BAB 2
LANDASAN TEORI

2.1 Gitar

Gitar merupakan salah satu jenis alat musik petik. Gitar digolongkan dalam tiga jenis, yakni gitar akustik, gitar elektrik dan gabungan keduanya (semi akustik). Suara dari masing-masing jenis tentu saja memiliki kekhasan tersendiri. Suara gitar merupakan salah satu bentuk gelombang yang dihasilkan dari getaran senar secara periodik. Gitar terbentuk atas sebuah bagian tubuh pokok dengan bagian leher yang padat sebagai tempat senar yang umumnya berjumlah enam didempetkan. Gitar secara tradisional dibentuk dari berbagai jenis kayu dengan senar yang terbuat dari nilon maupun baja dan beberapa gitar modern dibuat dari material polikarbonat [4].

2.2 Audio

Audio adalah suara atau bunyi yang dihasilkan ketika molekul diudara berubah oleh suatu getaran yang ditimbulkan sebuah objek yang menghasilkan sebuah getara. Objek ini bias berupa senar gitar, vokal manusia dan lain-lain. Frekuensi maksimum yang dihasilkan oleh audio adalah 20KHz dengan frekuensi *sampling* 40KHz. Sedangkan pada kenyataannya manusia hanya dapat mendengar frekuensi tinggi yang mendekati antara 15KHz sampai 17KHz [5].

Waveform digital sample yang disimpan dalam sebuah *pc* dengan format audio *wave. Wave* adalah standar format audio yang dikembangkan oleh Microsoft dan IBM dengan ekstensi *file* (*.wav) yang mempunyai atribut seperti durasi waktu, *bit dept* dan *sample rate.*

a. Durasi Waktu

   Durasi waktu adalah panjang data suara yang terkonversi kedalam format digital audio [5].

b. *Bit Depth*

Bit Depth adalah nama lain dari *sampling resolution. Bit Depth* merupakan banyaknya jumlah digit bilangan biner yang digunakan oleh sampel suara. Sampel dengan kedalaman 8 bit menggunakan sedikit memori atau *harddisk space* dari pada sampel dengan kedalaman 16 bit. Semakin kecil jumlah bit suatu sampel maka semakin sedikit jumlah data yang tersimpan. Pengaruh dari bit resolusi adalah saat sinyal kuantitas pada proses *analog to digital converter* dimana semakin besar *bit depth* maka *error* kuantitasi semakin kecil [5].

c. *Sample rate*

Gelombang dengan laju pencuplikan 44100Hz mempunyai ketepatan suara yang lebih dari 22500Hz. Laju pencuplikan sebesar 44100Hz adalah laju yang digunakan oleh radio FM (*Frequency Modulation).* Semakin besar laju sebuah *file wave,* semakin besar pula kapasitas memori atau *harddisk space* yang digunakan oleh *file* tersebut [5].

2.3 Suara

Suara merupakan fenomena fisik yang dihasilkan oleh getaran suatu benda selama bergetar, dimana perbedaan tekanann terjadi diudara dan disekitarnya. Getaran tersebut berupa sinyal *analog* dengan amplitude yang berubah secara kontinyu terhadap waktu. Pola osilasi yang terjadi dinamakan sebagai gelombang. Gelombang mempunyai pola sama yang berulang pada interval tertentu, yang disebut dengan periode. Contoh suara periodik adalah instrumen musik, nyanyian buruh dan lain-lain, sedangkan contoh suara yang non periodik adalah batuk, percikan ombak dan lain-lain. Suara berkaitan erat dengan :

a. Frekuensi

Frekuensi merupakan banyaknya periode dalam 1 detik. Satuan dari frekuensi adalah Hertz atau *cycles persecond.* Panjang gelombang suara dirumuskan = c/f, dimana c adalah cepat rambat bunyi dan f adalah

frekuensi. Manusia dapat membuat suara dengan frekuensi 50Hz-10KHz, sedangkan sinyal suara musik memiliki frekuensi 20Hz-20KHz. Suara yang berada pada range pendengaran manusia disebut sebagai audio dan gelombangnya disebut sebagai *acoustic signals*. Suara diluar range pendengaran manusia dapat dikatakan sebagai derau. Setiap gitar rata-rata memiliki setelan nada yang berbeda. Hal ini bergantung kepada penggunaan gitar serta keinginan oleh pengguna gitar itu. Nada yang paling umum digunakan dapat dilihat pada Table 2.1[5]:

