BAB 2
TINJAUAN PUSTAKA

2.1 Artificial Intelligence

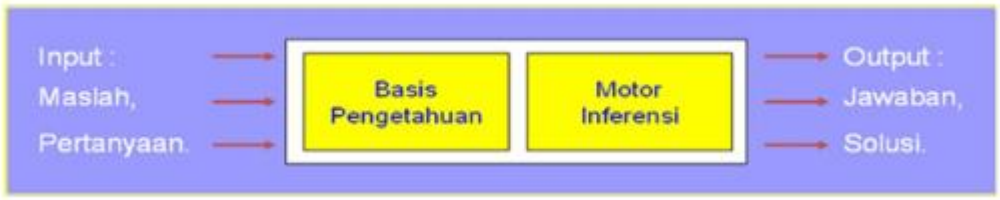
Kecerdasan buatan (*artificial intelligence*) merupakan salah satu bagian dari ilmu komputer yang mempelajari bagaimana membuat mesin (komputer) yang dapat melakukan pekerjaan seperti dan sebaik yang dilakukan oleh manusia bahkan bisa lebih baik daripada yang dilakukan oleh manusia [7].

Menurut John McCarthy pada tahun 1956, kecerdasan buatan adalah untuk mengetahui dan memodelkan proses – proses berpikir manusia dan mendesain mesin agar dapat menirukan perilaku manusia. Cerdas, berarti memiliki pengetahuan ditambah pengalaman, penalaran, dan moral yang baik. Manusia cerdas dalam menyelesaikan permasalahan karena manusia mempunyai pengetahuan dan pengalaman. Pengetahuan diperoleh dari belajar sehingga semakin banyak bekal pengetahuan yang dimiliki, tentu akan lebih mampu menyelesaikan permasalahan. Tentunya, dengan adanya pengetahuan saja itu belum cukup karena manusia diberi akal untuk melakukan penalaran serta mengambil keputusan / kesimpulan berdasarkan pengetahuan dan pengalaman yang dimiliki.

Demikian juga agar mesin bisa cerdas, maka harus diberi bekal pengetahuan, sehingga mempunyai kemampuan untuk menalar. Untuk membuat aplikasi kecerdasan buatan, terdapat 2 bagian utama yang penting, yaitu :

a. Basis pengetahuan (*Knowledge Base*), bersifat fakta – fakta, teori, pemikiran dan hubungan antar satu dengan yang lainnya.
b. Motor inferensi (*Inference Engine*), kemampuan menarik kesimpulan berdasarkan pengetahuan dan pengalaman.

Penerapan konsep kecerdasan buatan pada komputer adalah sebagai berikut :



Gambar 2.1 Konsep kecerdasan buatan

2.2 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 [8].

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. Kemampuan menjelaskan, merupakan salah satu fitur yang harus dimiliki oleh sistem pakar setelah tersedia program di dalam komputer.

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 [9].

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 Sensitif rasa dingin dan panas
AND waktu is 1 bulan
AND merokok is tidak
THEN Hiperami Pulpa

Menurut Sutojo, et al (2011), sistem pakar memiliki manfaat dan kemampuan sebagai berikut :

1. Meningkatkan produktivitas, karena Sistem Pakar dapat bekerja lebih cepat daripata Manusia.
2. Membuat seorang yang awam bekerja seperti layaknya seorang Pakar.
3. Meningkatkan kualitas, dengan member nasehat / pengarahan yang konsisten dan mengurangi kesalahan.

4. Mampu menangkap pengetahuan dan kepakaran seseorang.
5. Dapat beroperasi di lingkungan yang berbahaya.
6. Memudahkan akses pengetahuan seorang pakar.
7. Handal. Sistem pakar tidak pernah menjadi bosan dan kelelahan atau sakit.
8. Meningkatkan kapabilitas sistem komputer. 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 merespon dengan: "tidak tahu" atau "tidak yakin" pada satu atau lebih pertanyaan selama konsultasi dan Sistem Pakar tetap akan memberikan jawabannya.
10. Sistem Pakar dapat digunakan sebagai Media Pelengkap Pelatihan. Pengguna pemula 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.

