



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB II

LANDASAN TEORI

2.1. Sistem Pakar

Sistem pakar (*expert system*) biasa disebut juga *knowledge-based system* atau *knowledge-based expert system* adalah salah satu cabang kecerdasan buatan yang menggunakan pengetahuan-pengetahuan khusus yang dimiliki oleh seorang ahli untuk menyelesaikan suatu masalah tertentu (Giarratano dan Riley, 2005). Konsep dasar dari sistem pakar adalah *user* memberikan masukan berupa fakta atau informasi dan mendapat hasil dari pakar. Dalam konsep dasar sebuah sistem pakar terdapat (Russel dan Norvig, 2010):

1. *Knowledge Base*: menyediakan pengetahuan (*knowledge*) yang akan ditelusuri oleh *Inference Engine*. Pengetahuan pada *knowledge base* bisa direpresentasikan dengan berbagai cara. Cara paling umum adalah dengan model IF-THEN.
2. *Inference Engine*: menarik kesimpulan berdasarkan pengetahuan dari *Knowledge Base*. Metode inferensi paling umum adalah *forward chaining*, yaitu menelusuri dari fakta-fakta yang ada untuk menarik kesimpulan, dan *backward chaining*, yaitu mencari fakta-fakta jika diketahui suatu kesimpulan.

Sebuah sistem pakar biasa dirancang untuk memenuhi karakteristik berikut (Giarratano dan Riley, 2005):

1. Kinerja tinggi: mampu memberi respon yang mampu menandingi seorang pakar
2. Waktu respon yang cukup: mampu memberi respon dalam waktu yang cukup cepat, yang mampu menandingi waktu respon seorang pakar
3. Keandalan yang baik: sistem harus bisa diandalkan dan tidak mudah crash

4. Bisa dipahami: dapat menjelaskan mengapa suatu hasil bisa didapat dan bukan memberikan hasil tanpa alasan, seperti pakar bisa menjelaskan alasannya

2.2. Kulit dan Penyakit Kulit pada Manusia

Kulit merupakan bagian terluar tubuh manusia yang juga merupakan indera peraba. Lapisan kulit dari yang terluar terdiri dari Epidermis, Dermis, dan Hipodermis (Zimmerman, 2013). Kulit memisahkan organ dalam manusia dengan lingkungan luar. Kulit juga membantu mengatur suhu tubuh seseorang. Penyakit kulit selalu bisa timbul karena hal tersebut, oleh karena itu pakar kulit (dermatologis) bertugas untuk menangani masalah itu. Ada sangat banyak penyakit kulit yang bisa timbul pada kulit seseorang dari penyakit ringan dan mudah diobati hingga sangat parah dan bahkan bisa menyebabkan kematian. Untuk itu masalah kulit harus selalu diwaspadai. Berikut adalah delapan penyakit kulit yang sering ditemui dermatologis di Indonesia menurut pakar pada penelitian ini.

2.2.1. Dermatitis Atopik

Dermatitis atopik (DA) atau seringkali disebut eksim memiliki karakteristik kulit kering dan gatal, dan bila terlalu banyak diusap atau digaruk dapat menyebabkan peradangan (Wolff dan Johnson, 2009). Penyakit ini dapat dialami oleh siapa saja pada segala usia namun lebih sering terjadi saat usia balita dan lebih banyak terjadi pada laki-laki dibanding perempuan. Pada anak-anak, dermatitis biasa muncul di wajah atau bagian dalam lipatan dan pada dewasa, dapat pula terjadi pada bagian luar lipatan. Dermatitis atopik termasuk penyakit residif sehingga setelah sembuh bisa saja kambuh lagi di lain waktu. Penyakit ini merupakan penyakit turunan sehingga berhubungan dengan riwayat DA pribadi

maupun keluarga serta riwayat alergi *rhinitis* dan asma karena besar kemungkinan balita penderita DA dapat menderita asma di kemudian hari. Penderita DA sebaiknya tidak menggaruk atau mengusap bagian yang gatal tapi bisa menggunakan *lotion* anti gatal. Dermatitis atopik bisa diatasi dengan terapi menggunakan *emollient*, *topical calcieurin inhibitor*, dan *topical steroid*. Gambar 2.1. menunjukkan contoh dermatitis atopik pada telapak tangan.