Tabel 2.1 Frekuensi Senar Gitar

	0	1	2	3	4	5	6	7	8
C	16.35	32.70	65.41	130.81	261.63	523.25	1046.50	2093.00	4186.01
C#	17.32	34.65	69.30	138.59	277.18	554.37	1108.73	2217.46	4434.92
D	18.35	36.71	73.42	146.83	293.66	587.33	1174.66	2349.32	4698.64
D#	19.45	38.89	77.78	155.56	311.13	622.25	1244.51	2489.02	4978.03
E	20.60	41.20	82.41	164.81	329.63	659.26	1318.51	2637.02	5274.04
F	21.83	43.65	87.31	174.61	349.23	698.46	1396.91	2793.83	5587.65
F#	23.12	46.25	92.50	185.00	369.99	739.99	1479.98	2959.96	5919.91
G	24.50	49.00	98.00	196.00	392.00	783.99	1567.98	3135.96	6271.93
G#	25.96	51.91	103.83	207.65	415.30	830.61	1661.22	3322.44	6644.88
A	27.50	55.00	110.00	220.00	440.00	880.00	1760.00	3520.00	7040.00
A#	29.14	58.27	116.54	233.08	466.16	932.33	1864.66	3729.31	7458.62
B	30.87	61.74	123.47	246.94	493.88	987.77	1975.53	3951.07	7902.13

b. Amplitude

Amplitude merupakan keras lemahnya bunyi atau tinggu rendahnya gelombang. Amplitude dinyatakan dalam *decibel (dB).* Bunyi mulai dapat merusak telinga manusia jika tingkat volumenya lebih besar dari 85 *dB* dan pada ukuran 130 *dB* akan membuat gendang telinga hancur.

2.4 Pre Processing

2.4.1 Sampling

Sampling adalah proses mengambil nilai-nilai sinyal pada titik-titik diskrit sepanjang variabel waktu dari sinyal kontinyu, sehingga didapatkan sinyal waktu diskrit. Jumlah titik-titik yang diambil setiap detik dinamakan *sampling rate.* Dalam melakukan *sampling,* perlu diperhatikan kriteria *nyquist* yang menyatakan bahwa sebuah sinyal harus memiliki *sampling rate* yang lebih besar dari $2f_m$ dengan f_m adalah frekuensi paling tinggi yang muncul disebuah sinyal[6].

2.4.2 Framing

Framing adalah pembagian sinyal audio menjadi beberapa frame yang nantinya dapat memudahkan dalam perhitungan dan analisa sinyal, satu frame terdiri dari beberapa sampel tergantung tiap berapa detik suara akan disampel dan berapa besar frekuensi samplingnya. Pada proses ini dilakukan pemotongan sinyal dalam slot-slot tertentu agar memenuhi syarat linier dan time invariant[6].

2.4.3 Overlapping

Overlapping adalah proses yang diawalai dengan membandingkan panjang data yang akan dioverlapping dari maisng-masing *frame* dari beberapa *frame* yang akan digabungkan. Setelah itu akan dibandingkan dan dicari mana yang paling pendek dari keduanya akan dijadikan patokan dalam penentuan banyak data yang akan dioverlapping[6].

2.4.4 Windowing

Dalam melakukan pemrosesan sinyal, maka dari input yag dimasukkan akan terbentuk sinyal yang amplitudonya bervariasi pada awal maupun akhir *frame.* Hal tersebut dapat menghambat pemrosesan sinyal dan menghasilkan keluaran yang kurang akurat. Untuk itu perlu diaplikasikan suatu *window* penghalus pada setiap *frame* dengan melakukan *overlapping* antara satu *frame* dengan *frame* yang lain, sehingga dapat dibangkitkan suatu *feature* yang lebih halus sepanjang durasi waktu tersebut. *Windowing* terdiri dari beberapa macam

yaitu *hamming, hanning, barlet, rectangular* dan *blcakman*. Dalam penelitian ini akan menggunakan *window hamming*[7]. Karena *window hamming* memiliki *side lobe* yang paling kecil dan *main lobe* yang paling besar sehingga hasil *windowing* akan lebih halus dalam menghilangkan efek diskontinyuitas dengan persamaan sebagai berikut :

$$w(n) = 0.54 - 0.46\ .\ cos\left(\frac{2.\pi.n}{N-1}\right) \quad \text{(2.1)}$$

Dimana :

W(n) adalah *windowing*

N merupakan jumlah data dari sinyal

N merupakan waktu diskrit ke

2.5 Wavelet

Gelombang (*wave*) didefinisikan sebagai fungsi dari waktu atau frekuensi yang berisolasi. *Wavelet* adalah suatu gelombang kecil yang mempunyai energy yang dikonsentrasikan dalam waktu yang dapat memudahkan untuk analisis transien, nonstationer dan fenomena *time varying. Wavelet* mempunyai kemampuan untuk menganalisis frekuensi dan waktu yang simultan dengan fungsi-fungsi matematika yang fleksibel.

Wavelet merupakan suatu fungsi yang memenuhi persyaratan matematika tertentu yang digunakan untuk mempresentasikan data atau fungsi lain. Ide yang melandasi *wavelet* adalah untuk menganalisis fungsi menurut skala. Fungsi *wavelet* memotong-motong data menjadi komponen-komponen frekuensi yang berbeda sehingga komponen-komponen tersebut dapat dianalis dengan menggunakan skala resolusi yang sesuai. Tujuan dari *wavelet* untuk mempresentasikan data dilakukan dengan menggunakan pendekatan superposisi dari suatu fungsi.