2.3 Metode ANFIS (Adaptive Neuro Fuzzy Inference System)

ANFIS (*Adaptive Neuro Fuzzy Inference System* atau *Adaptive-based Fuzzy Inference System*) 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 [10].

Agar jaringan dengan fungsi basis radial ekuivalen dengan *fuzzy* berbasis aturan model. Sugeno orde 1 ini, diperlukan batasan sebagai berikut [10]:

1. Aturan – aturan harus memiliki metode agregasi yang sama untuk menghasilkan semua outputnya.

2. Jumlah fungsi aktivasi harus sama dengan jumlah aturan *fuzzy* (***IF – THEN***).
3. Jika ada beberapa input pada basis aturannya, maka tiap – tiap fungsi aktivasi harus sama dengan fungsi keanggotaan tiap – tiap inputnya.
4. Fungsi aktivasi dan aturan – aturan *fuzzy* harus memiliki fungsi yang sama untuk neuron – neuron dan aturan – aturan yang ada di sisi outputnya.

ANFIS pertama kali diperkenalkan oleh Lotfi A. Zadeh pada tahun 1965, dengan melihat kenyataan bahwa manusia dapat membuat keputusan lebih baik berdasarkan informasi yang bukan numerik dan kurang pasti. Dalam perkembangan berikutnya, diperkenalkan konsep variabel linguistik. Variabel linguistik adalah suatu variabel yang nilainya merupakan kata atau kalimat dan bukan bilangan. Pada implementasi berikutnya, variabel linguistik ini dikombinasikan dengan aturan ***IF – THEN***, sehingga konsep ini merupakan awal dari teori *fuzzy* [11].

Adaptive Neuro Fuzzy Inference System (ANFIS) merupakan suatu teknik optimasi yang menggabungkan konsep *neural network* dengan *fuzzy logic*. Neural network mengenal pola – pola dan menyesuaikan pola terhadap perubahan lingkungan. Sedangkan *fuzzy logic* menggabungkan pengetahuan manusia dan mencari kesimpulan untuk membuat suatu keputusan [12]. Untuk sistem yang berbasis aturan linguistik, teknik jaringan syaraf tiruan akan memberikan kemampuan pembelajaran dan adaptasi untuk mengekstrasi parameter – parameter (premis dan konsekuen) aturan *fuzzy* dari sekumpulan data numerik. Secara khusus, jaringan *neuro-fuzzy* menghilangkan kekurangan dalam desain sistem *fuzzy* konfensional di mana perancang harus menalar dengan *trial-error* fungsi keanggotaan dari himpunan *fuzzy* yang didefinisikan pada masukan dan keluaran dari semesta pembicaraan. ANFIS adalah sistem inferensi dari *fuzzy* yang diimplementasikan dalam jaringan adaptif. Pada ANFIS, parameter adalah fungsi keanggotaan premis dan konsekuensi. Pembelajaran ANFIS adalah pengubahan parameter fungsi keanggotaan masukan dan keluaran [13].

Fuzzy Inference System merupakan proses yang perhitungannya berdasarkan himpunan *fuzzy*, aturan "jika – maka" dan operator logika *fuzzy*. *Fuzzy*

Inference System memetakan input yang diketahui ke output dengan menggunakan logika *fuzzy*. Ada beberapa macam model dari *Fuzzy Inference System*, yaitu :

