

BAB II

TINJAUAN PUSTAKA

A. Penelitian Terdahulu

Chakrabarti, (2008) telah menerapkan sistem pakar untuk mendiagnosa penyakit pada tanaman mangga di Negara India. Metode yang digunakan dalam sistem pakar ini yakni metode logika *fuzzy*, hasil yang diperoleh dari pengembangan sistem yang dirancang sangat memuaskan dan dapat membantu para petani. Pengembangan aplikasi sistem pakar untuk mendiagnosa penyakit pada tanaman bawang merah. Sistem pakar ini menggunakan metode *Forward Chaining*. Sistem pakar yang dirancang dapat memberikan hasil diagnose yang baik dan sistem ini dapat membantu para petani .

Sasmito, Surarso, & Sugiharo (2011). *Mobile phone* sebagai sebuah perangkat cerdas yang dipadukan dengan sistem pakar untuk mendiagnosa penyakit pada manusia. Pasien memasukan gejala melalui sms kepada *server* dan *server* akan memprosesnya dengan metode *fuzzy* dan mengirimkan hasil diagnosa serta cara pencegahannya kepada pasien melalui sms. Penggunaan perangkat *mobile phone* menjadi lebih kompleks disamping sebagai media komunikasi, *mobile* juga bisa digunakan sebagai perangkat cerdas.

Anton Setiawan Honggowibowo (2009), mengembangkan Sistem Pakar untuk mendiagnosa tanaman padi dengan menggunakan metode *forward chaining* dan *backward chaining* berbasis *web* dapat memberikan hasil diagnosa melalui *web* dan pencegahannya ketika pengguna memasukan

gejala. Sistem ini sangat mudah dan dapat membantu para petani untuk mengakses pengetahuan tentang penyakit pada tanaman padi melalui internet berdasarkan gejala yang dilihatnya yang tidak terbatas pada waktu.

B. Kecerdasan Buatan

Kecerdasan buatan (*Artificial Intelligence*) adalah ide-ide untuk membuat suatu perangkat lunak komputer yang memiliki kecerdasan sehingga perangkat lunak komputer tersebut dapat melakukan suatu pekerjaan yang dilakukan oleh manusia (Artanti, 2004), dengan kata lain membuat sebuah komputer dapat berpikir dan bernalar seperti manusia. Tujuan dari kecerdasan buatan adalah membuat komputer lebih cerdas, mengerti tentang kecerdasan dan membuat mesin lebih berguna bagi manusia. Kecerdasan buatan dapat membantu meringankan beban kerja manusia misalnya dalam membuat keputusan, mencari informasi secara lebih akurat atau membuat komputer lebih mudah digunakan dengan tampilan yang lebih mudah dipahami. Cara kerja kecerdasan buatan adalah menerima input, untuk kemudian diproses dan kemudian mengeluarkan output yang berupa keputusan.

Menurut Arhami (2004) sistem pakar adalah proses untuk mengetahui dan memodelkan proses-proses berpikir manusia dan mendesain mesin agar dapat menirukan perilaku manusia. Kecerdasan atau kepandaian ini di dapat berdasarkan pengetahuan dan pengalaman, untuk itu agar perangkat lunak yang dikembangkan dapat mempunyai kecerdasan maka perangkat lunak tersebut harus diberi suatu pengetahuan dan kemampuan untuk menalar dari

pengetahuan yang telah didapat dalam menemukan solusi atau kesimpulan layaknya seorang pakar dalam bidang tertentu yang spesifik.

Kecerdasan buatan menawarkan media dan uji teori kecedasan. Teori ini dapat dinyatakan dalam bahasa program komputer dan dibutikan melalui eksekusinya pada komputer nyata. Implementasi dari kecerdasan buatan saat ini dapat ditemui dalam-daam bidang antara lain:

1. *Fuzzy logic* merupakan suatu metode kecerdasan buatan yan banyak terdapat pada alat elektronik dan robot dimana ala-alat elektronik dan robot terebut mampu berpikir dan bertindak laku seperti manusia.
2. *Komputer vision* merupakan suatu metode kecerdasan buatan yang memungkinkan sebuah sistem komputer mengenali gambar sebagai inputnya. Contohnya adalah mengenali dan membaca tulisan yang ada gambarnya.
3. *Artificial intelligence* dalam game merupakan metode kecerdasan buatan yang berguna untuk meniru cara berpikir manusia dalam bermain game.
4. *Speech recognition* merupakan suatu metode kecerdasan buatan yang berguna untuk mengenali suara manusia dengan cara dicocokkan dengan acuan aau pattern yang telah diprogram sebelumnya. Contohnya adalah suara dari user dapat diterjemahkan menjadi sebuah perintah bagi komputer.
5. *Expert system* merupakan metode kecerdasan buatan yang berguna untuk meniru cara berpikir dan penalaran seorang ahli dalam mengambil keputusan berdasarkan situasi yang ada.

C. Sistem Pakar

1. Pakar dan sistem pakar

Sistem pakar adalah sistem informasi berbasis komputer yang menggunakan pengetahuan pakar untuk mencapai performa keputusan tingkat tinggi dalam domain persoalan yang sempit.

Sedangkan Pakar adalah orang yang memiliki pengetahuan, penilaian, pengalaman dan metode khusus, serta kemampuan untuk menerapkan bakat ini dalam memberi nasihat dan memecahkan masalah. Adalah tugas pakar untuk menyediakan pengetahuan tentang bagaimana melaksanakan suatu tugas yang akan dijalankan oleh sistem berbasis-pengetahuan. Pakar mengetahui fakta mana yang penting dan memahami arti hubungan diantaranya. Misalnya dalam mendiagnosis persoalan sistem listrik mobil, pakar mekanik mengetahui bahwa pengikat kipas dapat diputus dan menyebabkan baterai *discharge*.

