BAB 2

TINJAUAN PUSTAKA

2.1. Profil Instansi

Gentra Rineka adalah sebuah sanggar seni yang didalamnya terdapat seni modern dan tradisional untuk dikaji dan dikembangkan dengan metode pembelajaran dalam bentuk kursus. Gentra Rineka didirikan tanggal 24 Maret 1990 beralamatkan di BTN Ciheuleut Indah Blok B.9 No.198 RT.43/14 Kelurahan Paska Kecamatan Subang Kabupaten Subang Jawa Barat.

2.1.1. Logo Instansi

Logo instansi merupakan suatu gambar yang memiliki fungsi dan arti dari sebuah instansi untuk menggambarkan instansi itu sendiri.

Logo Instansi dapat dilihat pada gambar 2.1:



Gambar 2.1 Logo Instansi

2.1.2. Visi dan Misi

2.1.2.1. Visi

"Menjadi lembaga sanggar seni yang terbaik dan berkualitas".

2.1.2.2. Misi

Membentuk seniman dan musisi yang bersikap etis, berwawasan akademis dan kultural, profesional serta berperan dalam masyarakat.

2.1.3. Strukur Organisasi dan *Job Description*

Gentra Rineka memiliki struktur organisasi dan *job description* :

Pelindung : Ir. Moch. Agus Tias Amin (Ketua Dewan Kesenian Subang)

Penasihat : Idit Supardi Madiana (Ketua BKKNI Kabupaten Subang)

Ketua : Pepen Ependi (Pemilik)

Wakil Ketua : Oni Sobani

Sekretaris : Drs. Agus Samsudin

Bendahara : Ikhsan Mutaqin

Pengajar : Deny Irawan (Gitar)

Deden Sopian (Gitar)

Hendry Chuenx (Gitar)

Elis Juitasari (Vokal)

Yosep Abdullah (Gitar dan Keyboard)

Nandang (Tari Tradisional dan Modern)

Ade Intra (Karawitan)

Aep (Karawitan)

2.2. Landasan Teori

Dalam sub bab ini akan dibahas tentang teori-teori yang berhubungan dengan pembuatan Aplikasi Pengenalan dan Pembelajaran Akord Gitar Berbasis Web.

2.2.1. *Sound Recognition*

Sound Recognition adalah teknologi pengenalan suara yang memiliki kemampuan untuk mengontrol sebuah sistem dengan menggunakan sebuah suara. *sound recognition* merupakan teknologi yang mampu menganalisis sebuah masukan berupa sinyal digital ke dalam bentuk besaran frekuensi. Kemampuan mengenal secara spesifik suara individual, mirip dengan pengenalan sidik jari. Teknologi ini memiliki kemiripan dengan teknologi *speech recognition* dengan kemampuan untuk menafsirkan kata-kata yang diucapkan oleh manusia. Dalam pengaplikasiannya dengan menggunakan teknologi ini, kita dapat memberi perintah kepada sistem dalam bentuk suara atau bunyi. Oleh sebab itu dibutuhkan sebuah mikrofon sebagai media untuk menjalankan sistem tersebut.

Secara garis besar, cara kerja *sound recognition* ialah sinyal suara audio yang diterima melalui mikrofon atau *line jack* sehingga menjadi sebuah masukan dalam bentuk sinyal digital melalui soundcard sebuah pc.

Sinyal digital yang masuk pada sistem dilakukan tahap normalisasi dengan menggunakan sebuah algoritma yang dapat merubah besaran spectrum menjadi

besaran frekuensi dalam bentuk Hz. Hal ini bertujuan agar perbedaan antara suara yang satu dengan yang lain terlihat lebih jelas sehingga ekstraksi parameter sinyal memberikan hasil yang lebih baik.[7]

2.2.2. Algoritma *Fast Fourier Transform*

Algoritma *fast fourier transform* merupakan suatu algoritma yang sering dipakai untuk menganalisa spektrum frekuensi sinyal analog ataupun data digital. Selain itu algoritma *fast fourier transform* merupakan salah satu metode untuk transformasi sinyal suara menjadi sinyal frekuensi. [4]

Fast Fourier Transform (FFT) merupakan salah satu dari algoritma *Discrete Fourier Transform* (DFT). DFT merupakan algoritma yang memiliki beban komputasi yang besar. Dengan FFT beban komputasi tersebut dapat dikurangi.