Gambar 2.1. Dermatitis atopik di telapak tangan (MedicineNet, 2013)

2.2.2. Dermatitis Kontak

Dermatitis kontak adalah istilah umum dari peradangan akut atau kronis kulit akibat reaksi dengan zat atau bahan tertentu (Wolff dan Johnson, 2009). Dermatitis kontak disebabkan reaksi alergi terhadap zat kimia tertentu (contoh: sabun cuci, obat pembersih). Dermatitis kontak bisa dihindari dengan mengurangi kontak dengan zat kimia tersebut, misalnya dengan menggunakan pelindung seperti sarung tangan dan masker ataupun krim pelindung.

2.2.3. Jerawat (Akne Vulgaris)

Jerawat adalah penyakit kulit yang sangat umum dan terjadi pada setidaknya 85% kaum remaja. Jerawat bisa terjadi di bagian-bagian tubuh, khususnya wajah

dan cenderung lebih banyak dialami laki-laki. Jerawat merupakan peradangan dari unit *pilosebaceous*, unit di mana terdapat kelenjar *sebaceous* yang menghasilkan sebum (zat berminyak) dan bisa berwujud komedo, *papulopustule*, *nodule*, dan kista. Gambar 2.2. menunjukkan contoh jerawat berbentuk *pustule* dan komedo.



Gambar 2.2. Akne Vulgaris berbentuk *pustule* dan komedo (Moskowitz, 2013)

2.2.4. Melasma (Flek)

Melasma atau biasa dikenal sebagai flek adalah hiperpigmentasi warna coklat muda atau tua yang biasa terjadi di area terbuka seperti wajah dan disebabkan pancaran sinar matahari (Wolff dan Johnson, 2009). *Melasma* lebih banyak dialami oleh perempuan. *Melasma* juga bisa disebabkan kehamilan, penggunaan kontrasepsi, serta pengobatan tertentu. *Melasma* yang disebabkan kehamilan bisa hilang seiring berjalannya waktu setelah melahirkan atau setelah berhenti menggunakan kontrasepsi atau pengobatan tertentu. *Melasma* dapat dihindari dengan penggunaan tabir surya saat keluar ruangan. Gambar 2.2. menunjukkan seorang wanita yang menderita melisma.



Gambar 2.3. *Melasma*
(Wolff dan Johnson, 2009)

2.2.5. Tinea (Jamur kulit)

Tinea merupakan kelainan kulit yang disebabkan oleh infeksi jamur. Penggunaan istilah tinea biasanya diikuti dengan lokasi anatomi yang terinfeksi, misalnya *tinea pedis* pada kaki (Wolff dan Johnson, 2009) seperti yang ditunjukkan Gambar 2.4. Tinea biasanya tidak bergejala namun bisa saja terasa gatal dan sakit. Wujud infeksi jamur bisa bervariasi baik bentuk maupun warnanya. Jamur kulit bisa diatasi salah satunya menggunakan krim atau bedak anti jamur.



Gambar 2.4. *Tinea Pedis* pada kaki
(Goldsmith, dkk., 2012)

2.2.6. Pioderma

Abses adalah peradangan lokal akut atau kronik dan biasanya bernanah atau adanya kerusakan jaringan kulit (Wolff dan Johnson, 2009). Abses biasanya terjadi pada anak-anak dan remaja, khususnya laki-laki. Abses ditandai dengan

pembengkakan di kulit berwarna merah dan bernanah. Pengobatan untuk abses bisa berupa operasi serta terapi *antimicrobial* sistemik.

2.2.7. Tumor Kulit

Ada tiga tumor kulit yang paling umum yaitu *Melanoma*, *Karsinoma sel basal*, dan *Karsinoma sel squamosa*. *Melanoma* merupakan tumor kulit paling ganas. Awalnya melanoma berasal dari tumor jinak namun kemudian menjadi ganas (Wolff dan Johnson, 2009). Ada 6 tanda untuk mengenali melanoma ganas yaitu asimetris, *border*/pinggiran tidak wajar, warnanya tidak rata, diameter cenderung besar, dan biasanya memiliki tonjolan yang tidak wajar.