Keahlian adalah pengetahuan ekstensif yang spesifik terhadap tugas yang dimiliki pakar. Tingkat keahlian menentukan performa keputusan. Keahlian sering dicapai dari pelatihan, membaca, dan mempraktikkan. Keahlian mencakup pengetahuan eksplisit, misalnya teori yang dipelajari dari buku teks atau kelas, dan pengetahuan implisit yang diperoleh dari pengalaman. (Sumber: Efraim Turban, Jay E. Aronson, Ting Peng Liang, 2005, 714-715).

2. Perbedaan Pakar dan Sistem Pakar

Menurut (Efraim Turban, Jay E. Aronson, Ting Peng Liang,2005,718) Perbedaan pakar dan sistem pakar dapat dilihat dari berbagai faktor, antara lain:

Tabel 2.1 Perbedaan Pakar Manusia dan Sistem Pakar

Fitur	Pakar manusia	Sistem pakar
Mortalitas	Ya	Tidak
Transfer pengetahuan	Sulit	Mudah
Dokumentasi pengetahuan	Sulit	Mudah
Konsistensi keputusan	Rendah	Tinggi
Unit biaya penggunaan	Tinggi	Rendah
Kreativitas	Tinggi	Rendah
Adaptabilitas	Tinggi	Rendah
Lingkup pengetahuan	Luas	Sempit
Tipe pengetahuan	Umum dan teknis	Teknis
Isi pengetahuan	Pengalaman	Simbol

Sumber: Turban (2005)

3. Manfaat dan kemampuan sistem pakar.

Menurut (Efraim Turban, Jay E. Aronson, Ting Peng Liang,2005,730-732) ada beberapa manfaat serta kemampuan sistem pakar,antara lain :

- a. Meningkatkan *output* dan produktivitas.
- b. Menurunkan waktu pengambila keputusan.
- c. Meningkatkan kualitas proses dan produk.

- d. Mengurangi *downtime*.
- e. Menyerap keahlian langka.
- f. Fleksibilitas.
- g. Operasi peralatan yang lebih mudah.
- h. Eliminasi kebutuhan peralatan yang mahal.
- i. Operasi di lingkungan berbahaya.
- j. Aksesibilitas ke pengetahuan *help desk*.
- k. Kemampuan untuk bekerja dengan informasi yang tidak lengkap atau tidak pasti.
- l. Kelengkapan Pelatihan.
- m. Peningkatan pemecahan masalah dan pengambilan keputusan.
- n. Meningkatkan proses pengambilan keputusan.
- o. Meningkatkan kualitas keputusan.
- p. Kemampuan untuk memecahkan persoalan kompleks.
- q. Transfer pengetahuan ke lokasi terpencil.
- r. Peningkatan sistem informasi lain.

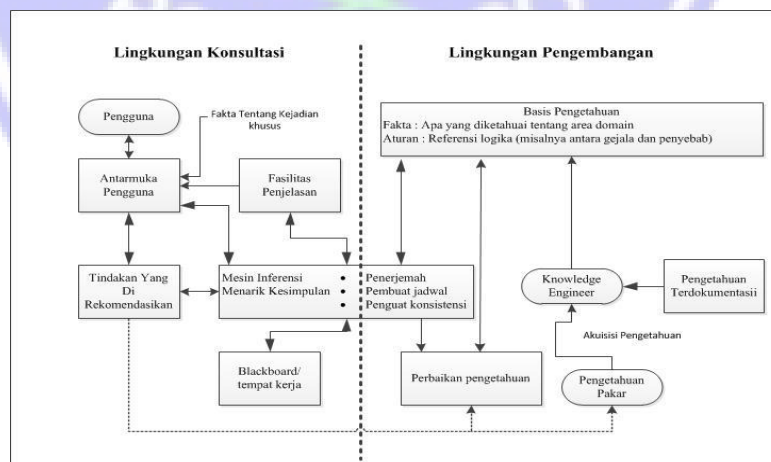
4. Keterbatasan sistem pakar.

Metodologi *ES* yang tersedia mungkin tidak langsung dan efektif, bahkan untuk banyak aplikasi dalam kategori umum. Persoalan-persoalan berikut telah memperlambat penyebaran komersial *ES* :

- a. Pengetahuan tidak selalu siap sedia.
- b. Akan sulit mengekstrak keahlian dari manusia.
- c. Pendekatan tiap pakar pada suatu penilaian situasi mungkin berbeda tetapi benar.

- d. Sulit, bahkan bagi pakar kemampuan tinggi, untuk mengikhtisarkan penilaian situasi yang baik pada saat berada dalam tekanan waktu.
- e. Pengguna siste pakar memiliki batasan kognitif alami.
- f. ES bekerja dengan baik hanya dalam domain pengetahuan sempit.
- g. Kebanyakan pakar tidak mempunyai sarana mandiri untuk memeriksa apakah kesimpulannya masuk akal.
- h. Kosakata atau jargon yang digunakan pakar unuk menyatakan fakta dan hubungan seirigkali terbatas dan tidak dipahami pakar lain.
- i. Acapkali dibutuhkan *knowledge engineer*-yang langka dan mahal-suatu fakta yang menjadikan konstruksi *ES* mahal.
- j. Kurangnya kepercayaan pada bagian pengguna akhir menjadi penghalang penggunaan *ES*.
- k. Transfer pengetahuan adalah subjek terhadap sekumpulan bias perseptual dan penilaian.(Sumber:Efraim Turban, Jay E. Aronson, Ting Peng Liang,2005,733).

5. Struktur sistem pakar.



Gambar 2.1 Struktur Sistem Pakar

(Sumber:Efraim Turban, Jay E. Aronson, Ting Peng Liang,2005,722).

Sistem pakar dapat ditampilkan dengan dua lingkungan yaitu: lingkungan pengembangan dan lingkungan konsultasi. Lingkungan pengembangan digunakan oleh ES builder untuk membangun komponen dan memasukkan pengetahuan kedalam basis pengetahuan. Lingkungan konsultasi digunakan nonpakar. Lingkungan konsultasi digunakan nonpakar untuk memperoleh pengetahuan dan nasihat pakar. Lingkungan ini dapat dipisahkan setelah sistem lengkap.