1. Model *fuzzy* Tsukamoto
2. Model *fuzzy* Sugeno
3. Model *fuzzy* Mamdani

2.4 Arsitektur ANFIS (Adaptive Neuro Fuzzy Inference System)

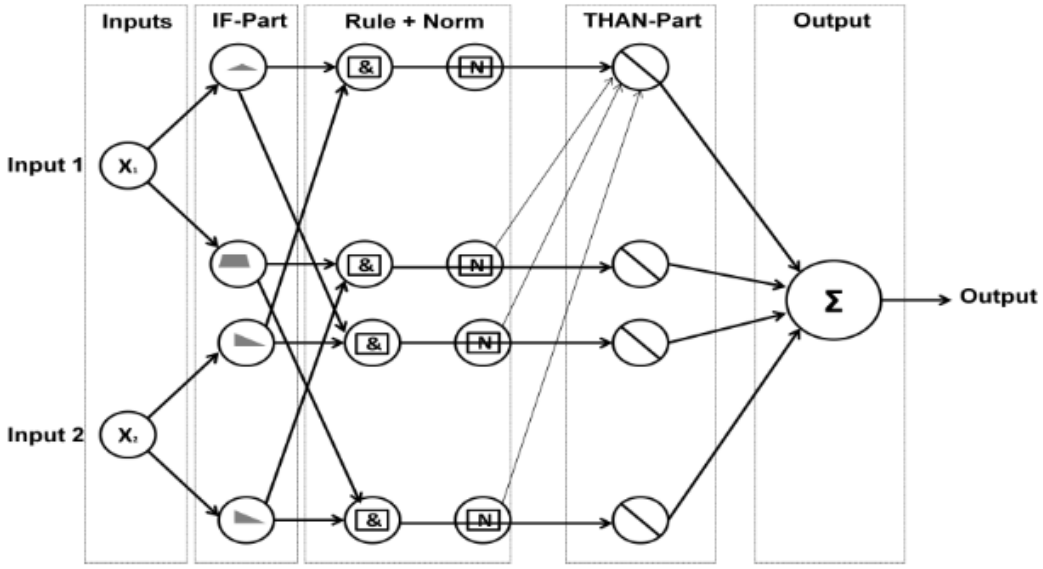
Dimisalkan ada 2 input x1, x2 dan satu output y. Ada 2 aturan pada basis aturan model Sugeno seperti pada rumus (2.1) dan (2.2) [6] :

if x_1 is A_1 and x_2 is B_1 Then $y_1 = c_{11}x_1 + c_{12}x_2 + c_{10}$ **(2.1)**

if x_1 is A_2 and x_2 is B_2 Then $y_2 = c_{21}x_1 + c_{22}x_2 + c_{20}$ **(2.2)**

Jika α predikat untuk aturan ke dua aturan adalah w_1 dan w_2, maka dapat dihitung rata – rata terbobot pada rumus (2.3) berikut :

$$y = \frac{w1y1 + w2y2}{w1 + w2} = w1y1 + w2y2 \quad \textbf{(2.3)}$$



Gambar 2.2 Arsitektur ANFIS Model Sugeno [10]

Jaringan ANFIS terdiri dari lapisan – lapisan sebagai berikut [10]:

1. Tiap – tiap neuron i pada lapisan pertama adaptif terhadap parameter suatu fungsi aktivasi. Output dari tiap neuron berupa derajat keanggotaan yang diberikan oleh fungsi keanggotaan input, yaitu : $\alpha_{A1}(x_1)$, $\alpha_{B1}(x_2)$, $\alpha_{A2}(x_1)$ atau $\alpha_{B2}(x_2)$. fungsi keanggotaan yang dipakai adalah jenis *generalized bell* (*gbell*) yang dirumuskan seperti pada rumus (2.4) sebagai berikut :

$$gbell(x, a, b, c) = \frac{1}{1 + \left|\frac{x - c}{a}\right|^{2b}} \quad \textbf{(2.4)}$$

   dimana {x, a, b, c} adalah parameter – parameter, biasanya b = 1. Jika nilai parameter – parameter ini berubah, maka bentuk kurva yang terjadi pun akan ikut berubah. Parameter – parameter pada lapisan itu biasanya dikenal dengan nama *premise parameters*.
2. Tiap – tiap neuron pada lapisan ke dua berupa neuron tetap yang outputnya adalah hasil dari masukan. Biasanya digunakan operator **AND**. Tiap – tiap node merepresentasikan α predikat dari aturan ke-i. Semua simpul pada lapisan ini adalah nonadaptif (parameter tetap). Fungsi simpul ini adalah mengalikan setiap sinyal masukan yang dating. Fungsi simpul seperti pada rumus (2.5) berikut :

$$O_{2.i} = w_i = \mu_{A_i}(x).\mu_{B_i}(y), untuk\ i = 1,2 \quad \textbf{(2.5)}$$

