitu simpul 3, 4, 5, dan 6 sehingga dapat dirumuskan sebagai berikut:

$$\varepsilon_3 = \varepsilon_7\left(\left(\frac{\partial f_7}{\partial X_3}\right)\right) \quad \ldots (2.13)$$

$$\varepsilon_4 = \varepsilon_8\left(\left(\frac{\partial f_8}{\partial X_4}\right)\right) \quad \ldots (2.14)$$

$$\varepsilon_5 = \varepsilon_7\left(\left(\frac{\partial f_7}{\partial X_5}\right)\right) \quad \ldots (2.15)$$

$$\varepsilon_6 = \varepsilon_8\left(\left(\frac{\partial f_8}{\partial X_6}\right)\right) \quad \ldots (2.16)$$

2.6 Perancangan Terstruktur

Perancangan ini bertujuan untuk membuat model solusi terhadap masalah yang sudah dimodelkan secara lengkap pada tahap analisis struktur. Tahap analisis ini meliput perancangan ERD, Diagram Konteks, DFD dan Kamus Data [10]. Berikut penjelasan dari setiap poinnya.

2.6.1 *Entity Relationship Diagram* (ERD)

ERD adalah diagram grafikal keseluruan struktur logikadari sebuah basis data. ERD tidak menggambarkan aliran data atau proses data, tetapi ERD lebih menggambarkan data pada data *store.* ERD ini berfungsi untuk menggambarkan relasi dari dua tabel yang dapat digolongkan dalam tiga macam bentuk relasi, yaitu relasi satu ke satu, satu kebanyak dan relasi banyak ke banyak.

Model ERD yang berisi komponen-komponen himpunan entitas dan himpunan relasi yang masing-masing dilengkapi dengan atribut-atribut yang mempresentasikan seluruh fakta yang ditinijau.

2.6.2 Diagram Konteks

Diagram Konteks adalah sebuah diagram yang mencakup masukan-maskan dasar, sistem umum dan keluaran. Diagram ini merupakan tingkatan tertinggi dalam diagram aliran data dan hanya memuat satu proses. Menunujukan sistem secara keseluruhan, diagram konteks tidak memuat penyimpanan dan penggambaran aliran data yang sederhana. Semua entitas luar yang ditunjukan pada diagram konteks berikut aliran data-aliran data utama yang menuju dan dari sistem.

2.6.3 *Data Flow Diagram* (DFD)

Data Flow Diagram (DFD) adalah representasi grafik dari sebuah sistem. DFD menggambarkan komponen-komponen sebuah sistem, aliran-aliran data dari komponen-komponen yang ada mulai dari aliran asal, tujuan, dan penyimpanan dari data tersebut.

DFD digunakan untuk dua hal utama, yaitu untuk membuat dokumentasi dari sistem yang ada atau untuk menyusun dokumentasi untuk sistem yang baru. DFD merupakan alat bantu dari pengembangan sebuah bantu dari pengembangan sebuah sistem yang dibangun secara terstruktur atau procedural.

2.6.4 Kamus Data

Kamus data adalah katalog fakta tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi. Kamus data dapat mendefinisikan data yang mengalir pada sistem dengan lengkap. Kamus data juga dapat digunakan pada tahap analisa dan perancangan sistem. Dengan adanya kamus data, didapat definisi-definisi dari bentuk-bentuk yang kurang dimengerti dalam isi DFD seperti aliran data, file, proses, dan elemen-elemen data.

2.6.5 *Flowchart*

Flowchart adalah penggambaran secara grafik dari langkah-langkah dan urut-urutan prosedur dari suatu program. *Flowchart* menolong *analyst* dan *programmer* untuk memecahkan masalah kedalam segmen-segmen yang lebih kecil dan menolong dalam menganalisis alternatif-alternatif lain dalam pengoperasian. *Flowchart* biasanya mempermudah penyelesaian suatu masalah khususnya masalah yang perlu dipelajari dan dievaluasi lebih lanjut.

Flowchart adalah bentuk gambar/diagram yang mempunyai aliran satu atau dua arah secara *sequensial*. *Flowchart* digunakan untuk merepresentasikan maupun mendesain program. Oleh karena itu *flowchart* harus bisa merepresentasikan komponen-komponen dalam bahasa pemrograman.

