

BAB II

LANDASAN TEORI

2.1 Citra

Menurut Munir (2004, p.2) secara harfiah citra (*image*) adalah gambar pada bidang dwimatra atau dua dimensi. Citra juga dapat diartikan sebagai kumpulan titik-titik dengan intensitas warna tertentu yang membentuk suatu kesatuan dan mempunyai pengertian artistik. Citra sebagai salah satu komponen multimedia yang memegang peranan sangat penting sebagai salah satu bentuk informasi visual.

Lestari (2012, p.2) menjelaskan bahwa Citra sebagai keluaran dari suatu sistem perekaman data dapat bersifat optik berupa foto, bersifat analog berupa sinyal video seperti gambar pada monitor televisi, dan bersifat digital yang dapat langsung disimpan pada suatu pita magnetik. Citra sebagai salah satu komponen multimedia memegang peranan penting sebagai bentuk informasi visual. Citra mempunyai karakteristik yang tidak dimiliki oleh data teks, yaitu citra kaya dengan informasi.

2.1.1 Citra Warna

Jain dalam Arifin (2009, p.16) menjelaskan bahwa citra warna adalah citra dengan sistem grafik yang memiliki satu set nilai tersusun (*aset of ordered values*) yang menyatakan berbagai tingkat warna. Citra warna bukanlah seperti citra *grayscale*. Dimana setiap set nilai tersusun mewakili satu 'scale' warna atau 'hue'.

Gonzales yang diterjemahkan oleh Arifin (2009, p.17) menyatakan bahwa sistem yang dipakai untuk mewakili warna yaitu sistem RGB (*Red, Green, Blue*). Sistem RGB adalah sistem penggabungan antara warna-warna primer (*additive primary colours*) yaitu merah (*Red*), hijau (*Green*) dan biru (*Blue*) untuk memperoleh warna tertentu. Misalnya warna putih diperoleh dari hasil gabungan warna merah =255, hijau = 255, dan biru = 255. Dalam

sistem RGB, warna putih cerah dinyatakan dengan RGB (255, 255, 255). *Range* nilai dari setiap warna primer adalah 0 sampai 255. Sehingga kemungkinan warna yang dapat terbentuk dengan sistem RGB adalah $256 \times 256 \times 256$ yakni kurang lebih 16.7 juta warna. Tabel kode warna seperti tertampil pada tabel 2.1 berikut diperlihatkan beberapa kode warna hasil gabungan warna RGB.

Tabel 2.1 Kode Warna dalam Nilai Integer

| Warna | Merah | Hijau | Biru |
|---------------------------------|-------|-------|------|
| Hitam | 0 | 0 | 0 |
| Biru | 0 | 0 | 255 |
| Hijau | 0 | 255 | 0 |
| Cyan (Biru+Hijau) | 0 | 255 | 255 |
| Merah | 255 | 0 | 0 |
| Magenta (Merah+Biru) | 255 | 0 | 255 |
| Kuning (Merah+Hijau) | 255 | 255 | 0 |
| Putih (Merah+Hijau+Biru) | 255 | 255 | 255 |
| Abu-abu | 128 | 128 | 128 |

2.1.2 Citra *Monokrom*

Munir dalam Arifin (2004, p.17) menyatakan bahwa citra monokrom adalah citra dengan suatu sistem grafik yang tidak memiliki kemampuan warna selain warna hitam atau warna putih. Perbedaan hanya diperoleh dengan menentukan tingkat intensitas *grayscale*. Nilai numerik yang digunakan biasanya adalah *range* 0 – 1. Citra monokrom yang diwakili dengan beberapa nilai kekuatan cahaya bernilai dari hitam sampai putih sebagai *grayscale image*. Tabel intensitas *grayscale* seperti yang terlihat pada tabel 2.2 terdapat empat tingkat intensitas yang dapat ditampilkan seperti yang terlihat dibawah ini.

Tabel 2.2 Intensitas *Grayscale*

| Kode Intensitas | Nilai Intensitas | Biner | Tingkat Intensitas yang ditampilkan |
|-----------------|------------------|-------|-------------------------------------|
| 0 | 0 | 0 0 | Hitam |
| 0.33 | 1 | 0 1 | Abu-abu gelap |
| 0.67 | 2 | 1 0 | Abu-abu terang |
| 1 | 3 | 1 1 | Putih |

Arifin (2009, p.18) menyatakan bahwa menyimpan tingkat intensitas dalam memori layar sama dengan menyimpan kode warna. Titik dengan ukuran 3 bit bisa menampilkan 8 tingkat intensitas sedangkan titik berukuran 1 bit hanya bisa menampilkan warna hitam (*black*) dan putih (*white*) saja. Pada Tabel 2.2 terdapat empat tingkat intensitas yang dapat ditampilkan. Nilai intensitas mendekati 0.33 akan disimpan dengan nilai biner 0 1 dalam memori *layer* dan menghasilkan titik dengan tingkat intensitas *darkgray* atau abu-abu kehitaman. Sedangkan Tingkat intensitas itu sendiri ditentukan oleh program aplikasi kemudian diubah menjadi nilai biner yang sesuai.

