BAB II

LANDASAN TEORI

Teori adalah alur logika atau penalaran, yang merupakan seperangkat konsep, definisi, dan proporsisi, yang disusun secara sistematis (Sugiyono, 2008).

2.1. Program Aplikasi

Program adalah ekspresi, pernyataan atau kombinasi yang disusun dan dirangkai menjadi satu kesatuan prosedur yang berupa urutan langkah – langkah untuk menyelesaikan masalah yang di implementasikan dengan menggunakan bahasa pemrograman sehingga dapat di eksekusi oleh komputer (jogiyanto, 1995).

Aplikasi adalah penerapan, menyimpan sesuatu hal, data, permasalahan pekerjaan ke dalam suatu sarana atau media yang digunakan untuk menerapkan atau mengimplementasikan hal atau permasalahan tersebut sehingga berubah menjadi bentuk yang baru tanpa menghilangkan nilai - nilai dasar dalam hal data, permasalahan dan pekerjaan. Jadi dalam hal ini hanya bentuk dari tampilan data yang berubah, sedangkan isi yang memuat dalam data tersebut tidak mengalami perubahan (Jogiyanto, 1995)

2.1.1. Pengertian program Aplikasi

program aplikasi adalah sederetan kode yang digunakan untuk mengatur komputer supaya dapat melakukan pekerjaan sesuai dengan keinginan programmer/user.

Menurut (jogiyanto, 1995) untuk dapat mengembangkan suatu program aplikasi yang berkualitas dibutuhkan prosedur dan perencanaan yang baik secara

terperinci di dasarkan atas sistem yang ada sehingga dapat menghasilkan aplikasi yang sesuai dengan tujuan dan sasaran. Langkah – langkah pengembangan program aplikasi sebagai berikut:

1. Mendefinisikan masalah.
2. Memilih bahasa pemrograman.
3. Merancang sistem.
4. Membuat kode program.
5. Pelacakan kesalahan program.

2.1.2 Aplikasi Mobile

Mobile dapat diartikan sebagai perpindahan yang mudah dari satu tempat ke tempat yang lain, misalnya telepon mobile berarti bahwa terminal telepon yang dapat berpindah dengan mudah dari satu tempat ke tempat lain tanpa terjadi pemutusan atau terputusnya komunikasi.

sistem Aplikasi mobile merupakan aplikasi yang dapat digunakan walaupun pengguna berpindah dengan mudah dari satu tempat ketempat lain lain tanpa terjadipemutusan atau terputusnya komunikasi. Aplikasi ini dapat diakses melalui perangkat nirkabel seperti pager, seperti telepon seluler dan PDA.

karakteristik perangkat mobile :

1. Ukuran yang kecil

   Perangkat mobile memiliki ukuran yang kecil. Konsumen menginginkan perangkat yang terkecil untuk kenyamanan dan mobilitas mereka.

2. Memory yang terbatas

Perangkat mobile juga memiliki memory yang kecil, yaitu primary (RAM) dan secondary (disk).

3. Daya proses yang terbatas

   Sistem mobile tidaklah setangguh rekan mereka yaitu desktop.

4. Mengkonsumsi daya yang rendah

   Perangkat mobile menghabiskan sedikit daya dibandingkan dengan mesin desktop.

5. Kuat dan dapat diandalkan

   Karena perangkat mobile selalu dibawa kemana saja, mereka harus cukup kuat untuk menghadapi benturan-benturan, gerakan, dan sesekali tetesan-tetesan air.

6. Konektivitas yang terbatas

   Perangkat mobile memiliki bandwith rendah, beberapa dari mereka bahkan tidak tersambung.

7. Masa hidup yang pendek

   Perangkat-perangkat konsumen konsumen ini menyala dalam hitungan detik kebanyakan dari mereka selalu menyala.

2.2 E-Lerning

E-learning merupakan singkatan dari *Elektronic Learning*, merupakan cara baru dalam proses belajar mengajar yang menggunakan media elektronik khususnya internet sebagai sistem pembelajarannya. E-learning merupakan dasar dan konsekuensi logis dari perkembangan teknologi informasi dan komunikasi.