Contoh perhitungan sederhana sebagai perbandingan antara DFT dan FFT dapat dilihat pada tabel 2.1:

Tabel 2.1 Hasil Perhitungan DFT

k	F(k)		k	F(k)
0	12		5	0
1	0		6	-2 + 2j
2	-2 – 2j		7	0
3	0		8	12
4	0			

Untuk menghitung hasil tersebut diperlukan 8x8 = 64 kali perhitungan (*looping*). Bila dihitung dengan menggunakan FFT seperti diatas, maka akan diperoleh (5x8 + 4) = 44 kali perhitungan, hal ini terlihat sangat meng/hemat

perhitungan. Tentunya dengan bertambahnya ukuran data, maka perbedaan kecepatan perhitungannya akan semakin terasa.

Untuk n buah data, DFT memerlukan n^2 kali perhitungan, dan FFT memerlukan (n/2 + 1)n + n/2 kali perhitungan. Misalkan jumlah data n=100 maka dengan menggunakan DFT diperlukan 100x100 = 10.000 kali perhitungan, dengan dengan menggunakan FFT cukup dilakukan (51x100 + 50) = 5150 kali perhitungan.

Untuk mendapatkan sinyal dalam domain frekuensi dari sebuah sinyal discrete, salah satu metode transformasi fourier yang digunakan adalah discrete fourier transform (DFT). DFT dilakukan terhadap masing-masing frame dari sinyal yang telah diubah. Namun, yang menjadi persoalan adalah bahwa DFT tersebut memerlukan waktu komputasi yang sangat panjang untuk data yang besar. Oleh karena itu, diperlukan suatu teknik komputasi yang efisien, baik dari sisi waktu maupun dari sisi penggunaan memori.

FFT adalah algoritma cepat untuk mengimplementasikan *discrete fourier transform* (DFT). FFT ini mengubah masing-masing frame N sampel dari domain waktu menjadi domain frekuensi.

Hasil transformasi ini dipengaruhi oleh beberapa parameter, yaitu sample rate sinyal suara dan FFT size. Sample rate adalah banyaknya sampel input analog yang diambil secara digital dengan satuan Hertz (Hz). Sample rate sinyal suara berpengaruh pada besarnya jangkauan frekuensi dari koefisien hasil FFT. Jangkauan frekuensi hasil FFT adalah setengah dari sample rate sinyal suara yang ditransformasi. Artinya, apabila terdapat sinyal suara dengan sample rate 44100

Hz, maka koefisien-koefisien hasil transformasi dari sinyal suara tersebut berkisar dari 0 Hz sampai 22050 Hz. Jadi, semakin besar sample rate , maka akan semakin detail pula sampel analog yang diambil secara digital.

Sedangkan FFT size adalah panjang dari FFT yang digunakan. FFT size berpengaruh terhadap ketelitian tiap koefisien FFT. Semakin besar FFT size, maka tiap koefisien hasil FFT akan mewakili rentang frekuensi yang semakin kecil, sehingga ketelitiannya semakin tinggi. Sebaliknya apabila ukuran sampel FFT semakin kecil, maka tiap koefisien hasil FFT akan mewakili rentang frekuensi yang semakin besar, sehingga ketelitiannya semakin rendah.

Output dari FFT ditransformasikan ke dalam rentang frekuensi. Nilai magnitude terhadap frekuensi didapatkan berdasarkan persamaan 2.1.

$$F(x+T/2) = Real\{F(T/2-x)\} - j\ Im\{F(x)\} \quad \ldots(2.1)$$

Untuk tiap frame akan dilakukan proses FFT dengan window function Hamming, sehingga menghasilkan N buah data RealOut dan Imaginer. Algoritma FFT dan fungsi window Hamming akan di-handle langsung oleh komponen tersebut. Begitu pula dengan ukuran data tiap frame yang ditentukan sendiri melalui komponen tersebut melalui properties BufferSize. Jadi, nilai BufferSize yaitu 1024 dan nilai N/2 buah data diinisialisaikan pada properties BufferSize2 secara otomatis.