Karsinoma sel basal adalah kanker kulit paling umum pada manusia. Penyakit ini disebabkan radiasi ultraviolet serta mutasi gen *PTCH*, yaitu salah satu gen penekan sel kanker (Wolff dan Johnson, 2009). Penderita biasanya berusia 40 tahun ke atas dan cenderung jarang diderita orang berkulit coklat dan hitam. Kanker kulit ini sangat berbahaya namun perkembangannya lambat dan hampir tidak mungkin menyebar. Gambar 2.5. menunjukkan contoh wujud *karsinoma sel basal*.



Gambar 2.5. *Karsinoma sel basal* (Goldsmith, dkk., 2012)

Karsinoma sel squamosa adalah tumor ganas pada keratinosit yang muncul di lapisan *epidermis* (Wolff dan Johnson, 2009). Wujud tumor ini adalah berupa flek atau nodul dengan keratinisasi yang bervariasi. Penyebab penyakit ini antara lain

yaitu radiasi ultraviolet, infeksi HPV (*Human Papillomavirus*), *imunosupresi*, peradangan kronis, karsinogen industri, dan arsenik inorganik.

Pengobatan tumor ganas kulit biasanya dilakukan melalui operasi atau radioterapi jika operasi tidak memungkinkan.

2.2.8. Dermatitis Akibat Penyakit Sistemik

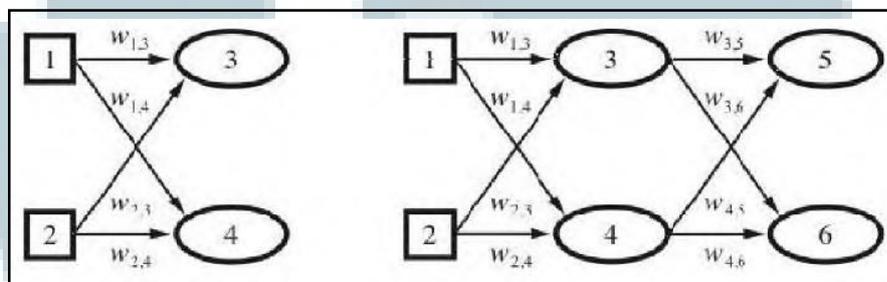
Penyakit sistemik seperti diabetes, *hipertensi*, gangguan hati, dan gangguan ginjal dapat menyebabkan berbagai gangguan kulit. Pada diabetes misalnya, gangguan kulit itu seringkali disebabkan penggunaan insulin serta infeksi bakteri atau jamur. Gejala dari penyakit ini biasanya cenderung mirip dengan dermatitis lainnya namun yang membedakan adalah penderita penyakit ini memiliki riwayat penyakit sistemik.

2.3. Artificial Neural Network

Jaringan saraf tiruan (*Artificial Neural network*) merupakan salah satu bagian dari kecerdasan buatan. Ide dari jaringan saraf tiruan adalah meniru sistem otak manusia untuk diimplementasikan dalam komputer digital karena kerja otak manusia dalam mengenali atau memutuskan sesuatu lebih cepat dari komputer. Menurut Haykin (1999) sebuah jaringan saraf adalah sebuah prosesor yang terdistribusi paralel dan mempunyai kecenderungan untuk menyimpan pengetahuan yang didapatkannya dari pengalaman dan membuatnya tetap tersedia untuk digunakan. Hal ini menyerupai kerja otak dalam dua hal yaitu:

1. Pengetahuan diperoleh oleh jaringan melalui suatu proses belajar.
2. Kekuatan hubungan antar sel saraf yang dikenal dengan bobot sinapsis digunakan untuk menyimpan pengetahuan.