Komponen-komponen yang ada pada sistem pakar :

1) Subsistem akuisisi pengetahuan

Akuisisi pengetahuan adalah akumulasi, transfer, dan transformasi keahlian pemecahan masalah dari pakar atau sumber pengetahuan terdokumentasi ke program komputer, untuk membangun atau memperluas basis pengetahuan .

2) Basis pengetahuan

Basis pengetahuan berisi pengetahuan relevan yang diperlukan untuk memahami, merumuskan, dan memecahkan persoalan. Basis tersebut mencakup dua elemen dasar yaitu :

- a) fakta, misalnya situasi persoalan dan teori area persoalan, dan
- b) heuristik atau aturan khusus yang mengarahkan penggunaan pengetahuan untuk memecahkan persoalan khusus

3) Mesin inferensi

“Otak” ES adalah mesin inferensi, yang dikenal juga sebagai struktur kontrol atau penterjemah aturan (dalam ES berbasis aturan).

Komponen ini sebenarnya adalah program komputer yang menyediakan

metodologi untuk mempertimbangkan informasi dalam basis pengetahuan dan *blackboard*, dan merumuskan kesimpulan.

4) Antarmuka pengguna

Sistem pakar berisi prosesor bahasa untuk komunikasi berorientasi-persoalan yang mudah antara pengguna dan komputer. Komunikasi ini paling baik dilakukan dalam bahasa alami. Dikarenakan batasan teknologi, maka kebanyakan sistem yang ada menggunakan pendekatan pertanyaan dan jawaban untuk berinteraksi dengan pengguna. Acapkali ditambahi dengan menu, formulir elektronik, dan grafik.

5) Blackboard (tempat kerja)

Blackboard adalah area kerja memori yang tersimpan sebagai database untuk deskripsi persoalan terbaru yang ditetapkan oleh data input, digunakan juga untuk perekaman hipotesis dan keputusan sementara. Tiga tipe keputusan dapat direkam dalam *blackboard*: rencana (bagaimana mengatasi persoalan), agenda (tindakan potensial sebelum eksekusi), dan solusi (hipotesis kandidat dan arah tindakan alternatif yang telah dihasilkan sistem sampai dengan saat ini).

6) Subsistem penjelasan (*justifier*)

Kemampuan untuk melacak tanggung jawab suatu kesimpulan terhadap sumbernya adalah penting untuk transfer keahlian dan dalam pemecahan masalah. Subsistem penjelasan (disebut juga *justifier*) dapat melacak tanggung jawab tersebut dan menjelaskan perilaku ES dengan menjawab pertanyaan berikut secara interaktif :

- a) Mengapa suatu pertanyaan ditanyakan oleh sistem pakar ?
- b) Bagaimana suatu kesimpulan dicapai ?
- c) Mengapa suatu alternatif ditolak ?
- d) Apa rencana untuk mencapai solusi ? misalnya,apa yang tetap tersisa sebelum diagnosis akhir ditetapkan.

Dalam sistem pakar sederhana, penjelasan menunjukkan aturan yang digunakan untuk memperoleh rekomendasi tertentu.

7) Sistem perbaikan pengetahuan

Pakar manusia memiliki sistem perbaikan-pengetahuan,yakni mereka dapat menganalisis pengetahuannya sendiri dan kegunaannya, belajar darinya,dan meningkatkannya untuk konsultasi mendatang.Serupa pula,evaluasi tersebut diperlukan dalam pembelajaran komputer sehingga program dapat menganalisis alasan keberhasilan atau kegagalannya.hal ini dapat mengarah kepada peningkatan sehingga menghasilkan basis pengetahuan yang lebih akurat dan pertimbangan yang lebih efektif.Komponen tersebut tidak tersedia dalam sistem pakar komersial saat ini,tetapi sedang dikembangkan dalam ES eksperimental pada beberapa universitas dan lembaga riset.(Sumber:Efraim Turban, Jay E. Aronson, Ting Peng Liang,2005,722-724).

6. Cara kerja sistem pakar : mekanisme inferensi

Diantara komponen-komponen dalam gambar 2.1, basis pengetahuan dan mesin inferensi adalah modul paling kritis agar sistem pakar dapat berfungsi dengan baik. Pengetahuan harus direpresentasikan dan diatur secara tepat dalam basis pengetahuan. Mesin inferensi

kemudian dapat untuk menarik kesimpulan baru dari fakta dan aturan yang ada. Dalam bagian ini, kami memperkenalkan struktur pengetahuan dan mesin inferensi pada sistem berbasis aturan.(Sumber:Efraim Turban, Jay E. Aronson, Ting Peng Liang,2005,724).

a. Representasi dan organisasi pengetahuan

Pengetahuan pakar harus direpresentasikan dalam format yang dapat dipahami komputer dan diatur dengan tepat dalam basis pengetahuan sistem pakar.Terdapat banyak cara yang berbeda untuk merepresentasikan pengetahuan manusia, antara lain aturan produksi, jaringan semantik, dan pernyataan logika.

b. Mesin inferensi

Dalam keputusan kompleks, pengetahuan pakar sering tidak dapat direpresentasikan dalam aturan tunggal. Sebaliknya, aturan dapat digabungkan secara dinamis untuk mencakup berbagai kondisi. Proses penggabungan banyak aturan berdasarkan data yang tersedia, disebut inferensi .Komponen yang melakukan inferensi dalam sistem pakar disebut mesin inferensi. Dua pendekatan populer untuk menarik kesimpulan adalah *forward chaining* dan *backward chaining*.

1) *Forward chaining*

Forward chaining merupakan metode inferensi yang melakukan penalaran dari suatu masalah kepada solusinya. Jika klausa premis sesuai dengan situasi (bernilai *TRUE*), maka proses akan menyatakan konklusi. *Forward chaining* adalah *data-driven* karena

inferensi dimulai dengan informasi yang tersedia dan baru konklusi diperoleh.