Ciri dari fungsi *wavelet* yaitu sifatnya yang terlokalisir dalam ruang. Hal ini menjadikan sejumlah fungsi yang menggunakan menjadi tersebar saat

ditransformasikan ke domain *wavelet.* Penyebaran ini menghasilkan sejumlah aplikasi-aplikasi yang berguna seperti kompresi, kemampuan deteksi pada sinyal dan menghilangkan derau. Ciri yang lain adalah seperti untuk memisahkan diskontinuitas sinyal diperlukan beberapa fungsi dasar yang singkat dan berfrekuensi tinggi. Sedangkan untuk mendapatkan analisis frekuensi secara mendetil, diperlukan beberapa fungsi dasar yang panjang dan berfrekuensi rendah. Variasi yang tajam pada sinyal tepi dan diskontinuitas merupakan ciri-ciri atau keistimewaan yang mempunyai banyak informasi yang dianggap cukup untuk mempresentasikan sinyal tersebut. Sifat-sifat dinamik dan variasi-variasi tajam pada sinyal dengan skala atau resolusi yang bervariasi dipercaya mempunyai banyak informasi yang berguna.[8].

2.5.1 Transformasi Wavelet Daubechies dB4

Daubechies gelombang singkat dinamakan oleh Ingrid Daubechies seorang professor dari departemen matematika dan penggunaan matematika di Universitas Princeton. Ingrid sangat ahli dengan pekerjaannya dengan gelombang singkat didalam *image compression*. *Daubechies* adalah keluarga dari gelombang singkat orthogonal didefinisikan dengan *discrete* gelombang singkat *transform* dan dikarakteristikan dengan *vanishing moment* jumlah maksimal dari beberapa pemberi dukungan dengan tiap *ripe* gelombang singkat disetiap *class*, ada sebuah fungsi *scaling* (disebut juga *father* gelombang singkat) yang menghasilkan analisis multiresolusi orthogonal.

Alihragam gelombang singkat *daubechies* didefinisikan hampir sama dengan alihragam singkat *haar* dengan mengkomputasi rata-rata yang berjalan dan perbedaan menggunakan skalar produk dengan sinyal *scaling* dan gelombang singkat satu-satunya perbedaan diantara mereka terdiri dari bagaimana sinyal *scaling* ini dan gelombang singkat didefinisikan.

Tipe gelombang singkat ini mempunyai respon frekuensi yang seimbang tapi *fase* responnya non-linier. *Daubechies* gelombang singkat digunakan bertumpuk dengan *windows*, jadi tingginya reflek koefisien spektrum semuanya mempunyai perubahan frekuensi yang tinggi. Gelombang singkat *daubechies* juga

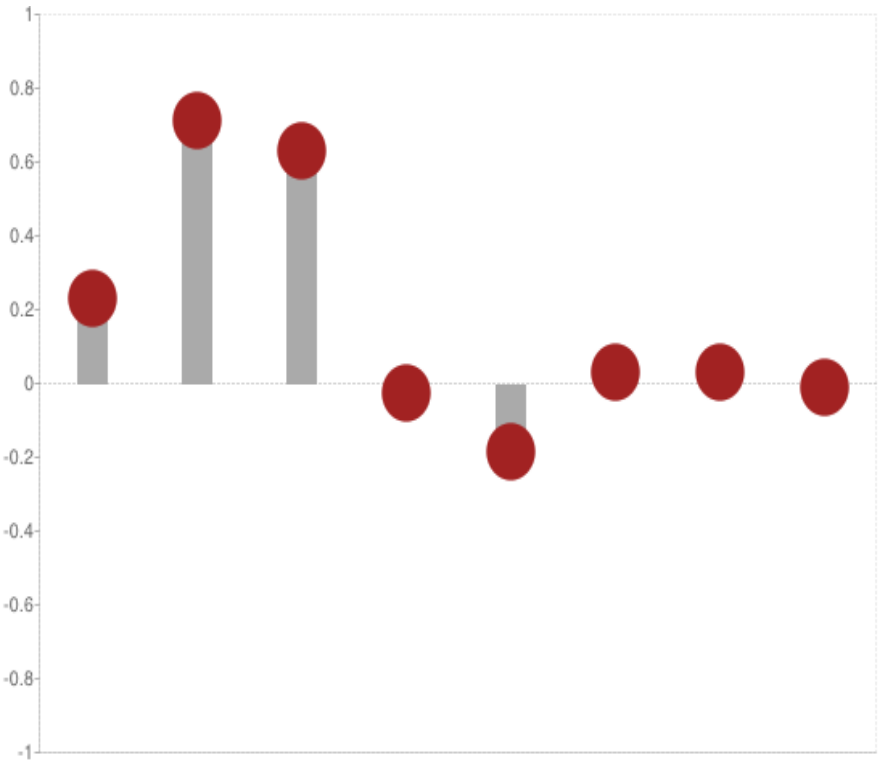
berguna untuk mengkompresi dan menghilangkan *noise* dalam proses sinyal audio.