E-learning dalam arti luas bisa mencakup pembelajaran yang dilakukan di media elektronik (internet) baik secara formal maupun informal. E-learning secara formal misalnya adalah pembelajaran dengan kurikulum, silabus, mata pelajaran dan tes yang telah diatur dan disusun berdasarkan jadwal yang telah disepakati pihak-pihak terkait (pengelola e-learning dan pembelajar sendiri).

Pembelajaran seperti ini biasanya tingkat interaksinya tinggi dan diwajibkan oleh perusahaan pada karyawannya atau pembelajaran jarak jauh yang dikelola oleh universitas dan perusahaan-perusahaan (biasanya perusahaan konsultan) yang memang bergerak dibidang penyediaan jasa *e-learning* untuk umum.

E-learning bisa juga dilakukan secara informal dengan interaksi yang lebih sederhana, misalnya melalui sarana mailing list, e-newsletter atau website pribadi, organisasi dan perusahaan yang ingin mensosialisasikan jasa, program, pengetahuan atau keterampilan tertentu pada masyarakat luas (biasanya tanpa memungut biaya).

2.3 Ilmu Tajwid Dasar

Mempelajari dan mengajarkan Al-Qur'an merupakan karakteristik dari generasi Rabbani sebagaimana yang digambarkan Allah dalam Surah Ali Imran ayat 79.

Rasul memerintahkan kaum Muslimin untuk mengajarkan anak-anak untuk membaca Al-Qur'an sejak umur 5-6 tahun, sebab pada umur 7 tahun sudah disuruh mengerjakan sembahyang, "*Suruhlah anak-anakmu mengerjakan shalat*

bila sudah berumur 7 tahun, dan pukullah ia bila tidak mengerjakannya kalau sudah berumur 10 tahun."

Penulisan Al-Qur'an sudah dimulai sejak masa khulafaur rasyidin, namun masih belum mempergunakan tanda-tanda baca. Disebabkan perbedaan penulisan Al-Qur'an, maka muncul berbagai qiraat yang dikenal sebagai qiraat sab'ah, qiraat sepuluh, dan qiraat empat belas. *Qiraat sab'ah* merupakan qiraat yang mutawatir. Pada masa Muawiyah, kemudian tulisan Al-Qur'an diberi tanda-tanda baca dengan menggunakan titik-titik dengan warna yang berbeda-beda, yang dilakukan oleh Abul Aswad Ad-Duali. Namun karena lamakelaman warna itu hampir menjadi serupa, maka setelah abad ke-4 Hijriah, Al Khalil membuat tanda-tanda baru yang kemudian dikenal oleh kita hingga sekarang.

2.3.1 Pengertian Tajwid

Tajwid menurut bahasa berasal dari kata جوّد-يجوّد-تجويدا yang berarti bagus atau membaguskan. Dalam ilmu Qiraah, tajwid berarti mengeluarkan huruf dari tempatnya dengan memberikan sifat-sifat yang dimilikinya. Jadi ilmu tajwid adalah suatu ilmu yang mempelajari bagaimana cara membunyikan atau mengucapkan huruf-huruf yang terdapat dalam kitab suci al-Quran maupun bukan.

Adapun masalah-masalah yang dikemukakan dalam ilmu ini adalah *m akharijul huruf* (tempat keluar-masuk huruf), *shifatul huruf* (cara pengucapan huruf), *ahkamul huruf* (hubungan antar huruf), *ahkamul maddi wal qasr* (panjang dan pendek ucapan), *ahkamul waqaf wal ibtida'* (memulai dan menghentikan bacaan) dan al-Khat al-Utsmani.

Inilah yang dimaksud dengan membaca al-Qur'an dengan tartil sebagaimana firman-Nya : (المزمّل : 4) و رتل القران ترتيلا yang artinya : "Bacalah al-Qur'an itu dengan tartil". Sedangkan arti tartil menurut *Ibn Katsir* adalah membaca dengan perlahan-lahan dan hati-hati karena hal itu akan membantu pemahaman serta perenungan terhadap al-Qur'an.