Contoh :

Terdapat 10 aturan dalam basis pengetahuan yaitu :

R1 : if A and B then C

R2 : if C then D

R3 : if A and E then F

R4 : if A then G

R5 : if F and G then D

R6 : if G and E then H

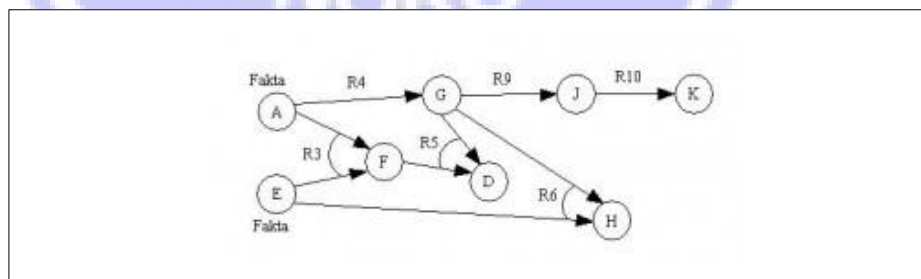
R7 : if C and H then I

R8 : if I and A then J

R9 : if G then J

R10 : if J then K

Fakta awal yang diberikan hanya A dan E, ingin membuktikan apakah K bernilai benar. Proses penalaran forward chaining terlihat pada gambar dibawah :



Sumber: Turban (2005)

Gambar 2.2 *Forward Chaining*.

Sumber:<http://diskusikuliaah.files.wordpress.com/2010/10/fc.jpg>

2) Backward chaining

Backward chaining merupakan metode inferensi yang dimulai dari harapan apa yang akan terjadi (hipotesis) dan kemudian mencari bukti yang mendukung atau yang berlawanan dengan tujuan yang dimaksud, dalam hal ini memerlukan perumusan dan pengujian hipotesis terlebih dahulu.

7. Kategori umum sistem pakar

Berikut adalah tabel yang menunjukkan kategori umum bidang-bidang masalah yang cocok untuk sistem pakar :

Tabel 2.2 Kategori Umum Sistem Pakar.²⁹⁾

Kategori	Persoalan yang ditangani
Sistem interpretasi	Menyimpulkan deskripsi situasi dari observasi
Sistem prediksi	Menyimpulkan kemungkinan konsekuensi dari suatu situasi
Sistem diagnostik	Menyimpulkan kegagalan sistem dari observasi
Sistem desain	Mengonfigurasi objek dengan batasan
Sistem perencanaan	Mengembangkan rencana untuk mencapai tujuan
Sistem pengawasan	Membandingkan observasi rencana, memunculkan pengecualian
Sistem debugging	Menyarankan pemulihan untuk kegagalan
Sistem perbaikan	Mengeksekusi rencana untuk mengelola pemulihan yang disarankan

Sistem instruksi	Mendiagnosis,men-debug,dan memperbaiki performa siswa
Sistem kontrol	Menginterpretasikan,memprediksi,memperbaiki ,dan mengawasi kelakuan sistem

Sumber: Turban (2005)

D. Kakao

Kakao (*Theobroma cacao*) adalah tanaman bawah hutan yang berasal dari hutan hujan tropika Amerika Selatan. Sebagai tanggapan terhadap perubahan musim. Perkebunan kakao di Indonesia mengalami perkembangan pesat dalam kurun waktu 20 tahun terakhir dan pada tahun 2007 areal perkebunan kakao di Indonesia tercatat seluas 992.448 ha. Perkebunan kakao tersebut sebagian besar (89,45%) dikelola oleh rakyat dan selebihnya (5,04%) perkebunan besar Negara serta (5,51%) perkebunan besar swasta. Dari segi kualitas, kakao Indonesia tidak kalah dengan kakao dunia di mana bila dilakukan fermentasi dengan baik dapat mencapai cita rasa setara dengan kakao berasal dari Ghana dan keunggulan kakao Indonesia tidak mudah meleleh sehingga cocok bila dipakai untuk blending (Darwis, 2007).

Permintaan akan produk-produk organik merupakan peluang dunia usaha baru, baik untuk tujuan ekspor maupun kebutuhan domestik. Beberapa Negara berkembangpun mulai memanfaatkan peluang pasar ekspor produk organik ini terhadap negara maju, diantaranya buah-buahan daerah tropik untuk industry makanan bayi ke Eropa, herbas Zimbabwe ke Afrika Selatan, kapas Afrika ke Uni Eropa dan teh Cina ke Belanda serta kentang ke Jepang. Umumnya, ekspor produk organik biasanya dijual 20% lebih tinggi dari

produk pertanian nonorganik. Keuntungan pokok pertanian organik sangat bervariasi, dalam beberapa kajian ekonomi menyatakan bahwa pertanian organik memiliki akses nyata terhadap prospek jangka panjang. Beberapa studi menunjukkan bahwa pertanian organik berpengaruh sangat nyata terhadap jumlah tenaga kerja dibandingkan dengan pertanian non-organik. Terutama pada sistem pertanian organik melalui diversifikasi tanaman, perbedaan pola tanam dan jadwal tanam dapat mendistribusikan kebutuhan tenaga kerja berdasarkan waktunya (Tino, 2006).

Adapun penyakit kakao yang di teliti antara lain sebagai berikut :

1. Penyakit Busuk Buah :

Penyakit busuk buah adalah karena adanya jamur *Phytophthora palmivora*. Penyakit ini ditandai dengan adanya bercak coklat kehitaman dengan batas yang tegas, serangan biasanya dimulai dari ujung atau pangkal buah, perkembangan bercak coklat cukup cepat sehingga dalam waktu beberapa hari seluruh permukaan buah menjadi busuk, basah, dan berwarna coklat kehitaman. Pada kondisi lembab di permukaan buah akan muncul serbuk berwarna putih. Serbuk ini adalah spora *Phytophthora palmivora* yang sering kali bercampur dengan jamur sekunder (jamur lain).

2. Penyakit Kanker Batang :

Penyakit kanker batang adalah cendawan *Phytophthora palmivora* dan sering terjadi pada saat peralihan musim kemarau ke musim hujan. Gejala awal berupa bercak kecil pada cabang, kemudian melebar dan basah

karena mengeluarkan blendok/lendir. Kemudian cendawan masuk ke batang serta berubah menjadi coklat tua dan pohon akan mati.