Hasil dari FFT tersebut adalah simetris, sehingga hanya N/2 buah data yang digunakan untuk proses selanjutnya. Dari N/2 buah data tersebut akan dihitung rata-rata nilai magnitude yang menjadi nilai dari frame tersebut. Misalkan sampling rate yang digunakan adalah sebesar 44100Hz, maka frekuensi

tertinggi yang diperoleh adalah 22050Hz dengan pembagian frekuensi sebesar 22050Hz/512 data = 43.07 Hz/data. Nilai frekuensi tersebut sesuai dengan batas frekuensi tertinggi yang dapat didengar oleh manusia. Tabel detail pembagian frekuensi sebagai hasil dari proses frequency extraction dapat dilihat pada tabel 2.2.

Tabel 2.2 Tabel Extraction Frekuensi

Data	Frekuensi
1	**0 Hz – 43.07 Hz**
2	**43.07 – 86.13 Hz**
--	**----------**
512	**22006.93 – 22050 Hz**

Konsep utama algoritma ini adalah mengubah sinyal suara yang berbasis masukan suara menjadi berbasis frekuensi dengan membagi masalah menjadi beberapa masalah yang lebih kecil. Kemudian, setiap masalah diselesaikan dengan cara melakukan pencocokan pola digital suara.

Pada umumnya proses *fast fourier transform* adalah membawa sinyal (gelombang suara) sebagai masukan untuk dibawa ke dalam kawasan algoritma. Selanjutnya setelah proses equalisasi terjadi akan merubah sinyal yang masuk menjadi sebuah frequensi. Sebuah akord sudah memiliki sebuah ketetapan frequensi, sehingga algoritma dalam hal ini tidak ada proses pengenalan sinyal atau masukan suara yang baru untuk dijadikannya sebuah frequensi.

2.2.3. Teori Pendukung Analisis

2.2.3.1. UML

Unified Modeling Language (UML) adalah bahasa spesifikasi standar untuk mendokumentasikan, menspesifikasikan, dan membangun sistem perangkat lunak. *Unified Modeling Language* (UML) adalah himpunan struktur dan teknik untuk pemodelan desain program berorientasi objek (OOP) serta aplikasinya. UML mulai diperkenalkan oleh *Object Management Group*, sebuah organisasi yang telag mengembangkan model, teknologi, dan standar OOP sejak tahun 1980-an. UML saat ini mulai banyak digunakan oleh para praktisi OOP. UML merupakan dasar bagi perangkat (tool) desain berorientasi objek dari IBM.

UML adalah suatu bahasa yang digunakan untuk menentukan, memvisualisasikan, membangun, dan mendokumentasikan suatu sistem informasi. UML dikembangkan sebagai suatu alat untuk analisis dan desain berorientasi objek oleh Grady Booach, Jim Rumbaugh, dan Ivar Jacobson. Namun demikian UML dapat digunakan untuk memahami dan mendokumentasikan setiap sistem informasi. Pengguna UML dalam industri terus meningkat. Ini merupakan standar terbuka yang menjadikannya sebagai bahasa pemodelan yang umum dalam industri perangkat lunak dan pengembangan sistem. [11]

2.2.3.1.1. Sejarah UML

Sampai era tahun 1990 puluhan metodologi pemodelan berorientasi objek telah bermunculan di dunia. Diantaranya adalah: metodologi booch, metodologi coad, metodologi OOSE, metodologi OMT, metodologi shlaer-mellor,

metodologi wirfs-brock dan masih banyak metodologi lainnya. Masa itu terkenal dengan masa perang metodologi (*method war*) dalam pendesainan berorientasi objek. Masing-masing metodologi membawa notasi sendiri-sendiri, yang mengakibatkan timbul masalah baru apabila kita bekerjasama dengan kelompok/perusahaan lain yang menggunakan metodologi yang berlainan.

Dimulai pada bulan Oktober 1994 Booch, Rumbaugh dan Jacobson, yang merupakan tiga tokoh yang boleh dikata metodo*login*ya banyak digunakan mempelopori usaha untuk penyatuan metodologi pendesainan berorientasi objek. Pada tahun 1995 direlease draft pertama dari UML (versi 0.8). Sejak tahun 1996 pengembangan tersebut dikoordinasikan oleh *Object* Management Group (OMG –http://www.omg.org).[11]

2.2.3.1.2. Diagram UML

UML menyediakan 10 macam diagram untuk memodelkan aplikasi berorientasi objek, yaitu:

1. *Use Case Diagram* untuk memodelkan proses bisnis.
2. *Conceptual Diagram* untuk memodelkan konsep-konsep yang ada di dalam aplikasi.
3. *Sequence diagram* untuk memodelkan pengiriman pesan (*message*) antar *objects*.
4. *Collaboration Diagram* untuk memodelkan interaksi antar *objects*.
5. *State Diagram* untuk memodelkan perilaku *objects* di dalam sistem.
6. *Activity Diagram* untuk memodelkan perilaku *Use Cases* dan *objects* di dalam *system*.