Ilmu Tajwid bertujuan untuk memberikan tuntunan bagaimana cara pengucapan ayat yang tepat, sehingga lafal dan maknanya terpelihara. Pengetahuan tentang makhraj huruf memberikan tuntunan bagaimana cara mengeluarkan huruf dari mulut dengan benar. Pengetahuan tentang sifat huruf berguna dalam pengucapan huruf. Dalam ahkamul maddi wal qashr berguna untuk mengetahui huruf yang harus dibaca panjang dan berapa harakat panjang bacaannya. Ahkamul waqaf wal ibtida' ialah cara untuk mengetahui dimana harus berhenti dan dari mana dimulai apabila bacaan akan dilanjutkan.

2.3.2 Makhraj Huruf Hija'iyah dan Tanda Baca dalam Al-Quran

Makhraj artinya tempat keluar. Makhraj huruf hija'iyah artinya tempat keluarnya huruf-huruf hija'iyah dari mulut. Mengingat huruf Arab berbeda dengan huruf latin, maka cara pembacaannya berbeda dengan bacaan pada bahasa melayu. Perbedaan pembacaan/bunyi huruf dalam bahasa Arab ditentukan oleh makhraj hurufnya.

Tanda Baca dalam Al-Qur'an

Beberapa tanda baca dalam Al-Qur'an yang dikenal standard ada delapan buah, yaitu tanda sukun/mati, fathah, kasrah, dhommah, tanwin (3 macam), dan

tasydid. Selain itu pula ada tanda baca lain untuk menyatakan tempat-tempat berhenti/dilarang berhenti, yaitu Waqaf.

2.3.3 Hukum Nun Mati dan Tanwin

Hukum nun mati dan tanwin dapat dikelompokkan menjadi beberapa jenis, diantaranya yaitu izhhaar, idgham, iklab, dan ikhfaa

1. Izhhaar

Izhhaar secara bahasa berarti jelas. Dalam ilmu tajwid, yang dimaksud dengan izhhaar adalah bacaan dimana bunyi huruf nun pada saat nun mati atau tanwin bertemu dengan huruf-huruf izhhaar dibaca jelas. Bacaan izhhaar adalah satu harakat. Huruf-huruf izhhaar adalah “alif”, “hamzah”, “ ‘ain”, “ghin”, “ha”, “kha”, dan “ha’ ”

2. Idgham

Idgham berarti memasukkan/merubah bunyi huruf nun pada saat nun mati atau tanwin bertemu dengan huruf-huruf idgham kepada huruf-huruf idgham. Setiap bacaan idgham adalah dua harakat. Huruf-huruf idgham terangkum dalam kata “yarmaluwna”, yaitu “ya”, “ra”, “mim”, “lam”, waw”, dan “nun”. Idgham terbagi atas dua macam, yaitu idgham bighunnah dan idgham bilaghunnah.

a) Idgham Bighunnah

   Hukum bacaan idgham bighunnah terjadi apabila nun mati menghadapi huruf-huruf “ya”,“nun”, “mim”, dan “waw”. Bighunnah artinya dengan dengung, mengandung makna bahwa pada saat membaca idhgam ini bunyi huruf dimasukkan ke hidung seakan seperti orang mendengung.

b) Idgham Bilaghunnah

Hukum bacaan idgham bilaghunnah merupakan kebalikan dari idgham bighunnah, dimana bacaan idgham tidak dengan memasukkan bunyi pada hidung, tetapi tempat keluarnya huruf idgham tersebut (pada makhrajnya). Huruf idgham bilaghunnah adalah “lam” dan “ra”.

3. Iqlab

Hukum *iqlab* terjadi apabila nun mati atau tanwin bertemu dengan huruf “ba”, dimana bunyi huruf nun pada saat nun mati atau tanwin bertemu dengan huruf “ba” dibaca menjadi mim disertai dengan dengung. Setiap bacaan yang mengandung iqlab dibaca dua harakat.

4. Ikhfa

Ikhfa berarti menyembunyikan/samar, yang maksudnya menyembunyikan atau menyamarkan bunyi huruf nun pada saat nun mati atau tanwin bertemu dengan huruf-huruf ikhfa. Semua bacaan dengan ikhfa adalah dua harakat. Huruf-huruf ikhfa adalah huruf-huruf selain pada izhhaar, idgham, dan iqlab, sebanyak 15 buah huruf. Ikhfa terbagi atas tiga macam berdasarkan bunyi huruf yang dihasilkannya, yaitu nun mati atau tanwin bertemu huruf-huruf :

1. “kaf” dan “qaf”, bunyi huruf nun menjadi “ng” mendengung.
2. “tsa”, “dal”, dan “tha”, lebih samar dari kasus pertama
3. selain hurur-huruf pada kasus pertama dan kedua dibaca lebih samar lagi.