3. Penyakit Antraknose Colletotrichum :

Penyakit antraknose colletotrichum adalah penyakit yang menyerang batang, polong dan tangkai daun, akibat serangan adalah perkecambahan biji terganggu, kadang-kadang bagian-bagian yang terserang tidak menunjukkan gejala. Gejala hanya timbul bila kondisi menguntungkan perkembangan jamur, tulang daun pada permukaan bawah tanaman terserang biasanya menebal dengan warna kecoklatan, pada batang akan timbul bintik-bintik hitam berupa duri-duri jamur yang menjadi ciri khasnya adalah jamur penyebab penyakit antraknosa.

4. Penyakit Vascular Streak Dieback :

Penyakit Vascular Streak Dieback adalah menginfeksi pucuk dan cabang kakao, tetapi gejala hanya terlihat pada daun yang tampak klorotik dan dapat berkembang pada gejala khas berupa belang hijau dengan latar belakang kuning. Pada tanaman yang sudah tua, gejala pada daun sering ditemukan pada bagian tengah cabang, sedangkan pada tanaman muda gejala dapat terjadi pada daun mana saja. Selain gejala tersebut di atas, terjadi pula perubahan warna jaringan vaskuler pada scars daun segar yang jatuh, pembengkakan lentisel pada kulit dalam daerah daun yang jatuh, serta sprouting tunas aksilar. Nekrosis antara tulang daun terminal tampak menyerupai gejala kekurangan kalsium. Selain itu garis-garis coklat terlihat pada cabang yang terinfeksi, bila cabang ini dibelah secara longitudinal.

5. Penyakit Jamur Upas :

Penyakit Jamur Upas adalah Jenis jamur ini merupakan penyakit utama yang menyerang tanaman kakao dan dialami hampir oleh semua petani kakao. Penyebab utama penyakit ini adalah kebersihan kebun yang kurang serta minimnya pemangkasan. Inspeksi pertama kali terjadi pada sisi bagian bawah cabang dan ranting, jamur mula-mula membentuk miselium tipis mengkilat seperti sutera atau perak, sangat mirip dengan sarang laba-laba. Pada fase ini jamur belum masuk ke dalam jaringan kulit, jamur kemudian membentuk kerak yang berwarna merah jambu seperti warna ikan salem, kerak tersebut terdiri atas lapisan basidia, kulit cabang dibawah kerak menjadi busuk, jamur akan berkembang terus dan akan membentuk piknidia yang berwarna merah tua dan biasanya terdapat pada sisi yang lebih kering, Pada bagian ujung dari cabang yang sakit, daun-daun layu mendadak dan banyak yang tetap melekat pada cabang, meskipun sudah kering.

6. Penyakit Akar :

Penyakit Akar adalah ada tiga macam jenis penyakit akar diantaranya : penyakit akar merah, penyakit akar coklat dan penyakit akar putih. Gejala diatas tanah dari ketiga jenis penyakit akar ini adalah sama. Mula-mula daun

menguning, layu dan akhirnya gugur, kemudian diikuti oleh kematian tanaman, untuk mengetahui patogennya dengan tepat harus melalui pemeriksaan akar. Penyakit akar merah disebabkan oleh jamur *Ganoderma Pseudoforeum* (Wakef) Ov. et Stein. Penularannya dengan

kontak akar sakit dengan tanaman yang sehat, Penyakit akar coklat disebabkan oleh jamur *Phellinus Noxius* Corner. Penularannya dengan kontak akar sakit dengan tanaman yang sehat akan tetapi sangat lambat. Penyakit akat putih disebabkan oleh jamur *Fomes Lignosus* Kloffzch. Penularannya dengan perantara.

7. Penyakit Kelayuan Pentil :

Penyakit Kelayuan Pentil adalah Penyakit layu (wilt disease) pada tanaman dapat disebabkan oleh faktor biotik yaitu bakteri sehingga disebut layu bakteri (*Pseudomonas solanacearum*) atau oleh jamur/cendawan yang disebut penyakit layu *Fusarium* (*Fusarium oxysporum*). Selain karena penyakit biotik, kelayuan pada tanaman juga dapat disebabkan karena factor abiotik (kekurangan air). Pengenalan gejala kelayuan pada tanaman dan ciriciri khususnya harus diketahui para petani.

E. *Fuzzy Tsukamoto*

Konsep logika *fuzzy* pertama kali diperkenalkan oleh professor Lotfi A. Zadeh dari Universitas California, pada bulan juni 1965. Logika fuzzy merupakan generalisasi dari logika klasik yang hanya memiliki dua nilai keanggotaan, yaitu 0 dan 1 (Arhami, 2005). Dalam logika fuzzy nilai kebenaran suatu pernyataan berkisar dari sepenuhnya benar sampai dengan sepenuhnya salah. Dengan teori himpunan *fuzzy*, suatu objek dapat menjadi anggota dari banyak himpunan dengan derajat keanggotaan yang berbeda dalam masing-masing himpunan. Logika *fuzzy* dapat dilihat sebagai superset

dari logika konvensional (Boolean) yang telah diperluas untuk menangani konsep nilai parsial yang benar antara “sama sekali benar” dan “sama sekali salah”. Pada tahun 1965, Zadeh menyarankan bahwa keanggotaan himpunan adalah kunci untuk mengambil keputusan ketika harus berhadapan dengan masalah yang tidak jelas.

Sebuah himpunan klasik (*crisp* atau *hard*) adalah sekumpulan objek yang jelas, yang didefinisikan sebagai cara untuk memisahkan elemen-elemen dari semesta pembicaraan yang terbagi menjadi dua, yaitu anggota dan bukan anggota. Himpunan *crisp* dapat didefinisikan dengan fungsi karakteristik. Misalkan, U adalah semesta pembicaraan. Fungsi karakteristik $\mu_A(x)$ dari himpunan *crisp* A pada U memberi nilai dalam $[0, 1]$ dan didefinisikan sebagai $\mu_A(x)=1$ jika x adalah suatu anggota dari A (misalkan : $x \in A$) dan 0 .