7. *Class Diagram* untuk memodelkan struktur kelas.
8. *Object Diagram* untuk memodelkan struktur *object*.
9. *Component Diagram* untuk memodelkan komponen *object*.
10. *Deployment Diagram* untuk memodelkan distribusi aplikasi.

2.2.3.1.3. *Use Case Diagram*

Use case diagram digunakan untuk memodelkan bisnis proses berdasarkan perspektif pengguna sistem. *Use case diagram* terdiri atas diagram untuk *use case* dan *actor*. *Actor* mempresentasikan orang yang akan mengoperasikan atau yang berinteraksi dengan sistem aplikasi.

Use case mempresentasikan operasi-operasi yang dilakukan oleh *actor*. *Use case* digambarkan berbentuk elips dengan nama operasi dituliskan di dalamnya. *Actor* yang melakukan operasi dihubungkan dengan garis lurus ke *use case*.[11]

2.2.3.1.4. *Sequence diagram*

Sequence diagram menjelaskan secara detil urutan proses yang dilakukan dalam sistem untuk mencapai tujuan dari *use case*. Interaksi yang terjadi antar *class*, operasi apa saja yang terlibat, urutan antar operasi, dan informasi yang diperlukan oleh masing-masing operasi.[11]

2.2.3.1.5. *Collaboration Diagram*

Collaboration diagram dipakai untuk memodelkan interaksi antar *object* di dalam sistem. Berbeda dengan *sequence diagram* yang lebih menonjolkan kronologis dari operasi-operasi yang dilakukan, collaboration

diagram lebih fokus pada pemahaman atas keseluruhan operasi yang dilakukan oleh *object*. [11]

2.2.3.1.6. *Class Diagram*

Class diagram merupakan diagram yang selalu ada di permodelan sistem berorientasi objek. Class diagram menunjukkan hubungan antar *class* dalam sistem yang sedang dibangun dan bagaimana mereka saling berkolaborasi untuk mencapai suatu tujuan. [11]

2.2.4. Teknologi *Java*

2.2.4.1. *Java*

Java adalah bahasa pemrograman yang dapat dijalankan diberbagai komputer termasuk telepon genggam. Dikembangkan oleh *Sun Microsystem* dan diterbitkan tahun 1995. *Java* tidak boleh disalahpahami sebagai *Java*Script. *Java*Script adalah bahasa *scripting* yang digunakan oleh *web browser*.[11]

2.2.4.2. Sejarah Perkembangan *Java*

Bahasa pemrograman *Java* pertama lahir 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 *Duke* yang 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 Maret 1995, 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 Mei 1995.

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 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*).[11]

2.2.4.3. Versi Awal

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:

1. *java*.lang: Peruntukan kelas elemen-elemen dasar.
2. *java*.io: Peruntukan kelas *input* dan *output*, termasuk penggunaan berkas.
3. *java*.util: Peruntukan kelas pelengkap seperti kelas struktur data dan kelas kelas penanggalan.
4. *java*.net: Peruntukan kelas TCP/IP, yang memungkinkan berkomunikasi dengan komputer lain menggunakan jaringan TCP/IP.
5. *java*.awt: Kelas dasar untuk aplikasi antarmuka dengan pengguna (GUI)
6. *java*.applet: Kelas dasar aplikasi antar muka untuk diterapkan pada penjelajah *web*.[11]

2.2.4.4. Kelebihan *Java*

Java memiliki beberapa kelebihan diantaranya adalah:

1. *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 operating system 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.
2. OOP (*Object Oriented Programming* - Pemrogram Berorientasi Objek) yang artinya semua aspek yang terdapat di *Java* adalah Objek. *Java* merupakan salah satu bahasa pemrograman berbasis objek secara murni. Semua tipe data diturunkan dari kelas dasar yang disebut *Object*. Hal ini sangat memudahkan pemrogram untuk mendesain, membuat, mengembangkan dan mengalokasi kesalahan sebuah program dengan basis *Java* secara cepat, tepat, mudah dan terorganisir. Kelebihan ini menjadikan *Java* sebagai salah satu bahasa pemograman termudah,

bahkan untuk fungsi fungsi yang advance seperti komunikasi antara komputer sekalipun.