3. Tiap – tiap neuron pada lapisan ke tiga berupa simpul nonadaptif menampilkan fungsi derajat pengaktifan ternomalisasi (normalized firing strength) yaitu rasio keluaran simpul ke-i pada lapisan sebelumnya terhadap seluruh keluaran lapisan sebelumnya didapat pada rumus (2.6) berikut :

$$\bar{w}_i = \frac{wi}{w1+\cdots+wn} \textbf{ , dengan i = 1,2.} \quad \textbf{(2.6)}$$

   Hasil ini dikenal dengan nama *normalised firing strength*.
4. Tiap – tiap neuron pada lapisan ke empat merupakan node adaptif terhadap suatu output didapat pada rumus (2.7) berikut :

$$\bar{w}_i yi = \bar{w}_i(p_i x + q_i y + r_i); \textbf{ dengan i = 1,2} \quad \textbf{(2.7)}$$

dengan $\bar{w}_i$ adalah *normalised firing strength* pada lapisan ke tiga dan {p_i, q_i, r_i} adalah parameter – parameter pada neuron tersebut. Parameter – parameter pada lapisan tersebut disebut dengan nama *consequent parameters*, didapat dari rumus (2.8) berikut :

$$\Theta = \text{inv}(A^T A)A^T.y \quad \textbf{(2.8)}$$

Dengan y adalah nilai keluaran atau target output yang telah ditentukan sebelumnya. Dalam mencari parameter konsekuen, dicari matriks A terlebih dahulu yang didapat berdasarkan hasil normalisasi dari lapisan 3, dengan persamaan 2.9 sebagai berikut :

$$A = \begin{bmatrix} G1_l(\bar{W}_i) & G2_l(\bar{W}_i) & \dots & 1hari1-4batang_l(\bar{W}_i) & \dots & 1hari>15batang_l(\bar{W}_i) & \dots & \bar{W}_i \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ G1_{10}(\bar{W}_i) & G2_{10}(\bar{W}_i) & \dots & 1hari1-4batang_{10}(\bar{W}_i) & \dots & 1hari>15batang_{10}(\bar{W}_i) & \dots & \bar{W}_i \end{bmatrix} \quad \textbf{(2.9)}$$

5. Tiap – tiap neuron pada lapisan ke lima adalah node tetap yang merupakan hasil jumlah dari semua masukan dari rumus (2.10) berikut :

$$\sum_i \bar{w}_i yi = \frac{\sum_i w_i f_i}{w_i} \quad \textbf{(2.10)}$$

f_i = sinyal kontrol ke –i

w_i = derajat pengaktifan ternomalisasi ke-i

Terdapat pula pencarian error tiap lapisan pada jaringan ANFIS sebagai berikut [8] :

1. Pada error lapisan ke lima, apabila kita memiliki jaringan adaptif , yang hanya memiliki 1 neuron pada lapisan output, maka propagasi error yang menuju lapisan kelima didapat dari rumus (2.11) berikut :

$$\varepsilon_{5,1} = -2(Y - Y^*) \quad \textbf{(2.11)}$$

2. Pada error lapisan ke empat, propagasi error yang menuju pada lapisan ke empat didapat dari rumus (2.12) berikut :

$$\varepsilon_{4,1} = \varepsilon_{5,1} \quad \textbf{(2.12)}$$

3. Pada error lapisan ke tiga, propagasi error pada lapisan ke tiga didapat dari rumus (2.13) berikut :

$$\varepsilon_{3,i} = (A_i G1 + B_i G2 + \cdots + O_i 1hari1 - 4batang + \cdots + Q_i 1hari > 15batang + \cdots + R_i)\,\varepsilon_{4,i} \quad \textbf{(2.13)}$$