2.1.3 Representasi Citra Digital

Yuswanto (2002, p.23) berpendapat, data-data dalam sistem komputer perlu dikodekan dengan menggunakan suatu sistem simbol diskrit. Sebuah citra digital dapat dianggap suatu matriks dimana baris dan kolomnya menunjukkan sebuah titik pada citra dan nilai elemen matriks menunjukkan tingkat keabuan (*graylevel*) pada titik tersebut. Elemen dari *array* digital tersebut disebut *picture elements (pixel)*. Pada umumnya, citra digital yang direpresentasikan dengan $a(x,y)$ merupakan sebuah fungsi dari banyak variabel yang mencakup kedalaman/*depth* (z), warna/*colour* (λ), dan waktu/*time*(t).

Hendry dalam Arifin (2009, p.21) menjelaskan bahwa resolusi gambar dikatakan sebagai jumlah *pixel* yang terkandung di dalam suatu citra. Pada resolusi rendah keterperincian dan kedalaman citra akan hilang sama sekali dimana *pixel-pixel* individu jelas kelihatan, pada resolusi tinggi keterperincian data lebih nyata dan tajam. *Aspect Ratio* adalah suatu bilangan yang dapat diperoleh bila bilangan *pixel* mendatar dibagi dengan bilangan *pixel* tegak. Dalam pembesaran resolusi citra *Aspect Ratio* perlu sama agar citra tidak kelihatan *distorted* (menyimpang) dan alami.

2.2 Mengkonversi Jenis Citra

Menurut Kadir (2013, p.25) citra berwarna seringkali harus dikonversi ke dalam bentuk citra berskala keabuan mengingat banyak pemrosesan citra yang bekerja pada skala keabuan. Namun, terkadang citra berskala keabuan pun perlu dikonversi ke citra biner, karena beberapa operasi dalam pemrosesan citra berjalan pada citra biner. Secara umum citra berwarna dapat dikonversi ke citra berskala keabuan melalui rumus :

$$I = a \times R + b \times G + c \times B, a+b+c=1$$

dengan R menyatakan nilai komponen merah, G menyatakan nilai komponen hijau, dan B menyatakan nilai komponen biru. Sebagai contoh, sebuah piksel mempunyai komponen R, G, B sebagai berikut :

$$R = 50 ; G = 70 ; B = 61$$

Jika a, b, c pada persamaan 1.1 dibuat sama, akan diperoleh hasil sebagai berikut :

$$I = (50 + 70 + 61) / 3 = 60$$

Salah satu contoh rumus yang biasa dipakai untuk menubah ke skala keabuan yaitu :

$$I = 0,2989 \times R + 0,5870 \times G + 0,1141 \times B$$

2.3 Format File Gambar

Berdasarkan pendapat Andreswari (2001, p.22) pada umumnya file gambar digunakan untuk menyimpan gambar yang ditampilkan dilayar ke dalam suatu media penyimpanan data. Untuk menyimpan sebuah file gambar ini digunakan

salah satu format file. Ada banyak format file gambar yang dapat digunakan untuk menyimpan file gambar, salah satunya adalah BMP, JPG/JPEG.

2.3.1 Format File *BMP (Bitmap)*

Format file BMP merupakan format grafis yang fleksibel untuk platform Windows sehingga dapat dibaca oleh program grafis manapun. Hal tersebut didukung kuat oleh Andreswari (2001, p.22) dengan berpendapat bahwa, format ini mampu menyimpan informasi dengan kualitas tingkat 1 bit sampai 24 bit. Format file ini mampu menyimpan gambar dalam mode warna RGB, *Grayscale*, *Indexed Color*, dan *Bitmap*. Kelemahan format file ini adalah tidak mampu menyimpan alpha channel serta ada kendala dalam pertukaran platform. Kelebihan tipe file BMP adalah dapat dibuka oleh hampir semua program pengolah gambar. Baik file BMP yang terkompresi maupun tidak terkompresi, file BMP memiliki ukuran yang jauh lebih besar daripada tipe-tipe yang lain.

2.3.2 Format File *JPG/JPEG (Joint Photographic Experts Group)*

Menurut Andreswari (2001, p.22), Joint Photographic Experts (JPEG, dibaca *jay-peg*) di rancang untuk kompresi beberapa *full-color* atau *gray-scale* dari suatu gambar yang asli, seperti pemandangan asli di dunia ini. JPEG bekerja dengan baik pada *continuous tone images* seperti *photographs* tetapi tidak terlalu bagus pada ketajaman gambar dan seni pewarnaan seperti penulisan, kartun yang sederhana atau gambar yang menggunakan banyak garis.