2.3.4 Hukum Mim Mati dan Ghunnah

2.3.4.1 Hukum Mim Mati

Hukum mim mati terdiri dari :

1. Idgham Mutamatsilayn/Miimi

Hukum *idgam mutamatsilayn* terjadi apabila mim mati bertemu dengan huruf “mim”, dimana bunyi huruf mim pada mim mati dimasukkan (diidghamkan) pada bunyi huruf selanjutnya dengan dengung. Pada keadaan ini bacaan dibaca dua harakat.

2. Ikhfa Syafawii

   Jika mim mati bertemu dengan huruf “ba”, maka bunyi huruf mim pada mim mati dibaca dengan samar-samar dengan sedikit dengung. Dalam hal ini bacaan dibaca dua harakat, dan hukumnya bernama ikhfa syafawii. Bacaan dengan ikhfa syafawii adalah dua harakat.

3. Izhhaar Syafawii

   Namun apabila huruf mim mati bertemu dengan huruf selain “mim” dan “ba”, maka bunyi huruf mim pada mim mati dibaca jelas. Karena bacaan izhhaar adalah satu harakat, begitupula halnya pada bacaan dengan menggunakan mim mati dibaca pula satu harakat.

2.3.4.2 Gunnah

Ghunnah berarti mendengung. Bacaan ghunnah terjadi dalam dua kasus, yaitu apabila huruf “mim” dan “nun” mempergunakan tanda tasydid. Bacaan ghunnah senantiasa dua harakat.

2.3.5 Hukum Alif-Lam dan Qalqalah

2.3.5.1 Hukum Alif-Lam

Hukum Alif-Lam dibagi atas dua macam, yaitu Alif-Lam Syamsiyah dan Alif-Lam Qamariyah.

1. Alif-Lam Syamsiyah

Apabila Alif-Lam bertemu dengan huruf-huruf syamsiyah, maka Alif-Lam dimasukkan ke dalam bunyi huruf selanjutnya.

2. Alif-Lam Qamariyah

   Apabila Alif-Lam bertemu dengan huruf-huruf qamariyah, maka Alif-Lam dibaca secara jelas.

2.3.5.2 Qalqalah

Qalqalah adalah membaca bunyi-bunyi huruf qalqalah dengan memantul karena diberi tanda sukun atau karena diwaqafkan. Huruf-huruf qalqalah terangkum dalam kalimat “baju di thoqo”, yaitu “ba”, “jim”, “dal”,“tho”, dan “qaf”. Ada dua macam qalqalah yang dikenal, yaitu qalqalah sughra dan qalqalah qubra.

1. Qalqalah Sughra

   Qalqalah sughra terjadi karena huruf-huruf qalqalah diberi tanda sukun/mati. Biasanya muncul pada pertengahan ayat.

2. Qalqalah Qubra

   Qalqalah sughra terjadi karena huruf-huruf qalqalah diwaqafkan. Biasanya muncul pada pertengahan ayat jika ia di waqafkan atau di akhir ayat (karena berhenti).

2.3.6 Hukum Mad

Mad artinya memanjangkan bunyi huruf-huruf. Di dalam pelajaran tajwid ada dua macam mad, yaitu mad ashli/tabi’i dan mad far’i. Ashli artinya pokok, dan far’i artinya cabang.

1. Mad Ashli/Thabi’i

Setiap bacaan mad ashli adalah dua harakat. Mad ini terjadi apabila terdapat alif sesudah baris huruf dengan fathah, ya mati sesudah baris huruf dengan kasrah, dan waw mati sesudah baris huruf dengan dhomah. Selain itu dapat pula mempergunakan tanda-tanda panjang jika ia tidak mempergunakan huruf-huruf alif, waw mati , ataupun ya mati.