4. Pada error lapisan ke dua, propagasi error yang menuju lapisan ke dua didapat dari rumus (2.14) berikut :

$$\varepsilon_{2,i} = \left(\frac{W_i}{(w_1 + \cdots + w_{10})^2}\right)\varepsilon_{3,i} \quad \textbf{(2.14)}$$

5. Pada error lapisan ke satu, propagasi error yang menuju pada lapisan ke satu dapat dirumuskan sebagai berikut :

$$\varepsilon_{1,i} = (\boldsymbol{\mu_{1minggu}(G2)} \cdot \boldsymbol{\mu_{1minggu}(G3)} \cdot \boldsymbol{\mu_{1minggu}(G4)} \cdot \boldsymbol{\mu_{1minggu}(G5)} \cdot \boldsymbol{\mu_{1minggu}(G6)} \cdot \boldsymbol{\mu_{1minggu}(G7)} \cdot \boldsymbol{\mu_{1minggu}(G8)} \cdot \boldsymbol{\mu_{1minggu}(G9)} \cdot \boldsymbol{\mu_{1minggu}(G10)} \cdot \boldsymbol{\mu_{1minggu}(G11)} \cdot \boldsymbol{\mu_{1minggu}(G12)} \cdot \boldsymbol{\mu_{1minggu}(G13)} \cdot \boldsymbol{\mu_{1hari1\text{-}4batang}(merokok)} \cdot \boldsymbol{\mu_{1hari5\text{-}15batang}(merokok)} \cdot \boldsymbol{\mu_{1hari>15batang}(merokok)})\,\varepsilon_{2,i} \quad \textbf{... (2.15)}$$

2.5 Ilmu Penyakit Gigi dan Mulut

Ilmu penyakit gigi dan mulut atau oral patologi adalah ilmu yang mempelajari penyakit-penyakit dan kelainan yang terjadi pada rongga mulut, tanda-tanda atau gejalanya, penyebabnya serta perawatannya.

Pada tabel 2.1, penyakit gigi dan mulut serta gejala umumnya.

Tabel 2.1 Penyakit Gigi dan mulut dan gejala umumnya

No	Penyakit	Gejala
1	Hiperemi Pulpa	Sensitif Rasa Dingin dan Panas, Gigi Terasa Sakit dan Berdenyut, Sakit Saat Mengunyah
2	Pulpitis	Sensitif Rasa Dingin dan Panas, Gigi Terasa Sakit dan Berdenyut, Gusi Mudah Berdarah
3	Abses	Gigi Terasa Sakit dan Berdenyut, Gusi Merah dan Bengkak, Sakit Saat mengunyah, Demam, Pembekakan Kelenjar Getah Bening
4	Gingivitis	Gusi Merah dan Bengkak, Bau Mulut, Gusi Mudah Berdarah, Gusi Terasa Sakit
5	Fissure Tongue	Gusi Merah dan Bengkak, Lidah Retak - retak, Lidah Terasa Sakit dan Bengkak
6	Stomatitis	Permukaan Lidah Berwarna Merah dan Sensasi Terbakar, Bau Mulut,Sakit Saat Menelan
7	Glositis	Gigi Terasa Sakit dan Berdenyut, Permukaan Lidah Berwarna Merah dan Sensasi Terbakar, Sakit,Saat Menelan, Lidah Terasa Sakit dan Bengkak
8	Kanker Mulut	Gusi Merah dan Bengkak, Sakit saat Mengunyah, Pembekakan Kelenjar Getah Bening, Gusi Mudah Berdarah, Sakit Saat menelan
9	Trench Mouth	Demam, Pembekakan Kelenjar Getah Bening, Bau Mulut, Gusi Mudah Berdarah, Gusi Terasa Sakit
10	Periodontitis	Sensitif rasa dingin dan panas, Gusi Merah dan Bengkak, Bau Mulut, Gusi Mudah Berdarah, Gusi Terasa Sakit

2.6 Pengertian dan Macam – macam Diagnosa

Dalam pelayanan asuhan keperawatan, diagnosis dapat diartikan sebagai analisis dari penyebab dan sifat dari suatu masalah dan atau situasi atau suatu pernyataan mengenai solusinya.