Umumnya *daubechies* gelombang singkat dipilih karena mempunyai nilai A tertinggi saat menghilang, untuk memberikan dukungan *width* N=2A, dan diantara 2^{A-1} solusi yang mungkin salah satunya memilih filter *scaling* yang mempunyai *fase* eksternal. Alihragam gelombang singkat juga sangat mudah ditempatkan untuk *training* menggunakan alihragam gelombang singkat. Gelombang singkat *daubechies* digunakan secara luas dalam memecahkan berbagai masalah, contohnya *fractal problem* sinyal, sinyal diskontinyus dan lain-lain, berikut adalah koefisien filter skala dan waveletnya [9] :

$$g_0 = \frac{1-\sqrt{3}}{4\sqrt{2}}, {}_1 = \frac{3-\sqrt{3}}{4\sqrt{2}}, g_2 = \frac{3+\sqrt{3}}{4\sqrt{2}}, g_3 = \frac{1+\sqrt{3}}{4\sqrt{2}} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (2.2)$$

Sedangkan untuk koefisien filter wavelet adalah sebagai berikut :

$$h_0 = \frac{-1-\sqrt{3}}{4\sqrt{2}}, h_1 = \frac{3+\sqrt{3}}{4\sqrt{2}}, h_2 = \frac{-3+\sqrt{3}}{4\sqrt{2}}, h_3 = \frac{1-\sqrt{3}}{4\sqrt{2}} \quad \ldots\ldots\ldots\ldots\ldots\ldots \quad (2.3)$$



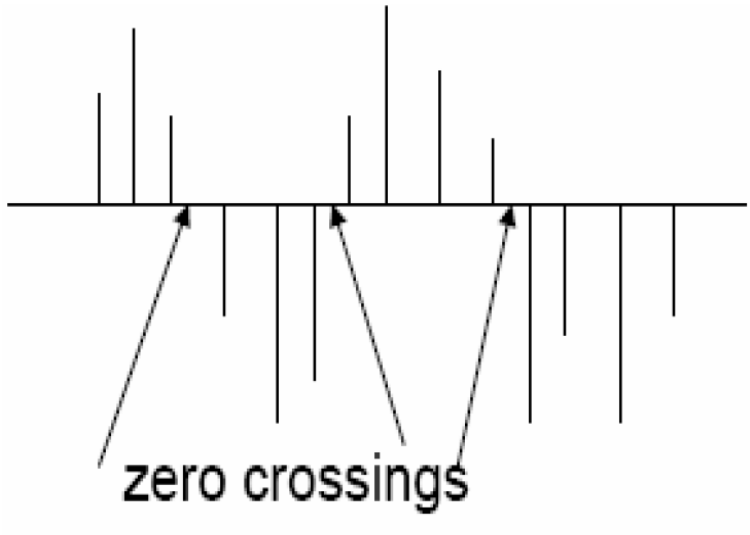
***Gambar 2.1 Grafik rekonstruksi* lowpass filter**

***Tabel 2.2 Koefisien rekonstruksi* lowpass filter**

dB4
0.2303778133088966
0.7148465705529161
0.6308807679298592
-0.0279837694168604
-0.1870348117190935
0.0308413818355607
0.0328830116668852
-0.0105974017850690

2.6 Zero Crossing Rate (ZCR)

Zero crossing rate adalah parameter penting dalam klasifikasi suara yang terucap dan suara yang tidak terucap. Hal ini juga sering digunakan sebagian *front-end processing* didalam sistem pengenalan suara otomatis. Nilai *zero crossing rate* adalah indikator dari frekuensi dimana energi sinyal terkonsentrasi didalam spektrum. Suara pembicara dihasilkan karena eksistasi dari suara periodik yang terbawa oleh udara dicelah suara dan biasanya menunjukkan nilai *zero crossing rate* yang rendah. Sedangkan suara yang tak terucap dihasilkan karena penyempitan saluran suara yang cukup untuk menyebabkan terjadinya aliran udara yang akhirnya menghasilkan *noise* dan menunjukkan nilai *zero crossing rate* yang tinggi[10].



Gambar 2.2 Definisi Zero Crossing Rate

Definisi untuk *zero crossing rate* adalah :

$$Z_t = \frac{1}{2N}\sum_{n=1}^{N} |sign(x[n]) - sign(x[n-1])| \quad \ldots\ldots \quad (2.4)$$

Dimana :

Z_t adalah Zero Crossing Rate (ZCR)

$sign$ adalah sinyal atau frekuensi

N adalah jumlah index

Model untuk produksi suara menyarankan bahwa kisaran energi suara harus terkonsentrasi dibawah 3KHz karena jatuhnya spektrum diperkenalkan oleh gelombang. Sedangkan suara yang tak terucap, sebagian besar energinya ditemukan pada frekuensi yang lebih tinggi. Karena frekuensi tinggi maka *zero crossing rate* tinggi dan frekuensi rendah menyiratkan *zero crossing rate* rendah ada korelasi kuat antara *zero crossing rate* dan distribusi energi dengan frekuensi.