2.4 Deteksi Tepi

Menurut Witeti (2004, p.2) Deteksi tepi merupakan salah satu proses prapengolahan yang sering dibutuhkan pada analisis citra. Proses tersebut bertujuan untuk meningkatkan penampakan garis pada citra, jadi prosesnya mempunyai sifat diferensiasi atau memperkuat komponen frekuensi tinggi.

Menurut Gonzales dan Woods dalam Kadir (2013, p.342) deteksi tepi berfungsi untuk memperoleh tepi objek. Deteksi tepi memanfaatkan perubahan nilai intensitas yang drastis pada batas dua area. Perlu diketahui, tepi sesungguhnya mengandung informasi sangat penting. Informasi yang diperoleh dapat berupa bentuk maupun ukuran objek. Umumnya, deteksi tepi menggunakan dua macam detektor, yaitu detektor baris (H_y) dan detektor kolom (H_x). Beberapa contoh yang tergolong jenis ini adalah operator *Roberts*, *Prewitt*, *Sobel* dan *Frei-Chen*. Beberapa operator pada deteksi tepi antara lain :

1. Operator *Roberts*

Operator *Roberts*, yang pertama kali dipublikasikan pada tahun 1965, terdiri atas dua filter berukuran 2×2 . Ukuran filter kecil membuat komputasi sangat cepat. Namun, kelebihan ini sekaligus menimbulkan kelemahan, yakni sangat terpengaruh oleh derau. Selain itu, operator *Roberts* memberikan tanggapan lemah terhadap tepi, kecuali jika tepi sangat tajam (Fisher, dkk., 2003).

Misalkan, f adalah citra yang akan dikenai operator *Roberts*. Maka, nilai operator *Roberts* pada (x,y) didefinisikan sebagai :

$$r(y,x) = (z_1 - z_4)^2 + (z_3 - z_2)^2 \dots\dots\dots (1.7)$$

Dalam hal ini, $z_1 = f(y,x)$, $z_2 = f(y,x+1)$, $z_3 = f(y+1,x)$, $z_4 = f(y+1,x+1)$.

2. Operator *Prewitt*

Menurut Kadir (2013, p.347) operator *Prewitt* dikemukakan oleh Prewitt pada tahun 1966. Untuk mempercepat komputasi, bagian bernilai nol tidak perlu diperoses. Oleh Karena itu, perhitungan dengan operator *Prewitt* ditulis menjadi :

$$r(y,x) = \text{sqrt}((f(y-1, x-1)+f(y, x-1)+f(y+1, x-1)-f(y-1, x+1)-f(y, x+1)-f(y+1, x+1))^2 + (f(y+1, x-1)+f(y+1, x)+f(y+1, x+1)-f(y-1, x-1)-f(y-1, x)-f(y-1, x+1))^2) \dots\dots\dots (1.8)$$

3. Operator *Sobel*

Operator *Sobel* lebih sensitif terhadap tepi diagonal daripada tepi vertikal dan horizontal. Hal ini berbeda dengan operator *Prewitt*, yang lebih sensitif terhadap tepi vertikal dan horizontal (Crane, 1997).

Rumusnya dapat ditulis :

$$G(y,x) = \text{sqrt}((f(y-1, x+1)+2*f(y, x+1)+f(y+1, x+1)-f(y-1, x-1)-2*f(y, x-1)-f(y+1, x-1))^2+f(y-1, x-1)+2*f(y-1, x)+f(y-1, x+1)-f(y+1, x-1)-2*f(y+1, x)-f(y+1, x+1))^2) \dots\dots\dots (1.9)$$

4. Operator *Canny*

Menurut Kadir (2013, p.361) Operator *Canny*, yang dikemukakan oleh John Canny pada tahun 1986, terkenal sebagai operator deteksi tepi yang optimal. Algoritma ini memberikan tingkat kesalahan rendah, melokalisasi titik-titik tepi (jarak piksel-piksel tepi yang ditemukan deteksi dan tepi yang sesungguhnya sangat pendek), dan hanya memberikan satu tanggapan untuk satu tepi.