2.6.1 Pengertian Oral Diagnosis

Diagnosis dalam kedokteran gigi dapat diartikan sebagai penentuan jenis penyakit yang diderita pasien. Pengertian lainnya adalah cara-cara pemeriksaan untuk menentukan suatu diagnosa. Mengidentifikasi kelainan–kelainan yang berhubungan dengan gigi dan jaringan sekitarnya dengan jalan menanyakan, memeriksa, dan menyatukan gambaran penyakit yang terlihat dengan faktor-faktor yang diperoleh dari wawancara tersebut yang dapat membedakan dari penyakit yang lain.

2.6.2 Prognosis

Ramalan apa yang akan terjadi dari setiap penyakit dan secara nyata dipengaruhi oleh perawatan yang diberikan.

2.6.3 Gejala dan Tanda

Kesatuan informasi, yang dicari di dalam diagnosis klinis dan didefinisikan sebagai fenomena atau tanda-tanda suatu permulaan keadaan sakit.

a. Gejala/symptom : adalah gejala yang dirasakan dan dilaporkan oleh pasien kepada pemeriksa.

b. Tanda/sign : adalah kondisi yang ditemukan/ dipastikan oleh pemeriksa melalui berbagai uji / tes.

2.7 Microsoft Visual Studio

Microsoft Visual Studio merupakan sebuah perangkat lunak lengkap yang dapat digunakan untuk melakukan pengembangan aplikasi, baik itu aplikasi bisnis, aplikasi personal, ataupun komponen aplikasinya, dalam bentuk aplikasi *console*, aplikasi *Windows*, ataupun aplikasi *Web*. *Visual Studio*

mencakup kompiler, SDK, *Integrated Development Environment* (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 di atas *.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).

2.8 Bahasa Pemrograman C#

C# (dibaca: C *sharp*) merupakan sebuah bahasa pemrograman yang berorientasi objek yang dikembangkan oleh Microsoft sebagai bagian dari inisiatif kerangka .NET *Framework*. Bahasa pemrograman ini dibuat berbasiskan bahasa C++ yang telah dipengaruhi oleh aspek - aspek atau pun fitur bahasa yang terdapat pada bahasa - bahasa pemrograman lainnya seperti *Java*, *Delphi*, *Visual Basic* dan lain - lain dengan beberapa penyederhanaan. Menurut standar ECMA-334 C# *Language Specification*, nama C# terdiri atas sebuah huruf latin C (U+0043) yang diikuti oleh tanda pagar yang menandakan angka # (U+0023). Tanda pagar # yang digunakan memang bukan tanda kres

dalam seni musik (U+266F), dan tanda pagar # (U+0023) tersebut digunakan karena karakter kres dalam seni musik tidak terdapat didalam *keyboard* standar.

2.9 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 [12].

Flowchart adalah bentuk gambar/diagram yang mempunyai aliran satu atau dua arah secara *sekuensial*. *Flowchart* digunakan untuk merepresentasikan maupun mendesain program. Oleh karena itu *flowchart* harus bisa merepresentasikan komponen-komponen dalam bahasa pemrograman [12].

2.10 Entity Relationship Diagram

ERD adalah model konseptual yang mendeskripsikan hubungan antara penyimpanan. ERD digunakan untuk memodelkan struktur data dan hubungan antar data. Dengan ERD, model dapat diuji dengan mengabaikan proses yang dilakukan [12].

ERD pertama kali dideskripsikan oleh Peter Chen yang dibuat sebagai bagian dari perangkat lunak *CASE*. Komponen – komponen yang termasuk dalam ERD antara lain, adalah**:** (Imbar, 2006)

a. Entitas (*Entity*)

   Sebuah barang atau obyek yang dapat dibedakan dari obyek lain.

b. Relasi (*Relationship*)

   Asosiasi 2 atau lebih entitas dan berupa kata kerja.

c. Atribut (*Attribute*)

   Properti yang dimiliki setiap entitas yang akan disimpan datanya.

d. Kardinalitas (*Kardinality*)