Koefisien gelombang singkat sinyal audio dalam setiap *subband* mengambil suatu bentuk osilasi gelombang. *Zero crossing rate* didalam

transformasi domain menggambarkan seberapa sering frekuensi sinyal telah berubah dalam periode waktu tertentu didalam sebuah *subband.* Normalisasi *zero crossing rate* didefinisikan sebagai berikut :

$$z = \frac{nilai\ zero\ crossing\ rate}{total\ sample}$$

Dimana :

z adalah normalisasi *zero crossing rate (ZCR)*

2.7 Android

Android adalah sistem operasi yang berbasis Linux untuk telepon seluler seperti telepon pintar dan komputer tablet. Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh berbagai macam peranti bergerak. Awalnya, Google Inc. membeli Android Inc., pendatang baru yang membuat piranti lunak untuk ponsel. Kemudian untuk mengembangkan Android, dibentuklah *Open Handset Alliance*, konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia. Pada saat perilisan perdana android, 5 November 2007, Android bersama *Open Handset Alliance* menyatakan mendukung pengembangan standar terbuka pada perangkat seluler. Di lain pihak, Google merilis kode–kode Android di bawah lisensi Apache, sebuah lisensi perangkat lunak dan standar terbuka perangkat seluler. Di dunia ini terdapat dua jenis distributor sistem operasi Android, pertama yang mendapat dukungan penuh dari Google atau Google Mail Services (GMS) dan kedua adalah yang benar-benar bebas distribusinya tanpa dukungan langsung Google atau dikenal sebagai *Open Handset Distribution* (OHD).

Pada Juli 2005, Google bekerjasama dengan Android Inc., perusahaan yang berada di Palo Alto, California Amerika Serikat. Para pendiri Android Inc. bekerja pada Google, di antaranya Andy Rubin, Rich Miner, Nick Sears, dan Chris White. Saat itu banyak yang menganggap fungsi Android Inc. hanyalah sebagai perangkat lunak pada telepon seluler. Sejak saat itu muncul rumor bahwa Google

hendak memasuki pasar telepon seluler. Di perusahaan Google, tim yang dipimpin Rubin bertugas mengembangkan program perangkat seluler yang didukung oleh kernel Linux. Hal ini menunjukkan indikasi bahwa Google sedang bersiap menghadapi persaingan dalam pasar telepon seluler.

Pada 9 Desember 2008, diumumkan anggota baru yang bergabung dalam program kerja Android ARM Holdings, Atheros Communications, diproduksi oleh Asustek Computer Inc, Garmin Ltd, Softbank, Sony Ericsson, Toshiba Corp, dan Vodafone Group Plc. Seiring pembentukan *Open Handset Alliance*, OHA mengumumkan produk perdana mereka, Android, perangkat bergerak (*mobile*) yang merupakan modifikasi kernel Linux 2.6[11].

2.8 SQLite

SQLite merupakan sebuah system manajemen basis data relasional yang bersifat ACID - compliant dan memiliki ukuran pustaka kode yang relatif kecil, ditulis dalam bahasa C. SQLite merupakan proyek yang bersifat public domain yang dikerjakan oleh D. Richard Hipp. Tidak seperti pada paradigma *client-server* umumnya, Inti SQLite bukanlah sebuah sistem yang mandiri yang berkomunikasi dengan sebuah program, melainkan sebagai bagian integral dari sebuah program secara keseluruhan. Sehingga protokol komunikasi utama yang digunakan adalah melalui pemanggilan API secara langsung melalui bahasa pemrograman. Mekanisme seperti ini tentunya membawa keuntungan karena dapat mereduksi *overhead, latency times*, dan secara keseluruhan lebih sederhana. Seluruh elemen basisdata (definisi data, tabel, indeks, dan data) disimpan sebagai sebuah file. Kesederhanaan dari sisi disain tersebut bisa diraih dengan cara mengunci keseluruhan file basis data pada saat sebuah transaksi dimulai.

`SQLite` merupakan database *Open Source* yang tertanam ke Android. `SQLite` mendukung fitur *database* relasional standar seperti sintaks SQL, transaksi dan pernyataan siap. Selain itu hanya memerlukan sedikit memori pada saat runtime (sekitar 250 KByte).

SQLite mendukung tipe data TEXT (mirip ke String di Java), INTEGER (mirip dengan yang lama di Java) dan REAL (mirip dengan ganda di Java). Semua jenis lain harus dikonversi ke pada bidang ini sebelum menyimpannya dalam database. SQLite sendiri tidak memvalidasi jika jenis ditulis untuk kolom sebenarnya dari jenis didefinisikan, Anda dapat menulis sebuah integer ke dalam kolom string.

2.9 Eclipse

Eclipse adalah sebuah IDE *(Integrated Development Environment)* untuk mengembangkan perangkat lunak dan dapat dijalankan di semua *platform (platform-independent).* Berikut ini adalah sifat dari Eclipse:

a. *Multi-platform:* Target sistem operasi Eclipse adalah *Microsoft Windows, Linux, Solaris, AIX, HP-UX* dan *Mac OS X*.
b. *Multi-language*: Eclipse dikembangkan dengan bahasa pemrograman Java, akan tetapi Eclipse mendukung pengembangan aplikasi berbasis bahasa pemrograman lainnya, seperti C/C++, Cobol, Python, Perl, PHP, dan lain sebagainya.
c. *Multi-role*: Selain sebagai IDE untuk pengembangan aplikasi, Eclipse pun bisa digunakan untuk aktivitas dalam siklus pengembangan perangkat lunak, seperti dokumentasi, test perangkat lunak, pengembangan web, dan lain sebagainya.