2. Mad Far'i

Merupakan mad yang memiliki banyak jenisnya dengan panjang harakat yang berbeda-beda. Begitupula pada aplikasinya dapat berbeda-beda untuk setiap qiraat. Dalam materi ini hanya akan disampaikan beberapa hukum mad cabang ini di dalam kebanyakan buku-buku pelajaran tajwid.

a) Mad Wajib Muttashil

   Di dalam mushaf Qur'an Saudi, jenis mad ini dapat mudah ditemukan yaitu dengan mempergunakan tanda baca panjang (~) yang diikuti dengan huruf hamzah dalam satu kata. Tanda mad ini memiliki panjang 5 harakat.

b) Mad Jaiz Munfashil

   Jenis mad ini juga dapat dengan mudah ditemukan di dalam mushaf Qur'an Saudi, yaitu dengan mempergunakan tanda baca panjang (~) yang diikuti dengan huruf hamzah yang tidak dalam satu kata. Tanda mad ini memiliki panjang 3-5 harakat. Sebagian yang lain mengatakan panjang harakat ini sama dengan panjang harakat pada mad wajib muttashil.

c) Mad 'Aridh Lisukuun

Mad ini terjadi apabila seorang qori Al-Qur'an mewaqafkan bacaan dimana pada bacaan yang diwaqafkannya ditemukan mad ashli/thabi'i yang diikuti oleh baris huruf dengan tanda selain tanwin fathah. Panjangnya mad ini adalah 2, 4, dan 6 harakat.

d) Mad 'Iwadh

Mad ini terjadi apabila seorang qori Al-Qur'an mewaqafkan bacaan dimana pada bacaan yang diwaqafkannya ditemukan tanda tanwin fathah. Maka bunyi nun pada tanwin fathah pada bacaannya dihilangkan. Panjangnya mad ini senantiasa 2 harakat.

e) Mad Laazim Mutsaqqal Kilmi

Jenis mad ini juga dapat dengan mudah ditemukan di dalam mushaf Qur'an Saudi, yaitu dengan mempergunakan tanda baca panjang (~) yang diikuti dengan huruf yang bertanda tasydid. Mad ini panjangnya 6 harakat.

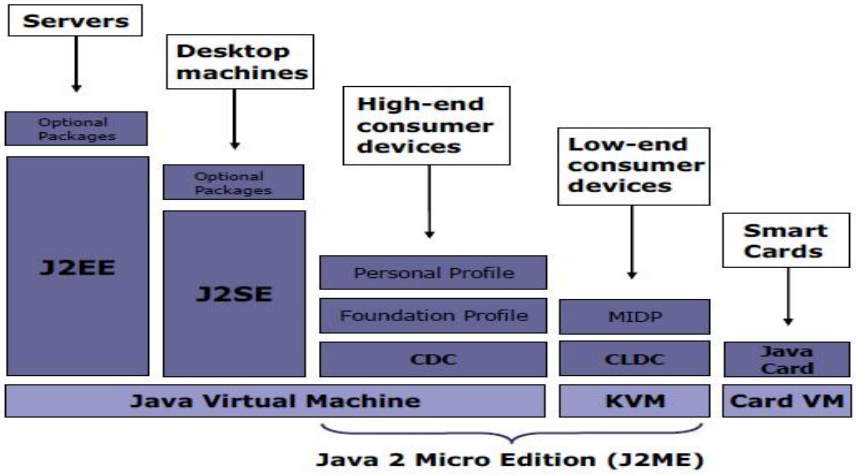
2.3.7 Hukum Waqaf

Secara bahasa *waqaf* artinya terhenti/tertahan. Menurut istilah ilmu tajwid, waqaf maksudnya memutuskan suara pembacaan atas suatu kalimat untuk menarik napas, dengan berniat untuk mengulangi bacaannya atau memang berwaqaf di tempat yang pantas.

Dalam setiap mushaf menampilkan tanda-tanda waqaf yang berbeda. Oleh karena itu,dalam pelajaran ini hanya akan dibahas tanda-tanda waqaf yang dipergunakan pada mushaf Qur'an Saudi saja. Tanda-tanda waqaf pada mushaf ini terdapat pada halaman-halaman terakhir mushaf.