Angka yang menunjukkan banyaknya kemunculan suatu obyek terkait dengan kemunculan obyek lain pada suatu relasi. Kardinalitas relasi yang terjadi diantara dua himpunan entitas (misalnya A dan B) dapat berupa:

1. Modalitas (*Modality*) adalah Partisipasi sebuah entitas pada suatu relasi, 0 jika partisipasi bersifat "optional"/parsial, dan 1 jika partisipasi bersifat "wajib"/total.
2. *Total constraint* adalah *constraint* yang mana data dalam entitas yang memiliki *constraint* tersebut terhubung secara penuh ke dalam entitas dari relasinya.

2.11 Data Flow Diagram

Data flow diagram adalah suatu grafik yang menjelaskan sebuah sistem dengan menggunakan bentuk-bentuk dan simbol-simbol untuk menggambarkan aliran data dari proses-proses yang saling berhubungan. *Data flow diagram* ini adalah salah satu alat pembuatan model yang sering digunakan, khususnya bila fungsi-fungsi sistem merupakan bagian yang lebih penting dan kompleks dari pada data yang dimanipulasi oleh sistem. Dengan kata lain, *data flow diagram* adalah alat pembuatan model yang memberikan penekanan hanya pada fungsi sistem. *Data flow diagram* ini merupakan alat perancangan sistem yang berorientasi pada alur data dengan konsep dekomposisi dapat digunakan untuk penggambaran analisa maupun rancangan sistem yang mudah dikomunikasikan oleh profesional sistem kepada pemakai maupun pembuat program [14].

2.12 Pengujian

Pengujian adalah suatu proses mengevaluasi suatu sistem atau komponen dengan maksud untuk menemukan apakah itu memenuhi persyaratan yang ditentukan atau tidak. Dengan kata lain pengujian ialah mengeksekusi sistem untuk mengidentifikasi kesenjangan, atau kesalahan. Dalam penelitian ini dilakukan dua pengujian yaitu pengujian black box dan pengujian akurasi.

2.12.1 Pengujian Black box

Pengujian yang mengabaikan mekanisme internal sistem atau komponen dan fokus semata-mata pada output yang dihasilkan yang merespon input yang dipilih dan kondisi eksekusi. Pengujian yang dilakukan untuk mengevaluasi pemenuhan sistem atau komponen dengan kebutuhan fungsional tertentu [15].
Tujuan *Black Box* diantaranya adalah menemukan fungsi yang tidak benar atau hilang, Kesalahan *interface*, *Error* pada struktur data atau akses database external, *Error* pada kinerja, *Error* pada saat inisialisasi dan terminasi, Kesensitifan sistem terhadap nilai input tertentu, Batasan dari suatu data.

2.12.2 Pengujian Confusion Matrix

Confusion matriks adalah sebuah matriks yang memberikan gambaran mengenai tingkat kesalahan atau *eror rate* dan kualitas prediksi sebuah model. Ukuran dimensi *confusion matriks* adalah L x L yang dimana L merupakan jumlah nilai yang ingin dibandingkan [14]. Pengujian ini digunakan untuk mengetahui hasil diagnosa awal pada penyakit gigi dan mulut yang sudah ditentukan (asli) dengan hasil diagnosa awal pada penyakit gigi dan mulut yang dilakukan oleh ANFIS (sistem).

Tabel 2.2 *Confusion Matrix*

		Hasil Diagnosa Awal (Sistem)	
		1	**2**
Hasil Diagnosa Awal (Asli)	**1**	a	b
	2	c	d

Setiap kolom dari *confusion matrix* mewakili contoh di kelas yang dilakukan oleh sistem, sedangkan setiap baris mewakili contoh di kelas yang sudah ditentukan hasilnya (asli). Setelah diketahui nilai a, b, c dan d maka selanjutnya akan dilakukan penghitungan nilai akurasinya. Rumus untuk mencari akurasinya didapat pada rumus (2.15) berikut :

$$\textbf{Akurasi} = \frac{a+d}{a+b+c+d} \qquad \textbf{2.16}$$