2.7 Bahasa Pemograman

Bahasa pemrograman diperlukan untuk menjalankan instruksi-instruksi apa yang harus dilakukan komputer. Komputer tidak bisa memahami bahasa manusia, sehingga diperlukan penggunaan bahasa komputer di dalam program komputer. Penelitian ini menggunakan bahasa pemrograman C# untuk mengembangkan aplikasi dan habasa pemrograman MySql untuk mengelola database.

2.7.1 Bahasa Pemograman C#

C# (dibaca: C sharp) merupakan sebuah bahasa pemograman yang berorientasi objek yang dikembangkan oleh Microsoft sebagai bagian dari inisiatif kerangka .NET Framework. Bahasa pemograman ini dibuat berbasiskan Bahasa C++ yang telah dipengaruhi aspek – aspek ataupun fitur bahasa yang terdapat pada bahasa – bahasa pemograman lainnya seperti Java, Delphi, Visual Basic dan lain – lain dengan beberapa penyederhanaan. Menurut standar ECMA-334 C# Language Spesification, nama C# terdiri atas sebuah huruf latin C (U+0043) yang diikuti oleh tanda pagar yang menandakan angka # (U+0023). Tanda pagar # yang digunakan bukan tanda kres dalam seni music (U+266F), dan tanda pagar #(U+0023) tersebut digunakan karena karakter kres dalam seni music tidak terdapat didalam keyboard standar.

2.7.2 MySQL

Perkembangannya disebut SQL yang merupakan kepanjangan dari Structured Query language, SQL merupakan bahasa terstruktur yang khusus digunakan untuk mengolah database, SQL pertama kali didefinisikan oleh American National Institute (ANSI) pada tahun 1986. MySQL adalah sebuah sistem manajemen database yang bersifat open source, MySQL pasangan serasi dari PHP, MySQL dibuat dan dikembangkan oleh MySQL AB yang berada di swedia.

MySQL dapat digunakan untuk mengelola database mulai dari yang kecil sampai dengan yang besar, MySQL juga dapat menjalankan perintah-perintah Structured Query Language (SQL) untuk mengelola database-database yang ada didalamnya, hingga kini MySQL sudah berkembang hingga versi 5, MySQL versi 5 sudah mendukung trigger untuk memudahkan pengelolaan tabel dalam database.

Sebuah website yang interaktif dan dinamis tentunya akan membutuhkan penyimpanan dara yang fleksibel dan cepat untuk diakses. Salah satu database untuk sever adalah MySQL, MySQL merupakan aplikasi RDBMS (*Relational Data Base Management System*) yang sangat cepat dan dapat diandalkan dalam menangani basis data. Jenis database ini sangat populer dan dapat digunakan pada banyak website, MySQL menggunakan bahasa SQL (*Structured Query Language*) yang bersifat free (gratis atau tidak perlu membayar untuk menggunakannya), selain itu MySQL dapat berjalan di berbagai platform diantaranya, Linux, Windows dan lain sebagainya.

2.8 Aplikasi

Aplikasi merupakan perangkat lunak siap pakai yang dapat digunakan untuk menjalankan perintah-perintah dari para pengguna aplikasi tersebut. Dalam penelitian ini digunakan aplikasi untuk mendukung penelitian yang dilakukan meliputi Xampp dan Visual Studio.

2.8.1 Xampp

Xampp merupakan software web server apache yang di dalamnya sudah tersedia database server mysql dan support php programming. Xampp juga mudah digunakan, gratis dan mendukung instalasi Linux dan Windows. Salah satu keuntungan xampp yaitu instalasi yang dilakukan satu kali dan sudah tersedia Apache Web Server, MySql Database Server, PHP Support, FileZilla, dan modul lainnya. Versi Xampp yang digunakna dalam penelitian ini adalah versi 5.6.19-0. Kemudian fitur yang digunakan adalah fitur MySQL dan Apache Web Server.

2.8.2 Microsoft Visual Studio

Microsoft Visual Studio merupakan sebuah perangkat lunak lengkap yang dapat digunakan untuk melakukan pengembangan aplikasi, baik itu aplikasi bisnis maupun personal, ataupun komponen aplikasinya, dalam bentuk aplikasi console, aplikasi windows, ataupun aplikasi Web, Visual Studio mencakup compiler, SDK, Integrated Development Envoirment (IDE), dan dokumentasi (umumnya berupa MSDN Library). Kompiler yang dimasukkan ke dalam paket Visual Studio antara lain Visual C++, Visual C#, Visual Basic, Visual Basic .NET, Visual InterDev, Visual J++, Visual J#, Visual FoxPro, dan Visual SourceSafe.