2.4 JAVA

Java adalah bahasa pemrograman yang dapat dijalankan di berbagai komputer termasuk telepon genggam. Dikembangkan oleh Sun Microsystems dan diterbitkan tahun 1995. Java tidak boleh disalahpahami sebagai JavaScript. JavaScript adalah bahasa *scripting* yang digunakan oleh *web browser*.



Gambar 2.1. Flatform Java

2.4.1 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).

2.4.2 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:

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.

2.4.3 Kelebihan Java

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, Mrac OS dan Sun Solaris. Penyebanya adalah setiap sistem operasi menggunakan programnya sendiri-sendiri (yang dapat diunduh dari situs Java) untuk menginterpretasikan *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).

2.4.4 Kekurangan Java

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/di-*reverse-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.

2.4.5 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 tidak bisa di pakai.

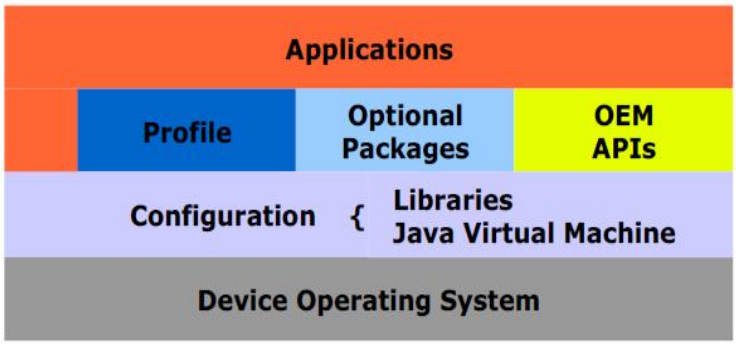
2.5 Java2 Micro Edition (J2ME)

2.5.1 Pengertian J2ME

Java2 Micro Edition atau yang biasa disebut J2ME adalah lingkungan pengembangan yang didesain untuk meletakan perangkat lunak java pada barang elektronik beserta perangkat pendukungnya. Pada J2ME, jika perangkat lunak berfungsi baik pada sebuah perangkat maka belum tentu juga berfungsi baik pada perangkat yang lainnya.

J2ME membawa java ke dunia informasi, komunikasi, dan perangkat komputasi selain perangkat komputer desktop yang biasanya lebih kecil dibandingkan perangkat komputer *desktop*. J2ME biasanya digunakan pada telepon selular, *pager*, *personal digital assistants* (PDA's) dan sejenisnya.

J2ME adalah bagian dari J2SE, karena itu tidak semua library yang ada pada J2SE dapat digunakan pada J2ME. Tetapi J2ME mempunyai beberapa library yang tidak dimiliki J2SE. Arsitektur J2ME dapat dilihat pada gambar berikut :



Gambar 2.2. Arsitektur J2ME.

Teknologi J2ME juga memiliki beberapa keterbatasan, terutama jika diaplikasikan pada ponsel. J2ME sangat tergantung pada perangkat *(device)* yang

digunakan, bisa dari segi merk ponsel, maupun kemampuan ponsel, dan dukungannya terhadap teknologi J2ME.

Configuration merupakan java library minimum dan kapabilitas yang dipunya oleh para pengembang J2ME, yang dimaksud sebuah *mobile device* dengan kemampuan java akan dioptimalkan untuk menjadi sesuai. Dalam J2ME telah didefinisikan dua buah konfigurasi yaitu CLDC *(Connected Limited Device Configuration*) untuk perangkat kecil dan CDC *(connected Device Configuration)* untuk perangkat yang lebih besar.

Profile berbeda dengan *configuration, profile* membahas sesuatu yang spesifik untuk sebuah perangkat. Dalam J2ME terdapat dua buah *profile* yaitu MIDP dan *Foundation profile.*

2.5.2 Connected Limited Device Configuration (CLDC)

CLDC atau *Connected Limited Device Configuration* adalah perangkat dasar dari J2ME, spesifikasi dasar yang berupa *library* dan API yang diimplementasikan pada J2ME, seperti yang digunakan pada telepon *selular, pager*, dan PDA. Perangkat tersebut dibatasi dengan keterbatasan memory, sumber daya, dan kemampuan memproses.