Dapat dirumuskan :

$$g(i,j) = \text{sqrt}(\sqrt{gx^2 + gy^2})$$

dimana ;

$$gx = (g(i, j+1)-g(i, j)+g(i+1, j+1)-g(i+1, j))/2$$

$$gy = (g(i, j)-g(i+1, j)+g(i, j+1)-g(i+1, j+1))/2$$

2.5 *Support Vector Machine*

Support Vector Machine (SVM) pertama kali diperkenalkan oleh Vapnik pada tahun 1992 sebagai rangkaian harmonis konsep-konsep unggulan dalam bidang *pattern recognition*. Sebagai salah satu metode *pattern recognition*, usia SVM terbilang masih relatif muda. Walaupun demikian, evaluasi kemampuannya dalam berbagai aplikasinya menempatkannya sebagai *state of the art* dalam *pattern recognition*, dan dewasa ini merupakan salah satu tema yang berkembang dengan pesat. SVM adalah metode *learning machine* yang bekerja atas prinsip

Structural Risk Minimization (SRM) dengan tujuan menemukan *hyperplane* terbaik yang memisahkan dua buah *class* pada *input space*.

Menurut Santoso (2007) *Support vector machine* (SVM) adalah suatu teknik untuk melakukan prediksi, baik dalam kasus klasifikasi maupun regresi. SVM berada dalam satu kelas dengan *Artificial Neural Network* (ANN) dalam hal fungsi dan kondisi permasalahan yang bisa diselesaikan. Keduanya masuk dalam kelas *supervised learning*.

Teori SVM dimulai dengan kasus klasifikasi yang secara linier bisa dipisahkan. Dalam hal ini fungsi pemisah yang dicari adalah fungsi linier. Fungsi ini bisa didefinisikan sebagai;

$$g(x) := \text{sgn}(f(x)) \quad (2.1)$$

$$\text{dengan} \quad (2.2)$$

$$\text{atau} \quad (2.3)$$

$$\text{dimana } x, w \in$$

$$\text{and } b \in$$

Masalah klasifikasi ini bisa dirumuskan set parameter (w, b) sehingga $f(x_i) = \langle w, x \rangle + b = y_i$ untuk semua i . Teknik SVM berusaha menemukan fungsi pemisah (klasifier/hyperplane) terbaik diantara fungsi yang tidak terbatas jumlahnya untuk memisahkan dua macam obyek.

2.6 Segmentasi Citra

Segmentasi citra merupakan bagian dari proses pengolahan citra. Proses segmentasi citra ini lebih banyak merupakan suatu proses pra pengolahan pada sistem pengenalan objek dalam citra. Segmentasi citra (*image segmentation*) mempunyai arti membagi suatu citra menjadi wilayah-wilayah yang homogen berdasarkan kriteria keserupaan yang tertentu antara tingkat keabuan suatu piksel dengan tingkat keabuan piksel – piksel tetangganya, kemudian hasil dari proses segmentasi ini akan digunakan untuk proses tingkat tinggi lebih lanjut yang dapat dilakukan terhadap suatu citra, misalnya proses klasifikasi citra dan proses identifikasi objek. Adapun dalam proses segmentasi citra itu sendiri terdapat

beberapa algoritma, diantaranya : algoritma Deteksi Titik, Deteksi Garis, dan Deteksi Sisi (berdasarkan *Operator Robert* dan *Operator Sobel*).

Gonzalez dan Wintz (1987) menyatakan bahwa segmentasi adalah proses pembagian sebuah citra kedalam sejumlah bagian atau obyek. Segmentasi merupakan suatu bagian yang sangat penting dalam analisis citra secara otomatis, sebab pada prosedur ini obyek yang diinginkan akan disadap untuk proses selanjutnya, misalnya: pada pengenalan pola. Algoritma segmentasi didasarkan pada 2 buah karakteristik nilai derajat kecerahan citra, yaitu: *discontinuity* dan *similarity*. Pada item pertama, citra dipisahkan/dibagi atas dasar perubahan yang mencolok dari derajat kecerahannya. Aplikasi yang umum adalah untuk deteksi titik, garis, area, dan sisi citra. Pada kategori kedua, didasarkan atas *thresholding*, *region growing*, dan *region spiltting and marging*. Prinsip segmentasi citra bisa diterapkan untuk citra yang statis maupun dinamis. Segmentasi citra adalah membagi suatu citra menjadi wilayah-wilayah yang homogen.