Eclipse pada saat ini merupakan salah satu IDE favorit dikarenakan gratis dan open source, yang berarti setiap orang boleh melihat kode pemrograman perangkat lunak ini. Selain itu, kelebihan dari Eclipse yang membuatnya populer adalah kemampuannya untuk dapat dikembangkan oleh pengguna dengan komponen yang dinamakan *plug-in*.

Konsep Eclipse adalah IDE yang terbuka *(open),* mudah diperluas *(extensible)* untuk apa saja, dan tidak untuk sesuatu yang spesifik. Jadi, Eclipse tidak saja untuk mengembangkan program Java, akan tetapi dapat digunakan untuk berbagai macam keperluan, cukup dengan menginstal *plug-in* yang

dibutuhkan. Selain itu, pengembangan secara visual bukan hal yang tidak mungkin oleh Eclipse, *plug-in* UML2 tersedia untuk membuat diagram UML. Dengan menggunakan PDE setiap orang bisa membuat *plug-in* sesuai dengan keinginannya[12].

2.10 Xml

XML (*Extensible Markup Language*) adalah bahasa markup untuk keperluan umum yang disarankan oleh W3C untuk membuat dokumen markup keperluan pertukaran data antar sistem yang beraneka ragam.XML merupakan kelanjutan dari HTML (*HyperText Markup Language*) yang merupakan bahasa standar untuk melacak Internet.

XML didesain untuk mempu menyimpan data secara ringkas dan mudah diatur.Kata kunci utama XML adalah data (jamak dari datum) yang jika diolah bisa memberikan informasi. XML menyediakan suatu cara terstandarisasi namun bisa dimodifikasi untuk menggambarkan isi dari dokumen. Dengan sendirinya, XML dapat digunakan untuk menggambarkan sembarang *view database*, tetapi dengan suatu cara yang standar.

2.10.1 Tipe XML

XML memiliki tiga tipe, diantaranya :

a. XML, merupakan standar format dari struktur berkas (*file*).
b. XSL, merupakan standar untuk memodifikasi data yang diimpor atau diekspor.
c. XSD, merupakan standar yang mendefinisikan struktur *database* dalam XML.

2.10.2 Keunggulan XML

Keunggulan XML bisa diringkas sebagai berikut :

a. Pintar (*Intelligence*). XML dapat menangani berbagai tingkat (*level*) kompleksitas.
b. Dpat beradaptasi. Dapat mengadaptasi untuk membuat bahasa sendiri. Seperti Microsoft membuat bahasa MSXML atau Macromdia mengembangkan MXML.
c. Mudah pemeliharaannya.
d. Sederhana. XML lebih sederhana.
e. Mudah dipindah-pindahkan (*Portability*). XML mempunyai kemudahan perpindahan (*Portabilitas)* yang lebih bagus.

2.11 Unfield Modeling Language (UML)

UML menurut Booch (1999: 14) dalam Jurnal Gintoro, Andreyus, Emilia dan Richard William (2010 , Hal : B-30) adalah sebuah bahasa yang telah menjadi standar dalam industri untuk menvisualisasi, menspesifikasi, merancang dan mendokumentasi sistem piranti lunak[13].

Unified Modelling Language (UML) menawarkan sebuah standar untuk merancang model sebuah sistem. Tujuan UML adalah[11] :

a. Memberikan model yang siap pakai, bahasa pemodelan visual yang ekspresif untuk mengembangkan dan saling menukar model dengan mudah dan dimengerti secara umum.
b. Memberikan bahasa pemodelan yang bebas dari berbagai bahasa pemrogramandan proses rekayasa. Menyatukan praktik-praktik terbaik yang terdapat dalam pemodelan.
c. Mentatukan praktik-praktik terbaik yang terdapat dalam pemodelan.

UML menyediakan beberapa notasi dan *artifact standard* yang bisa digunakan sebagai alat komunikasi bagi para pelaku dalam proses analisis dan desain. *Artifact* didalam UML didefinisikan sebagai informasi dalam bentuk yang digunakan atau dihasilkan dalam proses pengembangan perangkat. Contohnya

adalah *source code* yang dihasilkan oleh proses pemrograman dan terdapat beberapa diagram XML, diantaranya sebagai berikut :

2.11.1 Use Case Diagram

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Use Case dilakukan oleh satu actor yang digambarkan dengan simbol orang yang dihubungkan dengan garis yang menunjukan hubungan komunikasi Setiap Use Case harus diberi nama yang menyatakan apa hal yang dicapai dari hasil interaksi dengan Actor. Nama Use Case boleh terdiri dari beberapa kata dan tidak boleh ada Use Case yang memiliki nama yang sama[13].

2.11.2 Class Diagram

Diagram kelas menggambarkan tipe-tipe objek dalam *system* dan berbagai jenis hubungan atau relasi statis yang ada diantara mereka. Diagram ini memberikan gambaran umum dari *system*. Seperti tipe-tipe dari objek dengan menunjukan kelasnya dan *relationship* yang diantara mereka, serat penjelasan detail tiap-tiap kelas ke dalam model suatu *system*. *Class diagram* bersifat *static* (tidak berubah) yang akan menunjukan apa itu interaksi tapi tidak menjelaskan apa yang terjadi ketika mereka melakukan interaksi[13].