F. *PHP*

PHP merupakan bahasa pemrograman yang berjaa dalam sebuah *web server*. *PHP* diciptakan oleh seorang programmer unix dan perl yang bernama Rasmus Lerdorf pada bulan Agustus-September tahun 1994. Pada awalnya Rasmus mencoba menciptakan sebuah *script* dalam *website* pribadinya dengan tujuan untuk memonitor siapasaja yang pernah mengunjungi *websitenya*.

Pada awal tahun 1995 *PHP* mulai dikenalkan Rasmus kepada programmer pemula, dengan alasan bahwa bahasa yang digunakan dalam *PHP* cukup sederhana dan mudah dipahami. Selanjutnya Rasmus menulis

ulang *PHP* dengan bahasa C untuk meningkatkan kecepatan aksesnya. Pada bulan September-Oktober 1995 kode *PHP* ditulis ulang dan digabungkan menjadi *PHP/FI*. Baru kemudian di akhir tahun 1995 dirilis bagi umum secara gratis (Arief M Rudianto. 2011).

PHP adalah bahasa *server-side scripting* yang menyatu dengan *HTML* untuk membuat halaman *web* yang dinamis. Maksud dari *server side scripting* adalah sitaks dan perintah-perintah yang digunakan sepenuhnya dijalankan di *server* tetapi disertakan pada dokumen *HTML* (Arief M Rudianto. 2011)

G. XAMPP

XAMPP merupakan salah satu aplikasi *web server cross platform* yang bisa dipakai di komputer *Windows*, *Linux*, maupun *Mac*. Di modul ini sudah tersedia modul *Apache*, *MySQL*, dan *FileZilla*. *Web server* adalah suatu *server internet* yang menggunakan *protocol HTTP* untuk melayani semua proses penransferan data. *Web server* melihat hubungan dengan *internet* dan semua menuggu perintah atau permintaan dari *web browser* akan *HTML* atau dokumen.

H. MySQL

MySQL (baca : mai-se-kyu-el) merupakan *software* yang tergolong sebagai *DBMS (Database Managemen Sistem)* yang bersifat *Open Source*. *Open Source* menyatakan bahwa *software* ini dilengkapi dengan *source code* (kode yang dipakai untuk membuat *MySQL*), selain tentu saja bentuk

executable-nya atau kode yang dapat dijalankan secara langsung dalam sistem operasi, dan bisa dengan cara *download* (mengunduh) di internet secara gratis.

MySQL awalnya dibuat oleh perusahaan konsultan bernama *TcX* yang berlokasi di Swedia. Saat ini pengembangan *MySQL* berada dibawah naungan perusahaan *MySQL AB*.

Sebagai software *DBMS*, *MySQL* memiliki sejumlah fitur seperti yang dijelaskan dibawah ini.

1. Multiplatform.

MySQL tersedia pada beberapa *platform* (Windows, Linuk, Unix, dan lain-lain).

2. Handal, cepat, dan mudah digunakan.

MySQL tergolong sebagai *database server* (server yang melayani permintaan terhadap database) yang handal, dapat menangani database yang besar dengan kecepatan tinggi, mendukung banyak sekali fungsi untuk mengakses database, dan sekaligus mudah untuk digunakan.

Berbagai *tool* pendukung juga tersedia (walaupun dibuat oleh pihak lain). Perlu diketahui, *MySQL* dapat menangani sebuah table yang berukuran dalam *terabyte* (1 *terabyte* = 1024 *gigabyte*). Namun, ukuran yang sesungguhnya sangat bergantung pada batasan sistem operasi. Sebagai contoh, pada sistem *solaris 9/10*, batasan ukuran file sebesar 16 *terabyte*.

3. Jaminan keamanan akses.

MySQL mendukung pengamanan *database* dengan berbagai kriteria pengaksesan. Sebagai gambaran, dimungkinkan untuk mengatur *user* tertentu agar bisa mengakses data yang bersifat rahasia (misalnya gaji pegawai), sedangkan *user* lain tidak boleh. *MySQL* juga mendukung konektivitas ke berbagai *software*. Sebagai contoh, dengan menggunakan *ODBC (Open Database Connectivity)*, *database* yang ditangani *MySQL* dapat diakses melalui program yang dibuat dengan *Visual Basic*. *MySQL* juga mendukung program klien melalui *JDBC (Java Database Connectivity)*. *MySQL* juga bisa diakses melalui aplikasi berbasis *Web* : misalnya dengan menggunakan *PHP*.

4. Dukungan *SQL*.

Seperti tersirat dalam namanya, *MySQL* mendukung perintah *SQL (Structured Query Language)*. Sebagai mana diketahui, *SQL* merupakan standart dalam pengaksesan *database relasional*. Pengetahuan akan *SQL* akan memudahkan siapa pun untuk menggunakan *MySQL*. (Kadir; 2008; 2-3).

I. *Flowchart (Diagram Alur)*

1. Pengertian *flowchart (Diagram Alur)*.



Karena komputer membutuhkan hal-hal yang rinci, maka bahasa pemrograman bukanlah alat baik untuk merancang sebuah *algoritma* awal. Alat yang banyak dipakai untuk membuat *algoritma* adalah diagram alur (*flowchart*).

Diagram alur dapat menunjukkan secara jelas arus pengendalian suatu *algoritma*, yakni melaksanakan suatu rangkaian kegiatan secara *logis* dan *sistematis*. Suatu diagram alur dapat memberi gambaran dua *dimensi* berupa simbol-simbol grafis. Masing-masing simbol telah ditetapkan lebih dahulu fungsi dan artinya. Simbol-simbol tersebut dipakai untuk menunjukkan berbagai kegiatan operasi dan jalur pengendalian. Arti khusus dari sebuah *flowchart* adalah simbol-simbol yang digunakan untuk menggambarkan urutan proses yang terjadi di dalam suatu program komputer secara *sistematis* dan *logis*. (Sutabri; 2004; 21).

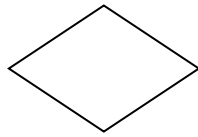
2. Simbol-simbol *flowchart*.