3. 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.
4. 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.
5. 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).[11]

2.2.4.5. Kekurangan *Java*

Java memiliki kekurangan diantaranya adalah:

1. Tulis sekali, perbaiki 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*.
2. 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/*direverse-engineer*.
3. 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.[11]

2.2.4.6. Tahap Kompilasi *Java*

Java memiliki beberapa tahap kompilasi, tahapan kompilasi *java* diantaranya adalah:

1. Tulis / Ubah. Pemrogram menulis program dan menyimpannya di media dalam bentuk berkas '.*java*'.

2. Kompilasi. Pengkompilasi membentuk *bytecodes* dari program menjadi bentuk berkas '.class'.
3. Muat. Pemuat kelas memuat *bytecodes* ke memori.
4. Verifikasi. Peng-verifikasi memastikan *bytecodes* tidak mengganggu sistem keamanan *Java*.
5. Jalankan. Penerjemah menerjemahkan *bytecodes* ke bahasa mesin.tidak bisa di pakai‘’.

2.2.5. *Pesonal Home Page* (PHP)

PHP: *Hypertext Preprocessor* adalah bahasa skrip yang dapat ditanamkan atau disisipkan ke dalam HTML. PHP banyak dipakai untuk memrogram situs *web* dinamis. PHP dapat digunakan untuk membangun sebuah CMS.[11]

2.2.5.1. Sejarah PHP

Pada awalnya PHP merupakan kependekan dari *Personal Home Page* (Situs personal). PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada waktu itu PHP masih bernama *Form Interpreted* (FI), yang wujudnya berupa sekumpulan skrip yang digunakan untuk mengolah data formulir dari *web*.

Selanjutnya Rasmus merilis kode sumber tersebut untuk umum dan menamakannya PHP/FI. Dengan perilisan kode sumber ini menjadi sumber terbuka, maka banyak pemrogram yang tertarik untuk ikut mengembangkan PHP.

Pada November 1997, dirilis PHP/FI 2.0. Pada rilis ini, *interpreter* PHP sudah diimplementasikan dalam program C. Dalam rilis ini disertakan juga modul-modul ekstensi yang meningkatkan kemampuan PHP/FI secara signifikan.

Pada tahun 1997, sebuah perusahaan bernama Zend menulis ulang interpreter PHP menjadi lebih bersih, lebih baik, dan lebih cepat. Kemudian pada Juni 1998, perusahaan tersebut merilis interpreter baru untuk PHP dan meresmikan rilis tersebut sebagai PHP 3.0 dan singkatan PHP dirubah menjadi akronim berulang *PHP: Hypertext Preprocessing*.

Pada pertengahan tahun 1999, Zend merilis interpreter PHP baru dan rilis tersebut dikenal dengan PHP 4.0. PHP 4.0 adalah versi PHP yang paling banyak dipakai pada awal abad ke-21. Versi ini banyak dipakai disebabkan kemampuannya untuk membangun aplikasi *web* kompleks tetapi tetap memiliki kecepatan dan stabilitas yang tinggi.

Pada Juni 2004, Zend merilis PHP 5.0. Dalam versi ini, inti dari interpreter PHP mengalami perubahan besar. Versi ini juga memasukkan model pemrograman berorientasi objek ke dalam PHP untuk menjawab perkembangan bahasa pemrograman ke arah paradigma berorientasi objek.

2.2.5.2. Kelebihan PHP

Seluruh aplikasi berbasis *web* dapat dibuat dengan PHP. Namun kekuatanyang paling utama PHP adalah pada konektivitasnya dengan system database di dalam *web*. Sistem database yang dapat didukung oleh PHP adalah Oracle, MySQL , Sybase, PostgreSQL, dan lainnya. PHP dapat berjalan di berbagai system operasi seperti windows 98/NT, UNIX/LINUX, solaris maupun macintosh.