2.11.3 Activity Diagram

Activity diagram memodelkan alur kerja *(work flow)*. Sebuah proses bisnis dan urutan aktifitas dalam suatu proses. Diagram ini sangat mirip dengan sebuah *flowchart* karena kita dapat memodelkan sebuah alur kerja dari aktifitas keaktifitas lainnya atau dari suatu aktifitas kedalam keadaan sesaat *(state)*[13].

2.11.4 Sequence Diagram

Sequence diagram merupakan gambaran interaksi antar objek di dalam dan di sekitar sistem berupa *message* yang digambarkan terhadap waktu. Diagram ini secara khusus berasosiasi dengan *Use Case*. *Sequence Diagram* juga digunakan untuk menggambarkan skenario atau rangkain langkah-langkah apa yang seharusnya terjadi sebagai respons dari sebuah event untuk menghasilkan sesuatu

didalam *Use Case* sebagai *output*. Untuk *message* digambarkan sebagai garis berpanah dari satu objek ke objek lainnya. Pada fase desain berikutnya, *message* dipetakan menjadi operasi/metode dari *class. Activation bar* menunjukan lamanya eksekusi sebuat proses, biasanya diawali dengan sebuah *message*[13].

2.12 Java

Java adalah bahasa pemrograman berorientasi objek yang dikembangkan oleh Sun Microsystems sejak tahun 1991.Bahasa ini dikembangkan dengan model yang mirip dengan bahasa C++ dan Smalltalk, namun dirancang agar lebih mudah dipakai dan *platform independent*, yaitu dapat dijalankan di berbagai jenis sistem operasi dan arsitektur computer[14].

2.12.1 Sejarah Java

Bahasa pemrograman Java terlahir dari The Green Project, yang berjalan selama 18 bulan, dari awal tahun 1991 hingga musim panas 1992. Proyek tersebut belum menggunakan versi yang dinamakan Oak.Proyek ini dimotori oleh Patrick Naughton, Mike Sheridan, James Gosling dan Bill Joy, beserta sembilan pemrogram lainnya dari Sun Microsystems. Salah satu hasil proyek ini adalah maskot Dukeyang dibuat oleh Joe Palrang.

Pertemuan proyek berlangsung di sebuah gedung perkantoran Sand Hill Road di Menlo Park. Sekitar musim panas 1992 proyek ini ditutup dengan menghasilkan sebuah program Java Oak pertama, yang ditujukan sebagai pengendali sebuah peralatan dengan teknologi layar sentuh (*touch screen*), seperti pada PDA sekarang ini. Teknologi baru ini dinamai "*7" (*Star Seven*).

Setelah era *Star Seven* selesai, sebuah anak perusahaan Tv kabel tertarik ditambah beberapa orang dari proyek The Green Project. Mereka memusatkan kegiatannya pada sebuah ruangan kantor di 100 Hamilton Avenue, Palo Alto. Perusahaan baru ini bertambah maju: jumlah karyawan meningkat dalam waktu singkat dari 13 menjadi 70 orang. Pada rentang waktu ini juga ditetapkan pemakaian Internet sebagai medium yang menjembatani kerja dan ide di antara

mereka.Pada awal tahun 1990-an, Internet masih merupakan rintisan, yang dipakai hanya di kalangan akademisi dan militer.

Mereka menjadikan perambah (*browser*) Mosaic sebagai landasan awal untuk membuat perambah Java pertama yang dinamai Web Runner, terinsipirasi dari film 1980-an, Blade Runner. Pada perkembangan rilis pertama, Web Runner berganti nama menjadi Hot Java.

Pada sekitar bulan Maret1995, untuk pertama kali kode sumber Java versi 1.0a2 dibuka. Kesuksesan mereka diikuti dengan untuk pemberitaan pertama kali pada surat kabar *San Jose Mercury News* pada tanggal 23 Mei1995. Sayang terjadi perpecahan di antara mereka suatu hari pada pukul 04.00 di sebuah ruangan hotel Sheraton Palace.Tiga dari pimpinan utama proyek, Eric Schmidt dan George Paolini dari Sun Microsystems bersama Marc Andreessen, membentuk Netscape.

Nama Oak, diambil dari pohon oak yang tumbuh di depan jendela ruangan kerja "Bapak Java", James Gosling. Nama Oak ini tidak dipakai untuk versi release Java karena sebuah perangkat lunak lain sudah terdaftar dengan merek dagang tersebut, sehingga diambil nama penggantinya menjadi "Java". Nama ini diambil dari kopi murni yang digiling langsung dari biji (kopi tubruk) kesukaan Gosling.Konon kopi ini berasal dari Pulau Jawa. Jadi nama bahasa pemrograman Java tidak lain berasal dari kata Jawa (bahasa Inggris untuk Jawa adalah Java)[14].