2.7 Osteoporosis

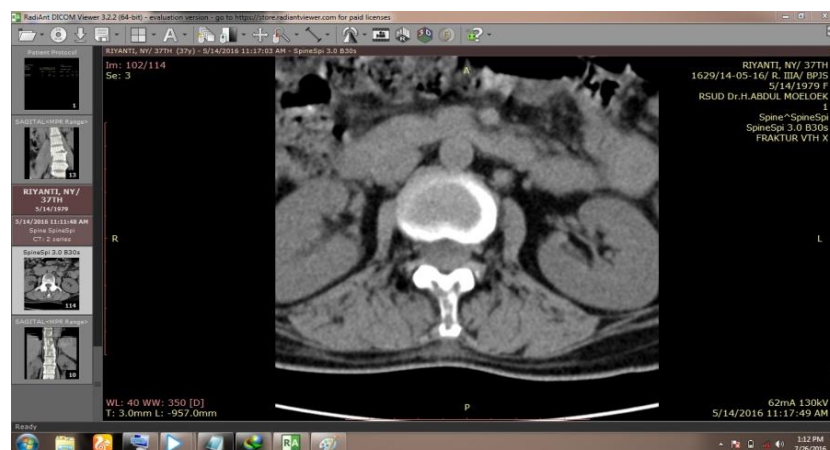
2.7.1 Pengertian Osteoporosis

Osteoporosis berasal dari kata *osteo* dan *porous*, *osteo* artinya tulang, dan *porous* berarti berlubang-lubang atau keropos. Jadi, *osteoporosis* adalah tulang yang keropos, yaitu penyakit yang mempunyai sifat khas berupa massa tulangnya rendah atau berkurang, disertai gangguan mikro-arsitektur tulang dan penurunan kualitas jaringan tulang, yang dapat menimbulkan kerapuhan tulang (Tandra, 2009).

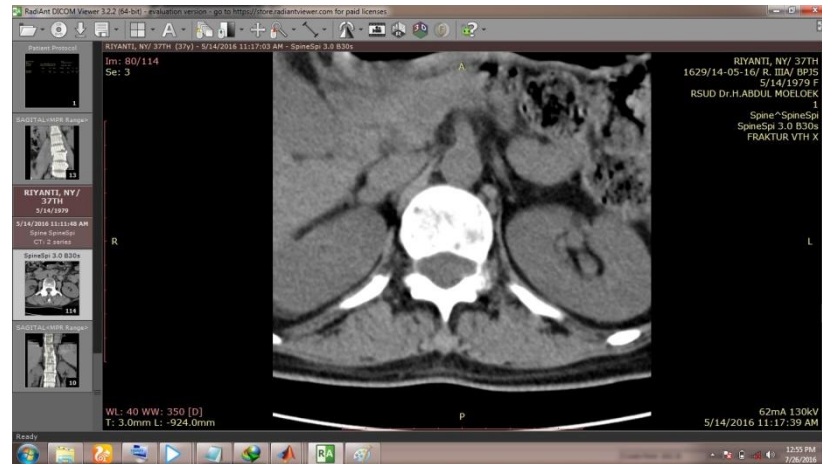
Menurut WHO pada *International Consensus Development Conference*, di Roma, Itali, 1992 *Osteoporosis* adalah penyakit dengan sifat-sifat khas berupa massa tulang yang rendah, disertai perubahan mikroarsitektur tulang, dan penurunan kualitas jaringan tulang, yang pada akhirnya menimbulkan akibat meningkatnya kerapuhan tulang dengan risiko terjadinya patah tulang (Suryati, 2006).

Tulang adalah jaringan yang hidup dan terus bertumbuh. Tulang mempunyai struktur, pertumbuhan dan fungsi yang unik. Bukan hanya memberi kekuatan dan membuat kerangka tubuh menjadi stabil, tulang juga terus mengalami perubahan karena berbagai stres mekanik dan terus mengalami pembongkaran, perbaikan dan pergantian sel. Untuk mempertahankan kekuatannya, tulang terus menerus mengalami proses penghancuran dan pembentukan kembali. Tulang yang sudah tua akan dirusak dan digantikan oleh tulang yang baru dan kuat. Proses ini merupakan peremajaan tulang yang akan mengalami kemunduran ketika usia semakin tua.

Pembentukan tulang paling cepat terjadi pada usia akil balig atau pubertas, ketika tulang menjadi makin besar, makin panjang, makin tebal, dan makin padat yang akan mencapai puncaknya pada usia sekitar 25-30 tahun. Berkurangnya massa tulang mulai terjadi setelah usia 30 tahun, yang akan makin bertambah setelah diatas 40 tahun, dan akan berlangsung terus dengan bertambahnya usia, sepanjang hidupnya. Hal inilah yang mengakibatkan terjadinya penurunan massa tulang yang berakibat pada *osteoporosis* (Tandra, 2009).



Gambar 2.1 Contoh Tulang Positif *Osteoporosis*



Gambar 2.2 Contoh Tulang negatif *Osteoporosis*

2.7.2 Faktor Faktor Penyebab *Osteoporosis*

Ada beberapa faktor yang menjadi penyebab atau faktor-faktor yang berisiko terkena *osteoporosis*, antara lain:

1. Riwayat Keluarga

Seseorang termasuk berisiko tinggi bila orang tuanya juga menderita osteoporosis. Faktor genetik ini terutama berpengaruh pada ukuran dan densitas tulang.

2. Jenis Kelamin

Osteoporosis lebih banyak terjadi pada wanita. Hal ini disebabkan pengaruh hormon estrogen yang mulai menurun kadarnya dalam tubuh sejak usia 35 tahun.