Sebuah jaringan saraf tiruan terdiri dari *nodes* atau unit yang terhubung dengan garis terarah. Tiap unit memiliki bobotnya masing-masing yang mewakili kekuatan masing-masing hubungan. Sebuah jaringan saraf tiruan bisa memiliki satu *layer* saja atau lebih yang disebut *single-layer perceptrons* dan *multi-layer perceptrons*. Setiap *layer*-nya bisa menerima *input* dan menghasilkan *output* lebih dari satu. Pada *single-layer perceptrons* setiap *input* yang diberikan langsung terhubung pada *output*-nya, sedangkan pada *multi-layer perceptrons* terdapat *layer* diantara *input* dan *output* yang biasa disebut *hidden layer*.

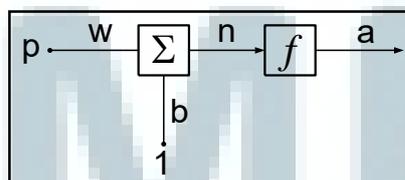


Gambar 2.6. *Neural network single-layer* dengan 2 *input* dan 2 *output* serta *neural network multi-layer* dengan 2 *input* dan 2 *output* (Russell dan Norvig, 2009)

Model matematis dari *single-input neuron* (*neuron* dengan satu *input*) adalah

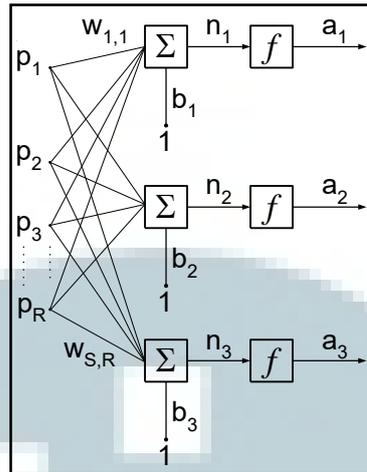
$$a = f(n) = f(w.p + b) \dots\dots\dots (2.1)$$

dengan *p* sebagai *input*, *w* adalah bobot, *b* adalah bias, *a* adalah *output*, dan *f(n)* adalah fungsi transfer.



Gambar 2.7. *Single-input neuron*

Fungsi yang serupa juga dapat diterapkan pada *multiple-input neuron* pada tiap-tiap *neuron* hingga mendapat hasil akhir.



Gambar 2.8. *Multiple-input neuron*

Pada *neural network* tentunya juga diperlukan *learning rule*, yaitu untuk menentukan bobot dan bias yang tepat pada *perceptron*. Ada 3 macam *learning rule* untuk *perceptron*:

1. *Supervised learning*: mempelajari setiap pasangan *input* dan *output* dengan fungsi yang memetakannya. Termasuk di dalamnya adalah algoritma *backpropagation*
2. *Unsupervised learning*: mempelajari pola pada *input* tanpa *output*.
3. *Reinforcement learning*: mirip *supervised learning* tapi mengenal adanya *reward* dan *punishment* sebagai penilaian.

2.4. Algoritma Backpropagation

Algoritma *backpropagation* dalam *neural network* biasa diterapkan pada *multi-layer neural network* untuk membantu menentukan bobot yang tepat pada *hidden layer*-nya dengan mempropagasi balik *error* dari *output layer*. *Backpropagation* termasuk algoritma yang digunakan dalam *supervised learning* (Papagelis dan Kim, Tanpa Tahun) karena pada *backpropagation* pelatihan dilakukan dengan *input* dan *output* yang sudah diketahui. *Input* dipropagasikan maju untuk didapat *output* sementara *output* digunakan untuk menghitung *error*

yang akan dipropagasikan balik hingga didapat bobot yang tepat (Russel dan Norvig, 2009). Proses dalam *backpropagation* meliputi:

1. Forward Propagation

Pada tahap ini, nilai *input* dan *output* dari setiap unit *hidden layer* akan dipropagasikan ke *hidden layer* menggunakan fungsi aktivasi yang ditentukan dan seterusnya hingga menghasilkan nilai *output* pada jaringan. Tahap ini dapat dirumuskan:

$$a_j = g(\sum_i w_{i,j} a_i + b_j) \dots\dots\dots(2.2)$$

Kemudian *error* akan didapat dari selisih *output* dengan target. Jika *error* di atas batas toleransi, maka akan dilakukan *weight update* sesuai tahap berikutnya. Jika *error* bisa ditoleransi, maka iterasi dihentikan.