Microsoft Visual Studio dapat digunakan untuk mengembangkan aplikasi dalam native code (dalam bentuk bahasa mesin yang berjalan di atas Windows) ataupun managed code (dalam bentuk Microsoft Intermediate Language di atas .NET Framework). Selain itu, Visual Studio juga dapat digunakan untuk mengembangkan aplikasi Silverlight, aplikasi Windows Mobile (yang berjalan diatas .NET Compact Framework).

Visual Studio kini telah menginjak versi Visual Studio 9.0.21022.08, atau dikenal dengan sebutan Microsoft Visual Studio 2008 yang diluncurkan pada 19 November 2007, yang ditujukan untuk platform Microsoft .NET Framework 3.5. Versi sebelumnya, Visual Studio 2005 ditujukan untuk platform .NET Framework 2.0 dan 3.0. Visual Studio 2003 ditujukan untuk .NET Framework 1.1, dan Visual Studio 2002 ditujukan untuk .NET Framework 1.0.

Versi-versi tersebut di atas kini dikenal dengan sebutan Visual Studio .NET, karena memang membutuhkan Microsoft .NET Framework. Sementara itu, sebelum muncul Visual Studio .NET terdapat Microsoft Visual Studio 6.0 (VS1998).

2.9 Pengujian Sistem

Pengujian adalah proses pemeriksaan atau evaluasi sistem atau komponen sistem secara manual atau otomatis untuk memverikasi apakah sistem memenuhi kebutuhan-kebutuhan yang dispesifikan atau mengidentifikasi perbedaan

perbedaan antara hasil yang diharapkan dengan hasil yang terjadi [12]. Pengujian seharusnya meliputi tiga konsep berikut.

1. Demonstrasi validitas perangkat lunak pada masing-masing tahap di siklus pengembangan sistem
2. Penentuan validitas sistem akhir dikaitkan dengan kebutuhan pemakai
3. Pemeriksaan perilaku sistem dengan mengeksekusi sistem pada data sampel pengujian

Awalnya pengujian diartikan sebagai aktivitas yang dapat atau hanya dilakukan setelah pengkodean (kode program selesai). Namun, pengujian seharusnya dilakukan dalam skala lebih luas. Pengujian dapat dilakukan begitu spesifikasi kebutuhan telah dapat didefinisikan. Evaluasi terhadap spesifikasi dan perancangan juga merupakan teknik di pengujian. Kategori pengujian dapat dikategorikan menjadi dua [12], yaitu :

1. Berdasarkan ketersediaan logik sistem, terdiri dari Black box testing dan White box testing
2. Berdasarkan arah pengujian, terdiri dari Pengujian top down dan Pengujian bottom up

2.9.1 Pengujian Black Box

Konsep *black box* digunakan untuk merepresentasikan sistem yang cara kerja di dalamnya tidak tersedia untuk diinspeksi. Di dalam *black box*, item-item yang diuji dianggap "gelap" karena logiknya tidak diketahui, yang diketahui hanya apa yang masuk dan apa yang keluar dari *black box* [11].

Pada pengujian *black box*, kasus-kasus pengujian berdasarkan pada spesifikasi sistem. Rencana pengujian dapat dimulai sedini mungkin di proses pengembangan perangkat lunak. Teknik pengujian konvensional yang termasuk pengujian "*black box*" adalah sebagai berikut [11].

1. Graph-based testing
2. Equivalence partitioning
3. Comparison testing
4. Orthogonal array testing

Pada pengujian *black box*, kita mencoba beragam masukan dan memeriksa keluaran yang dihasilkan. Kita dapat mempelajari apa yang dilakukan kotak, tapi tidak mengetahui sama sekali mengenai cara konversi dilakukan. Teknik pengujian *black box* juga dapat digunakan untuk pengujian berbasis skenario, di mana isi dalam sistem mungkin tidak tersedia untuk diinspeksi tapi masukan dan keluaran yang didefinisikan dengan *use case* dan informasi analisis yang lain.

2.9.2 Pengujian RMSE

Pengujian RMSE merupakan pengujian sistem pada saat pelatihan dan pengujian dengan menggunakan batasan *epoch*, pengujian berfungsi untuk mendapatkan nilai RMSE terkecil dati tiap-tiap data latih. Hasil dari data latih tersebut akan dipilih yang mana yang memiliki nilai RMSE terkecil kemudian nilai parameter update dari pelatihan tersebut dipakai pada saat pengujian.