3. Usia

Kehilangan masa tulang meningkat seiring dengan meningkatnya usia. Semakin bertambah usia, semakin besar risiko mengalami osteoporosis karena tulang menjadi berkurang kekuatan dan kepadatannya. Berkurangnya massa tulang mulai terjadi setelah usia antara 30 sampai 35 tahun.

4. Aktifitas Fisik

Kurang kegiatan fisik menyebabkan sekresi Ca yang tinggi dan pembentukan tulang tidak maksimum. Namun aktifitas fisik yang terlalu berat pada usia menjelang menopause justru dapat menyebabkan penyusutan tulang.

2.7.3 Macam Macam *Osteoporosis*

Menurut Tjahjadi, (2002) klasifikasi osteoporosis adalah:

1. *Osteoporosis pascamenopause*

Osteoporosis jenis ini terjadi karena kurangnya hormon estrogen (hormon utama pada wanita), yang membantu mengatur pengangkutan kalsium kedalam tulang dan menekan aktivitas *osteoklas*. Biasanya gejala timbul pada perempuan yang berusia antara 51-75 tahun, tetapi dapat muncul lebih cepat atau lebih lambat. *Hormon estrogen* produksinya mulai menurun 2-3 tahun sebelum *menopause* dan terus berlangsung 3-4 tahun setelah *menopause*. Hal ini berakibat menurunnya massa tulang sebanyak 1-3% dalam waktu 5-7 tahun pertama setelah *menopause*.

2. *Osteoporosis senilis*

Osteoporosis senilis kemungkinan merupakan akibat dari kekurangan kalsium yang berhubungan dengan usia dan ketidak seimbangan antara kecepatan hancurnya tulang (*osteoklas*) dan pembentukan tulang baru (*osteoblas*). *Senilis* berarti bahwa keadaan ini hanya terjadi pada usia lanjut. Penyakit ini biasanya terjadi pada orang-orang berusia diatas 70 tahun.

3. *Osteoporosis* sekunder

adalah *osteoporosis* yang disebabkan oleh keadaan medis lain atau obat-obatan. Penyakit ini bisa disebabkan oleh gagal ginjal kronis dan kelainan hormonal (terutama *tiroid*, *paratiroid*, dan *adrenal*) serta obat-obatan (misalnya *kortikosteroid*, *barbiturat*, antikejang, dan hormon *tiroid* yang berlebihan). Pemakaian alkohol yang berlebihan dan merokok juga dapat memperburuk keadaan ini.

4. *Osteoporosis juvenil idiopati*

merupakan jenis *osteoporosis* yang penyebabnya tidak diketahui. Hal ini terjadi pada anak-anak dan dewasa muda yang memiliki kadar dan fungsi hormon yang normal, kadar vitamin yang normal, dan tidak memiliki penyebab yang jelas dari rapuhnya tulang.

2.8 Radiologi

Radiologi merupakan suatu ilmu tentang penggunaan sumber sinar pengion dan non-pengion, yang biasa digunakan adalah sinar X. Radiologi adalah cabang atau spesialisasi kedokteran yang berhubungan dengan studi dan penerapan berbagai teknologi pencitraan untuk mendiagnosis dan mengobati penyakit. Pencitraan dapat menggunakan sinar-X, USG, *CT scan*, *tomografi emisi positron* (PET) dan MRI.

Pemeriksaan radiologi adalah cara-cara pemeriksaan yang menghasilkan gambar bagian dalam tubuh manusia untuk tujuan diagnostik yang dinamakan pencitraan diagnostik. Menurut Patel (2005 : 2), radiologi merupakan ilmu kedokteran yang digunakan untuk melihat bagian tubuh manusia yang menggunakan pancaran atau radiasi gelombang *elektromagnetik* maupun gelombang mekanik. Modalitas pencitraan (*modality*) merupakan istilah dari alat-alat yang digunakan dalam bidang radiologi untuk melakukan diagnosa terhadap penyakit. Pemeriksaan radiologi memungkinkan suatu penyakit terdeteksi pada tahap awal sehingga akan meningkatkan keberhasilan pengobatan yang dilakukan. Jenis pemeriksaan ini dilakukan dengan menggunakan peralatan pencitraan diagnostik yang perkembangannya sangat dipengaruhi oleh kemajuan ilmu fisika, kimia, dan biologi serta teknologi elektronika, dan komputer.