Software ini juga dapat berjalan pada *web server* seperti PWS (Personal *Web Server*), Apache, 2S, AOL*Server*, fhttpd, phttpd dan sebagainya. PHP juga merupakan bahasa pemograman yang dapat kita kembangkan sendiri seperti untuk menambah fungsi-fungsi baru. Keunggulan lainnya dari PHP adalah bahwa PHP juga mendukung komunikasi dengan layanan seperti *protocol* IMAP, SNMP, NNTP, POP3 dan bahkan HTTP. PHP dapat diinstal sebagai bagian atau modul dari apache *web server* atau sebagai CGI script yang mandiri.

Banyak keuntungan yang dapat diperoleh jika menggunakan PHP sebagai modul dari apache, di antaranya adalah :

1. Tingkat keamanan yang cukup tinggi
2. Waktu eksekusi yang lebih cepat dibandingkan dengan bahasa pemograman *web* lainnya yang berorientasi pada *server*-side scripting.
3. Akses ke sistem database yang lebih fleksibel. seperti MySQL.

2.2.6. Gitar

Gitar adalah alat musik berdawai yang dimainkan dengan jari jemari tangan atau sebuah plektrum (alat petik gitar). Bunyinya dihasilkan dari senar-senar yang bergetar. Gitar bisa berupa gitar akustik atau listrik, atau gabungan keduanya. [11]

2.2.6.1. Sejarah Gitar

Keaslian gitar tidak dapat dilihat dari keantikannya. Beberapa ahli merasa alat ini berasal dari benua Afrika, dimana banyak Replika modern dalam bentuk kotak bulat seperti kulitkerang dengan Gut

/ benang benang sutera, di banyak daerah benua itu. Ahli lain menemukan alat ini dalam bentuk kaca di relief relief batu tua di zaman Asia Tengah dan Asia Kuno. Bahan pemikiran lain juga timbul dengan ditemukannya vas vas Yunani Kuno yang bercorak. Greek Strings mungkin adalah alat pertama yang dikatagorikan sebagai gitar. Gitar modern kemungkinan berakar dari gitar Spanyol, tetapi berbagai jenis gitar seperti instrumen instrumen yang kita bisa saksikan dilukisan lukisan pada zaman Medieval dan Renaiassance yang banyak terdapat diseluruh Eropa.

2.2.6.2. Teori Dasar Gitar

Secara umum untuk bisa memahami dan memainkan gitar kita perlu mempersiapkan beberapa hal, diantaranya:

1. Teori musik dasar secara umum, seperti Akord gitar dan tangga nada musik, tangga nada baik # maupun b (mol). Mempelajari akord dasar gitar yang mudah, sperti menghafal C, Dm, Em, F, G, Am, untuk tahap sampai menghafal kord, kita tidak memperdulikan tangga nada musik.
2. Mengetahui cara menggunakan gitar dan kegunaannya. Setelah menghafal akord-akord tadi, tahap memainkan perpaduan-perbaduan dari akord, memainkan gitar dengan pindah dari kord 1 ke akord berikutnya, misal C Em F dan G.
3. Belajar tuning gitar atau stem gitar sendiri stem gitar agak sulit, hal tersebut seorang pemula haru memiliki insting dalam mengenali sebuah nada. Cara lain dalam gitar tuning, yaitu dengan menggunakan sebuah

perangkat seperti gitar tuner/tuning atau sebuah perangkat lunak gitar tuning.

3.2.3.1. Akord

Akord adalah kumpulan tiga nada atau lebih yang bila dimainkan secara bersamaan terdengar harmonis. Akord bisa dimainkan secara terputus-putus ataupun secara bersamaan. Akord ini digunakan untuk mengiringi suatulagu. Ketika Anda menekan tiga tuts piano C, E dan G secara bersamaan, ini berarti anda sudah memainkan akord. Contoh alat musik lainnya yang bisa memainkan akord adalah gitar (akustik dan listrik), organ, *electone*. [11]