2. Backpropagation

Pada tahap ini, ditentukan nilai Δ_j menurut error yang didapatkan, kemudian dipropagasikan kembali ke *layer* sebelumnya dan seterusnya hingga semua *node* yang terhubung langsung antara *hidden* dan *output layer* dihitung. Berikut rumus tahap backpropagation untuk layer output:

$$\Delta_j = g'(\sum_i w_{i,j} a_i + b_j) * (t - a_j) \dots\dots\dots(2.3)$$

Perhitungan tahap backpropagation untuk layer-layer sebelumnya dapat dirumuskan sebagai:

$$\Delta_i = g'(\sum_i w_{i,j} a_i + b_j) * \sum_i w_{i,j} \Delta_j \dots\dots\dots(2.4)$$

3. Weight Update

Lakukan *update* bobot sesuai nilai Δ yang telah dihitung jika *error* masih di atas batas toleransi. Tahap *update* bobot dipengaruhi *learning rate* dan *momentum*. Update bobot dapat dirumuskan sebagai:

$$w_{i,j} = w_{i,j} + \Delta w_{i,j} \dots \dots \dots (2.5)$$

Learning rate adalah konstanta yang cenderung bernilai kecil yang mengindikasikan perubahan bobot (Papagelis dan Kim, Tanpa Tahun). *Learning rate* berguna untuk mengatur cepat lambatnya proses pembelajaran jaringan saraf tiruan. *Learning rate* biasanya bernilai antara nol sampai dengan satu. Semakin kecil *learning rate*, pembelajaran akan semakin lambat, namun *output* bisa semakin akurat. Sebaliknya, jika *learning rate* terlalu besar, maka *output* jaringan akan semakin menyimpang meski pembelajaran jaringan saraf tiruan lebih cepat.

Momentum berguna untuk mempercepat pelatihan dengan menyesuaikan bobot dengan perubahan bobot dari iterasi sebelumnya. *Momentum* biasanya bernilai antara nol sampai dengan satu (Papagelis dan Kim, Tanpa Tahun).

```

function BACK-PROP-LEARNING( examples, network) returns a neural
network
  inputs: examples, a set of examples, each with input vector x and
output vector y network, a multilayer network with L layers, weights
wi,j activation function g
  local variables:  $\Delta$ , a vector of errors, indexed by network node

  repeat
  for each weight wi,j in network do
    wi,j  $\leftarrow$  a small random number
  for each example (x, y) in examples do
    /* Propagate the inputs forward to compute the outputs */
    for each node i in the input layer do
      ai  $\leftarrow$  xi
    for l = 2 to L do
      for each node j in layer l do
        inj  $\leftarrow$   $\sum_i w_{i,j} a_i$ 
        aj  $\leftarrow$  g(inj)
    /* Propagate deltas backward from output layer to input layer */
    for each node j in the output layer do
       $\Delta|_j| \leftarrow g'(in_j) * (y_j - a_j)$ 
    for l = L - 1 to 1 do
      for each node i in layer l do
         $\Delta|i| \leftarrow g'(in_i) \sum_i w_{i,j} \Delta|_j|$ 
    /* Update every weight in network using deltas */
    for each weight wi,j in network do
      wi,j  $\leftarrow$  wi,j +  $\alpha * a_i * \Delta|_j|$ 
  until some stopping criterion is satisfied
  return network

```

Gambar 2.9. Pseudocode algoritma learning backpropagation (Russell dan Norvig, 2009)