Sudah dikemukakan di atas bahwa diagram alur atau *flowchart* memiliki beberapa simbol yang biasa digunakan untuk menggambarkan rangkaian proses yang harus dilaksanakan. Simbol-simbol tersebut dijelaskan di bawah ini: (Sutabri; 2004; 21-22)

Tabel 2.3 Simbol *Flowchart*

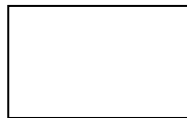
Simbol <i>Flowchart</i>	Fungsi
<i>TERMINAL</i>	
	Simbol ini digunakan untuk mengawali atau mengakhiri suatu proses/kegiatan.
<i>PREPARATION</i>	
	Simbol ini digunakan untuk mempersiapkan harga awal/nilai awal suatu variabel yang akan diproses.

DECISION



Simbol ini digunakan untuk pengujian suatu kondisi yang sedang diproses.

PROSES



Simbol ini digunakan untuk menggambarkan suatu proses yang sedang dieksekusi.

INPUT/OUTPUT



Simbol ini digunakan untuk menggambarkan proses input (*read*) maupun proses output (*print*).

SUBROUTINE



Simbol ini digunakan untuk menggambarkan proses pemanggilan subprogram dari main program.

FLOW LINE



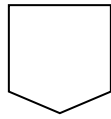
Simbol ini digunakan untuk menggambarkan arus proses dari suatu kegiatan ke kegiatan lain.

CONECTOR



Simbol ini digunakan sebagai penghubung antara suatu proses dengan proses lainnya yang ada di dalam suatu lembar halaman.

PAGE CONECTOR



Simbol ini digunakan sebagai penghubung antara suatu proses dengan proses lainnya, tetapi berpindah halaman.

MANUAL OPERATION



Simbol ini digunakan untuk menggambarkan suatu kegiatan atau proses yang bersifat manualisasi.

PRINTER



Digunakan untuk menggambarkan suatu kegiatan mencetak suatu informasi dengan mesin printer.

CONSOLE



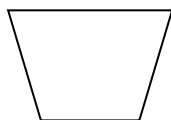
Simbol ini digunakan untuk menggambarkan suatu kegiatan menampilkan data atau informasi melalui monitor atau CRT (*Cathode Ray Tube*).

DISK



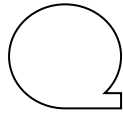
Simbol ini digunakan untuk menggambarkan suatu kegiatan membaca atau menulis data menggunakan media *magnetic disk*.

MANUAL INPUT



Digunakan untuk menggambarkan proses pemasukan data melalui media keyboard.

TAPE



Simbol ini digunakan untuk menggambarkan suatu kegiatan membaca atau menulis data menggunakan media *magnetic tape*.

Sumber : Analisis Sistem Informasi (Sutabri , 2004)

3. Jenis *flowchart*.

Bentuk diagram alur (*flowchart*) yang sering digunakan dalam proses pembuatan suatu program komputer adalah sebagai berikut:

a. Program *flowchart*.

Simbol-simbol yang menggambarkan proses secara rinci dan detail antara intruksi yang satu dengan intruksi yang lainnya dalam suatu program komputer yang bersifat *logic*.

b. Sistem *flowchart*.

Simbol-simbol yang menggambarkan urutan prosedur secara detail dalam suatu sistem komputerisasi. Bersifat fisik.

c. Teknik pembuatan *flowchart*.

Sebelum kita membuat sebuah program komputer, yang harus kita lakukan sebelumnya adalah membuat *flowchart*. Jenis *flowchart* yang sering digunakan adalah program *flowchart*.



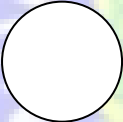
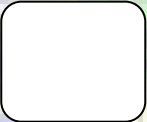


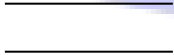

J. Data Flow Diagram (DFD)

Data Flow Diagram (DFD) adalah representasi grafik dari sebuah sistem. *DFD* menggambarkan komponen-komponen sebuah sistem, aliran-aliran data di mana komponen-komponen tersebut, dan asal, tujuan, dan

penyimpanan dari data tersebut. Kita dapat menggunakan *DFD* untuk dua hal utama, yaitu untuk membuat dokumentasi dari sistem informasi yang ada, atau untuk menyusun dokumentasi untuk sistem informasi yang baru

Empat simbol yang digunakan :

Tabel 2.4 Simbol *DFD*

Notasi	Notasi <i>Gane</i>	Fungsi
<i>Yourdon</i>	<i>Sarson</i>	
<i>DeMarco</i>		
		Simbol Entitas eksternal atau <i>terminator</i> menggambarkan asal atau tujuan data di luar sistem
		Simbol lingkaran menggambarkan entitas atau proses dimana aliran data masuk ditransformasikan ke aliran data keluar
		Simbol aliran data menggambarkan aliran data
		Simbol <i>file</i> menggambarkan tempat data disimpan

Sumber : Analisis Sistem Informasi (Sutabri , 2004)

Ada 3 (tiga) jenis *DFD*, yaitu :

1. *Diagram contex.*

Jenis pertama *Context Diagram*, adalah *data flow diagram* tingkat atas (*DFD Top Level*), yaitu diagram yang paling tidak detail, dari sebuah sistem informasi yang menggambarkan aliran-aliran data ke dalam dan ke luar sistem dan ke dalam dan ke luar *entitas-entitas eksternal*. (CD menggambarkan sistem dalam satu lingkaran dan hubungan dengan *entitas* luar. Lingkaran tersebut menggambarkan keseluruhan proses dalam sistem).

Beberapa hal yang harus diperhatikan dalam menggambar CD :

a. Terminologi sistem :

- 1) Batas Sistem adalah batas antara “daerah kepentingan sistem”.
- 2) Lingkungan Sistem adalah segala sesuatu yang berhubungan atau mempengaruhi sistem tersebut.
- 3) *Interface* adalah aliran yang menghubungkan sebuah sistem dengan lingkungan sistem tersebut.

b. Menggunakan satu simbol proses.