3.2.3.1.2. Sejarah Akord

Musik mengenal notasi sejak tahun 590 yang ditemukan oleh Paus Agung Gregori. Notasi ini disebut Notasi Gregorian. Namun disempurnakan sekitar tahun 950 dengan dikenalnya Notasi Balok. Pada awalnya tahun 1600-1750 (Era Barok) orang mengenal musik polifoni dengan ilmu kontrapung, suara kedua berupa melodi lain yang saling bertentangan. Baru tahun 1725 (Era Klasik 1750-1820) orang mengenal akord dalam musik homofoni, *Remeau* menulis buku *Traite d'Harmonie* tahun *1722*. Pada awalnya akord disusun dengan tiga nada, kemudian dibalikkan, berupa mayor, minor, augmented,dmiminished, dan seterusnya ditambah nada septim, sixth, dst. Dan mulai tahun 1750 di Era Klasik akord juga berkembang hingga kini di era blues (1870), jazz (1920), pop dan rock (1955), dll. yang mengenal akord ninth, eleeventh, dan thirdteenth.

3.2.3.1.3. Jenis Akord

Akord itu banyak macamnya. Antara lain akord mayor, akord minor, akord dominan septim, akord *diminished*, akord *augmented*, akord minor 6, akord mayor 7, akord *suspended* dan masih banyak yang lainnya. Akord yang paling sering dipakai dalam suatu lagu yang sederhana adalah akord mayor, akord minor dan akord dominan septim. Akord lainnya digunakan untuk memperindah atau mengubah kualitas suatu lagu. Penyisipan akord yang berbeda akan memberikan efek rasa yang berbeda dalam iringan suatu lagu.

3.2.4. Sistem Multimedia

Sistem multimedia adalah sistem yang menyajikan berbagai bentuk informasi dengan barbagai tehnik seperti gambar, suara, teks, animasi, video yang disajikan secara interaktif untuk memberi informasi atau menghibur.

3.2.4.1. Suara dan Audio

Suara adalah fenomena fisik yang dihasilkan oleh getaran benda berupa sinyal analog dengan amplitudo yang berubah secara kontinyu terhadap waktu. Suara dihasilkan oleh getaran suatu benda. Selama bergetar, perbedaan tekanan terjadi di udara sekitarnya. Pola osilasi yang terjadi dinamakan sebagai "GELOMBANG".

Suara berkaitan erat dengan:

1. Frekuensi
    a. Banyaknya periode dalam 1 detik.
    b. Satuan : Hertz (Hz) atau cycles per second (cps).

c. Panjang gelombang suara (wavelength) dirumuskan = c/f

Dimana c = kecepatan rambat bunyi

Dimana f = frekuensi

Contoh:

Berapakah panjang gelombang untuk gelombang suara yang memiliki kecepatan rambat 343 m/s dan frekuensi 20 kHz?

Jawab:

WaveLength = c/f = 343/20 = 17,15 mm.

Berdasarkan frekuensi, suara dibagi menjadi:

Infrasound 0Hz – 20 Hz

Pendengaran manusia 20Hz – 20 KHz

Ultrasound 20KHz – 1 GHz

Hypersound 1GHz – 10 THz

Manusia membuat suara dengan frekuensi : 50Hz – 10KHz.

Sinyal suara musik memiliki frekuensi : 20Hz – 20Khz.

Sistem multimedia menggunakan suara yang berada dalam range pendengaran manusia.

Suara yang berada pada range pendengaran manusia sebagai “AUDIO”, dan gelombangnya sebagai “ACCOUSTIC SIGNALS”. Suara diluar range pendengaran manusia dapat dikatakan sebagai “NOISE” (getaran yang tidak teratur dan tidak berurutan dalam berbagai frekuensi, tidak dapat didengar manusia).

2. Amplitudo
    a. Keras lemahnya bunyi atau tinggi rendahnya gelombang.
    b. Satuan amplitudo adalah decibel (db)
    c. Bunyi mulai dapat merusak telinga jika tingkat volumenya lebih besar dari 85 dB dan pada ukuran 130 dB akan mampu membuat hancur gendang telinga
2. Velocity
    a. Kecepatan perambatan gelombang bunyi sampai ke telinga pendengar.
    b. Satuan yang digunakan : m/s.
    c. Pada udara kering dengan suhu 20 °C (68 °F)m kecepatan rambat suara sekitar 343 m/s.

REPRESENTASI SUARA

Gelombang suara analog tidak dapat langsung direpresentasikan pada komputer. Komputer mengukur amplitudo pada satuan waktu tertentu untuk menghasilkan sejumlah angka. Tiap satuan pengukuran ini dinamakan “SAMPLE”.