Tugas pokok radiologi adalah untuk menghasilkan gambar dan laporan temuan pemeriksaan untuk keperluan diagnosis, yang bersama-sama dengan teknik dan temuan diagnostik lainnya akan menjadi dasar tindakan perawatan pasien. Meskipun radiologi merupakan komponen utama dari diagnosis, namun radiologi tidak terbatas hanya untuk keperluan pencitraan diagnostik. Radiologi juga

berperan dalam terapi intervensi seperti *biopsi*, dan pengobatan lainnya, seperti aplikasi pembuluh darah termasuk *recanalization* (menghilangkan penyumbatan) atau *lysis* (pengurangan simptom suatu penyakit akut secara bertahap (*gradually*))

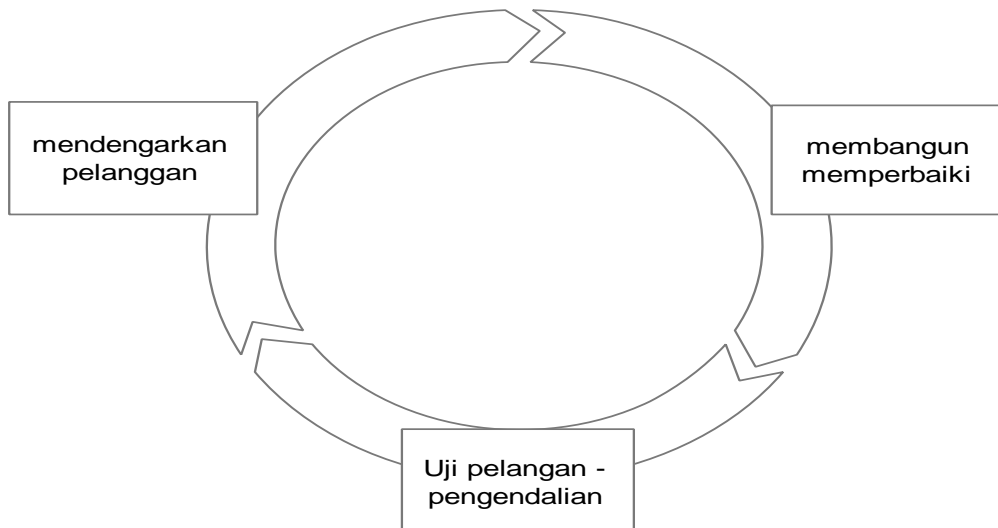
Berikut ini beberapa cabang ilmu dari radiologi:

1. Radiodiagnostik : merupakan salah satu cabang ilmu radiologi yang memanfaatkan sinar pengion untuk membantu diagnosa dalam bentuk foto yang didokumentasikan.
2. Radioterapi : merupakan salah satu terapi penyakit terutama untuk penyakit seperti tumor yang mengalami keganasan dengan sinar radioaktif.
3. Kedokteran Nuklir : merupakan bidang kedokteran yang memanfaatkan materi radioaktif untuk menegaskan diagnosa dan mengobati penderita serta mempelajari penyakit manusia.
4. USG : ultrasonografi merupakan penggunaan gelombang suara frekuensi tinggi untuk membantu diagnosa.
5. MRI : merupakan teknik diagnosa yang memanfaatkan medan magnet dan gelombang frekuensi radio. Pemeriksaan ini tidak menimbulkan radiasi ionisasi, dan dapat diperoleh hasil berupa penampang dari berbagai arah.

2.9 MATLAB

Tim Wahana Komputer (2013, p.21) *MATLAB* merupakan sebuah lingkungan komputasi numerikal dan bahasa pemrograman komputer generasi keempat. Dikembangkan oleh *The MathWorrrks*, *MATLAB* memungkinkan manipulasi matriks, pemplotan fungsi dan data, implementasi algoritma, pembuatan antarmuka pengguna, dan pengantarmukaan dengan program dalam bahasa lainnya. Tipe data dasar adalah matriks (*array*). *Matlab* tidak membutuhkan dimensi, oleh karena itu penggunaan memori dapat dihemat.

input dan format *output*). Perancangan kilat membawa kepada kontruksi sebuah *prototype*. *Prototype* tersebut dievaluasi oleh pelanggan dan dipakai untuk kebutuhan pengembangan perangkat lunak.



Gambar 2.5 *Prototype*

Prototype juga dapat didefinisikan sebagai proses pengembangan suatu *prototype* secara cepat untuk digunakan terlebih dahulu dan ditingkatkan terus menerus sampai didapatkan *system* yang utuh. Merupakan proses yang digunakan untuk membantu pengembang perangkat lunak dalam membentuk *prototype* dari perangkat lunak yang harus dibuat. Proses pada model *prototyping* dapat dijelaskan sebagai berikut :

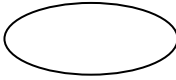
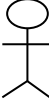
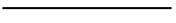
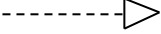
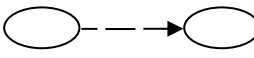
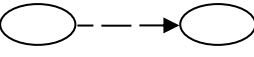
- 1) Pengumpulan kebutuhan : *Developer* dan klien bertemu dan menentukan tujuan umum, kebutuhan yang diketahui dan gambaran bagian - bagian yang akan dibutuhkan berikutnya.
- 2) Perancangan : Perancangan dilakukan cepat dan rancangan mewakili semua aspek perangkat lunak yang diketahui, dan rancangan ini menjadi dasar pembuatan *prototype*.
- 3) Evaluasi *prototype* : Klien mengevaluasi *prototype* yang dibuat dan digunakan untuk memperjelas kebutuhan perangkat lunak.