Yang masuk didalam lingkaran konteks (simbol proses) adalah kegiatan pemrosesan informasi (Batas Sistem). Kegiatan informasi adalah mengambil data dari *file*, *mentransformasikan* data, atau melakukan *filang* data, misalnya mempersiapkan dokumen, memasukkan, memeriksa, mengklasifikasi, mengatur, menyortir, menghitung, meringkas data, dan melakukan *filang* data (baik yang

melakukan secara manual maupun yang dilakukan secara *terotomasi*).

- c. Nama/keterangan di simbol proses tersebut sesuai dengan fungsi sistem tersebut.
- d. Antara *Entitas Eksternal/Terminator* tidak diperbolehkan komunikasi langsung.
- e. Jika terdapat *termintor* yang mempunyai banyak masukan dan keluaran, diperbolehkan untuk digambarkan lebih dari satu sehingga mencegah penggambaran yang terlalu rumit, dengan memberikan tanda *asterik* (*) atau garis silang (#).
- f. Jika Terminator mewakili individu (personil) sebaiknya diwakili oleh peran yang dipermainkan *personil* tersebut.
- g. Aliran data ke proses dan keluar sebagai output keterangan aliran data berbeda.

2. *DFD fisik*.

DFD fisik adalah *representasi* grafik dari sebuah sistem yang menunjukkan *entitas-entitas internal* dan *eksternal* dari sistem tersebut, dan aliran-aliran data ke dalam dan keluar dari *entitas-entitas* tersebut. *Entitas-entitas internal* adalah *personal*, tempat (sebuah bagian), atau mesin (misalnya, sebuah komputer) dalam sistem tersebut yang *mentransformasikan* data. Maka *DFD* fisik tidak menunjukkan apa yang dilakukan, tetapi menunjukkan dimana, bagaimana, dan oleh siapa proses-proses dalam sebuah sistem dilakukan.

Perlu diperhatikan didalam memberikan keterangan di lingkaran-lingkaran (simbol proses) dan aliran-aliran data (simbol aliran data) dalam *DFD* fisik menggunakan label/keterangan dari kata benda untuk menunjukkan bagaimana sistem *mentransmisikan* data antara lingkaran-lingkaran tersebut.

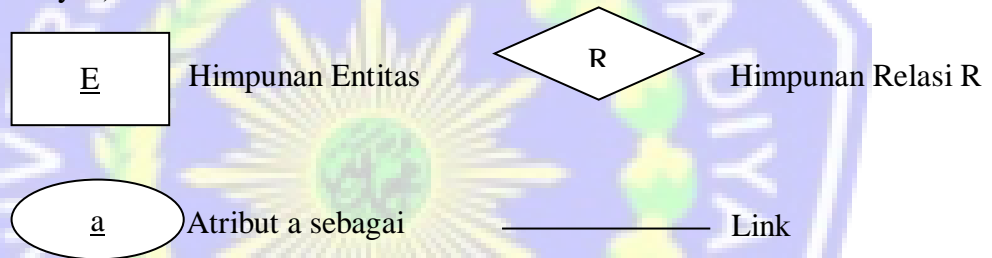
3. *DFD* logis.

DFD Logis adalah *representasi* grafik dari sebuah sistem yang menunjukkan proses-proses dalam sistem tersebut dan aliran-aliran data ke dalam dan ke luar dari proses-proses tersebut. Kita menggunakan *DFD logis* untuk membuat dokumentasi sebuah sistem informasi karena *DFD logis* dapat mewakili logika tersebut, yaitu apa yang dilakukan oleh sistem tersebut, tanpa perlu menspesifikasi dimana, bagaimana, dan oleh siapa proses-proses dalam sistem tersebut dilakukan. Keuntungan dari *DFD logis* dibandingkan dengan *DFD* fisik adalah dapat memusatkan perhatian pada fungsi - fungsi yang dilakukan sistem.

K. *ERD* (*Entity Relationship Diagram*)

Model *Entity-Relationship* yang berisi komponen-komponen Himpunan Entitas dan Himpunan Relasi yang masing-masing dilengkapi dengan atribut-atribut yang mempresentasikan seluruh fakta dari 'dunia nyata' yang kita tinjau, dapat digambarkan dengan lebih sistematis dengan menggunakan Diagram *Entity-Relationship* (Diagram E-R). Notasi-notasi simbolik didalam Diagram E-R yang dapat kita gunakan adalah :

1. Persegi panjang, menyatakan Himpunan Entitas.
2. Lingkaran/*Ellips*, menyatakan Atribut (Atribut yang berfungsi sebagai *key* digarisbawahi).
3. Belah ketupat, menyatakan Himpunan Relasi.
4. Garis, sebagai penghubung antara Himpunan Relasi dengan Himpunan Entitas dengan Atributnya.
5. Kardinalitas relasi dapat dinyatakan dengan banyaknya garis cabang atau dengan pemakaian angka (1 dan 1 untuk relasi satu-ke-satu, 1 dan N untuk relasi satu-ke-banyak atau N dan N untuk relasi banyak-ke-banyak).



Gambar 2.3 Kardinalitas Relasi
 Sumber : Analisis Sistem Informasi (Sutabri , 2004)

L. Adobe Dreamweaver CS6

Adobe Dreamweaver CS6 merupakan *HTML editor professional* yang berfungsi mendesain, melakukan *editing* dan mengembangkan aneka *website*. Salah satu kelebihan *Adobe Dreamweaver CS6* yaitu ruang kerja *Adobe Dreamweaver CS6* beserta *tools* yang tersedia dapat digunakan dengan sangat mudah dan cepat sehingga anda bisa membangun suatu *website* dengan cepat dan tanpa harus melakukan *coding*. Selain itu, *Adobe Dreamweaver CS6* juga mempunyai integrasi dengan produk *macromedia*

lainnya, seperti *flash* dan *firework*, *flash* sudah sangat terkenal sebagai sebagai program untuk membuat animasi yang berbasis *web* dengan perkembangan kebutuhan dan teknologi, *flash* akhir-akhir ini juga digunakan untuk membuat animasi dan video.