Pseudocode pada Gambar 2.9 hanya menggunakan *learning rate* dalam perhitungan bobot baru. Momentum digunakan dengan mengalikannya dengan delta bobot iterasi sebelumnya. Delta bobot yang dihitung tanpa momentum dirumuskan sebagai:

$$\Delta w_{i,j} = \alpha * a_i * \Delta|j| \dots\dots\dots(2.6)$$

Sementara delta bobot yang dihitung dengan momentum (μ) dirumuskan sebagai:

$$\Delta w_{i,j}^n = \alpha * a_i * \Delta|j| + \mu * \Delta w_{i,j}^{n-1} \dots\dots\dots(2.7)$$

2.5. Pemrograman Java dan Android

Android merupakan sebuah sistem operasi untuk perangkat *mobile* seperti *smartphone* dan *tablet*. Android adalah sebuah sistem yang berjalan pada Linux 2.6 kernel (DiMarzio, 2008). Pengembangan Android dilakukan secara *open source*. Saat ini Android menjadi *platform* paling populer dalam dunia *mobile*. Android menyediakan *platform* yang luar biasa untuk mengembangkan aplikasi dan *game* bagi semua penggunanya dan menyediakan pasar terbuka untuk penyebaran aplikasi itu (Google(1), Tanpa Tahun). Android juga menyediakan *tools* yang sangat efisien bagi pengembang untuk mengembangkan aplikasi Android dengan adanya *Android Developer Tools* yang berisi IDE Java dengan fitur untuk pengembangan, *debug*, dan mengemas aplikasi Android (Google (2) , Tanpa Tahun). Versi terbaru Android saat penelitian ini dilakukan adalah versi 5.1 (Lollipop).

Android menyediakan *framework* yang kaya untuk membangun aplikasi dan permainan yang inovatif. Aplikasi Android ditulis dengan bahasa pemrograman Java. Oracle (Tanpa Tahun), sebagai pengembang bahasa Java saat ini

mendeskripsikan bahasa Java merupakan bahasa tingkat tinggi yang memiliki karakteristik *simple, object oriented, distributed, multithreaded, dynamic, architecture netral, portabel, high performance, robust, dan secure*. Kode aplikasi yang sudah ditulis kemudian akan di-*compile* oleh Android SDK (*Software Development Kit*) *Tools* beserta semua data dan *resource* menjadi sebuah APK (*Android Package*). Secara *default* aplikasi Android hanya memiliki akses ke komponen-komponen yang dibutuhkan saja sehingga jika tidak ada izin, sebuah aplikasi tidak boleh mengakses komponen lain dari sistem.

Aplikasi merupakan kombinasi komponen terpisah yang bisa dipanggil secara individu (Google(3) , Tanpa Tahun). Sebuah komponen bisa memanggil untuk memulai komponen lain menggunakan *intent*. Komponen di aplikasi lain juga bisa dipanggil melalui komponen yang sedang berjalan, misalnya memanggil komponen di aplikasi maps untuk menunjukkan lokasi alamat. Komponen aplikasi adalah bagian penting dari aplikasi yang menandai bagaimana sistem mengakses aplikasi. Ada empat jenis komponen:

- *Activities*: sebuah layar tunggal dengan *user interface*.
- *Services*: komponen yang berjalan di *background* untuk mengerjakan proses yang membutuhkan waktu lama atau untuk proses *remote*.
- *Content providers*: komponen yang mengatur data aplikasi baik disimpan dalam *file system, database SQLite, web, atau lokasi penyimpanan lain*.
- *Broadcast receivers*: komponen yang menanggapi pengumuman *broadcast* seluruh sistem.

Activities, services, dan broadcast receivers membutuhkan *intent* untuk aktivasi. *Intent* berguna untuk mengikat satu komponen ke komponen lain.

Sementara *content providers* diaktifkan ketika ada permintaan dari *content resolver*. Setiap komponen harus dideklarasikan dalam *file manifest* aplikasi tersebut karena sistem akan membaca *file* tersebut sebelum memulai suatu komponen. *File manifest* juga berisi kemampuan komponen tersebut serta izin-izin kebutuhan aplikasi. Aplikasi juga memerlukan *resource* yang berisi segala hal yang berkaitan dengan presentasi *visual* aplikasi yang terpisah dari kode program seperti media (gambar, suara), *layout user interface*, dan lain sebagainya.