Perulangan ketiga proses ini terus berlangsung hingga semua kebutuhan terpenuhi. *Prototype - prototype* dibuat untuk memuaskan kebutuhan klien dan untuk membangun perangkat lunak lebih cepat, namun tidak semua *prototype* bisa dimanfaatkan. Demi kebutuhan klien lebih baik *prototype* yang dibuat diusahakan dapat dimanfaatkan.

2.11 Use Case Diagram

Rosa dan Salahudin (2011, p.131) menguraikan bahwa *use case* diagram merupakan pemodelan untuk kelakuan sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi - fungsi itu. Simbol - simbol yang digunakan untuk pembuatan *use case* diagram dapat dilihat pada tabel 2.3 berikut:



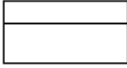




Tabel 2.3 Simbol *Use Case Diagram*

| Simbol | Keterangan |
|--|---|
| <i>Use Case</i>  | Menggambarkan bagaimana seseorang akan menggunakan atau memanfaatkan sistem. |
| Aktor  | Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri. |
| Asosiasi  | Komunikasi antara <i>use case</i> dan aktor yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor. |
| Generalisasi  | Sebagai penghubung antara aktor- <i>use case</i> atau <i>use case-use case</i> . |
| <<Include>>  | <i>Include Relationship</i> (relasi cakupan) : Memungkinkan suatu <i>use case</i> untuk menggunakan fungsionalitas yang disediakan oleh <i>use case</i> yang lainnya. |
| <<Extend>>  | <i>Extend Relationship</i> : Memungkinkan relasi <i>use case</i> memiliki kemungkinan untuk memperluas fungsionalitas yang disediakan oleh <i>use case</i> yang lainnya. |

2.12 Class Diagram

Class diagram adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class diagram* menggambarkan struktur dan deskripsi *class*, *package* beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain. Simbol - simbol yang digunakan untuk pembuatan *class diagram* dapat dilihat pada tabel 2.4 berikut:

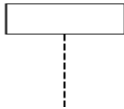


Tabel 2.4 Simbol *Class Diagram*

| SIMBOL | NAMA | KETERANGAN |
|---|-------------------------|---|
|  | <i>Generalization</i> | Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>). |
|  | <i>Nary Association</i> | Upaya untuk menghindari asosiasi dengan lebih dari 2 objek. |
|  | <i>Class</i> | Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama. |
|  | <i>Collaboration</i> | Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor |
|  | <i>Realization</i> | Operasi yang benar-benar dilakukan oleh suatu objek. |
|  | <i>Dependency</i> | Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempegaruhi elemen yang bergantung padanya elemen yang tidak mandiri |
|  | <i>Association</i> | Apa yang menghubungkan antara objek satu dengan objek lainnya |

2.13 Sequence Diagram

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa pesan yang digambarkan terhadap waktu. *Sequence diagram* terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). Simbol-simbol yang digunakan untuk pembuatan *class diagram* dapat dilihat pada tabel 2.5 berikut :



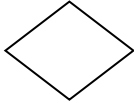


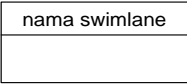
Tabel 2.5 Simbol *Sequence Diagram*

| GAMBAR | NAMA | KETERANGAN |
|---|-----------------|--|
|  | <i>LifeLine</i> | Objek <i>entity</i> , antarmuka yang saling berinteraksi. |
|  | <i>Message</i> | Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi |
|  | <i>Message</i> | Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas. |

2.14 Activity Diagram

Activity Diagram menggambarkan aliran kerja atau aktivitas dari sebuah sistem atau proses bisnis bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh *system*. (Rosa A.S dan Salahudin (2011, p.134). Simbol-simbol yang digunakan untuk pembuatan *activity diagram* dapat dilihat pada tabel 2.6 di bawah ini :

Tabel 2.6 Simbol *Activity Diagram*

| Simbol | Keterangan |
|--|--|
| Status Awal  | Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal. |
| Aktivitas  | Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja. |
| Percabangan  | Asosiasi percabangan dimana ada pilihan aktivitas lebih dari satu. |
| Penggabungan  | Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu. |
| Status Akhir  | Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir. |
| <i>Swimlane</i>  | Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas. |