2.12.2 Versi Awal Java

Versi awal Java ditahun 1996 sudah merupakan versi *release* sehingga dinamakan Java Versi 1.0. Java versi ini menyertakan banyak paket standar awal yang terus dikembangkan pada versi selanjutnya.

a. java.lang: Peruntukan kelas elemen-elemen dasar.
b. java.io: Peruntukan kelas *input* dan *output*, termasuk penggunaan berkas.

c. java.util: Peruntukan kelas pelengkap seperti kelas struktur data dan kelas-kelas penanggalan.
d. java.net: Peruntukan kelas TCP/IP, yang memungkinkan berkomunikasi dengan kompiter lain menggunakan jaringan TCP/IP.
e. java.awt: Kelas dasar untuk aplikasi antarmuka dengan pengguna (GUI).
f. java.applet: Kelas dasar aplikasi antarmuka untuk diterapkan pada penjelajahan web.

2.12.3 Kelebihan Java

Berikut kelebihan-kelebihan Java[14] :

a. *Multiplatform*. Kelebihan utama dari Java ialah dapat dijalankan di beberapa *platform* / sistem operasi komputer, sesuai dengan prinsip tulis sekali, jalankan di mana saja. Dengan kelebihan ini pemrogram cukup menulis sebuah program Java dan dikompilasi (diubah, dari bahasa yang dimengerti manusia menjadi bahasa mesin / *bytecode*) sekali lalu hasilnya dapat dijalankan di atas beberapa platform tanpa perubahan. Kelebihan ini memungkinkan sebuah program berbasis java dikerjakan diatas *operatingsystem* Linux tetapi dijalankan dengan baik di atas Microsoft Windows. *Platform* yang didukung sampai saat ini adalah Microsoft Windows, Linux, Mac OS dan Sun Solaris. Penyebanya adalah setiap sistem operasi menggunakan programnya sendiri-sendiri (yang dapat diunduh dari situs Java) untuk meninterpretasikan *bytecode* tersebut.
b. OOP (*Object Oriented Programming* - Pemrogram Berorientasi Objek).
c. Perpustakaan kelas yang lengkap, Java terkenal dengan kelengkapan *library*/perpustakaan (kumpulan program program yang disertakan dalam pemrograman Java) yang sangat memudahkan dalam penggunaan oleh para pemrogram untuk membangun aplikasinya. Kelengkapan perpustakaan ini ditambah dengan keberadaan komunitas

Java yang besar yang terus menerus membuat perpustakaan-perpustakaan baru untuk melingkupi seluruh kebutuhan pembangunan aplikasi.

d. Bergaya C++, memiliki sintaks seperti bahasa pemrograman C++ sehingga menarik banyak pemrogram C++ untuk pindah ke Java. Saat ini pengguna Java sangat banyak, sebagian besar adalah pemrogram C++ yang pindah ke Java. Universitas-universitas di Amerika Serikat juga mulai berpindah dengan mengajarkan Java kepada murid-murid yang baru karena lebih mudah dipahami oleh murid dan dapat berguna juga bagi mereka yang bukan mengambil jurusan komputer.
e. Pengumpulan sampah otomatis, memiliki fasilitas pengaturan penggunaan memori sehingga para pemrogram tidak perlu melakukan pengaturan memori secara langsung (seperti halnya dalam bahasa C++ yang dipakai secara luas).

2.12.4 Kekurangan Java

Berikut kekurangan-kekurangan Java[14] :

a. Tulis sekali, jalankan di mana saja - Masih ada beberapa hal yang tidak kompatibel antara *platform* satu dengan *platform* lain. Untuk J2SE, misalnya *SWT-AWT bridge* yang sampai sekarang tidak berfungsi pada Mac OS X.
b. Mudah didekompilasi. Dekompilasi adalah proses membalikkan dari kode jadi menjadi kode sumber. Ini dimungkinkan karena kode jadi Java merupakan *bytecode* yang menyimpan banyak atribut bahasa tingkat tinggi, seperti nama-nama kelas, metode, dan tipe data. Hal yang sama juga terjadi pada Microsoft .NET *Platform*. Dengan demikian, algoritma yang digunakan program akan lebih sulit disembunyikan dan mudah dibajak/di-*reverse-engineer*.
c. Penggunaan memori yang banyak. Penggunaan memori untuk program berbasis Java jauh lebih besar daripada bahasa tingkat tinggi generasi sebelumnya seperti C/C++ dan Pascal (lebih spesifik lagi, Delphi dan

Object Pascal). Biasanya ini bukan merupakan masalah bagi pihak yang menggunakan teknologi terbaru (karena trend memori terpasang makin murah), tetapi menjadi masalah bagi mereka yang masih harus berkutat dengan mesin komputer berumur lebih dari 4 tahun.

2.12.5 Tahap Kompilasi Java

Berikut tahap kompilasi Java :

a. Tulis / Ubah. Pemrogram menulis program dan menyimpannya di media dalam bentuk berkas '.java'.
b. Kompilasi, pengkompilasi membentuk *bytecodes* dari program menjadi bentuk berkas '.class'.
c. Muat, pemuat kelas memuat *bytecodes* ke memori.
d. Verifikasi, peng-verifikasi memastikan *bytecodes* tidak mengganggu sistem keamanan Java.
e. Jalankan, penerjemah menerjemahkan *bytecodes* ke bahasa mesin.