Spesifikasi CLDC pada J2ME adalah spesifikasi minimal dari *package,* kelas, dan sebagian fungsi *Java Virtual machine* yang dikurangi agar dapat diimplemetasikan dengan keterbatasan sumber daya pada alat-alat tersebut, JVM yang digunakan disebut KVM (*Kilobyte Virtual Machine*).

2.5.3 Connected Device Configuration (CDC)

CDC atau *Connected Device Configuration* adalah spesifikasi dari konfigurasi J2ME. CDC merupakan komunitas proses pada java yang memiliki standarisasi. CDC terdiri dari *virtual machine* dan kumpulan *library* dasar untuk dipergunakan pada profile industri. Implemetasi CDC pada J2ME adalah *source code* yang menyediakan sambungan dengan macam-macam platform.

2.5.4 Mobile Information Device Profile (MIDP)

MIDP adalah spesifikasi untuk sebuah profil J2ME. MIDP memiliki lapisan di atas CLDC, API tambahan untuk daur hidup aplikasi, antarmuka, jaringan, dan penyimpanan persisten, pada saat ini terdapat MIDP 1.0 adalah API untuk multimedia, pada MIDP 2.0 terdapat dukungan memainkan *tone, tone sequence* dan *file* WAV walaupun tanpa adanya *mobile media* API (MMAPI).

2.5.5 Kilobyte Virtual Machine (KVM)

KVM adalah paket JVM yang di desain untuk perangkat kecil. KVM mendukung sebagian dari fitur-fitur JVM, seperti misalnya KVM tidak mendukung operasi *floating-point* dan finalisasi objek, KVM diimplemetasikan dengan menggunakan C sehingga sangat mudah beradaptasi pada tipe *platform* yang berbeda.

2.5.6 C-Virtual Machine (CVM)

CVM adalah paket JVM optimal yang digunakan pada CDC. CVM mempunyai seluruh fitur dari *virtual machine* yang didesain untuk perangkat yang memerlukan fitur-fitur java2 *virtual machine.*

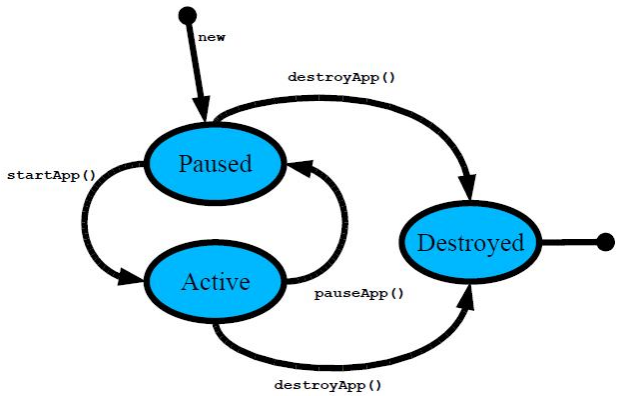
2.5.7 MIDlet

MIDlet adalah aplikasi yang ditulis untuk MIDP. Aplikasi MIDlet adalah bagian dari kelas javax.microedition.midlet.MIDlet yang didefinisikan pada MIDP. MIDlet berupa sebuah kelas abstrak yang merupakan subkelas dari bentuk sasar aplikasi menajemen pada perangkat dapat terbentuk.

2.5.7.1 Siklus MIDlet

kehidupan MIDlet dimulai ketika di-instantiate oleh AMS. MIDlet pada awalnya masuk status “Pause” setelah perintah baru dibuat. AMS memanggil constructor public tanpa argumen dari MIDlet. Jika sebuah exception terjadi dalam constructor, MIDlet memasuki status “Destroyed” dan membuangnya segera. MIDlet masuk ke dalam status “Active” atas pemanggilan method startUp() oleh AMS. MIDlet masuk ke dalam status “Destroyed” ketika AMS memanggil method destroyApp().

Status ini juga kembali diakses ketika method notifyDestroyed() kembali dengan sukses kepada aplikasi. Dengan catatan bahwa MIDlet hanya bisa memasuki status “Destroyed” sekali dalam masa hidupnya.



Gambar 2.3. daur hidup MIDlet.

2.5.8 JAD (Java Application Descriptor)

Digunakan untuk mendeskripsikan isi aplikasi untuk keperluan pemetaan. File JAD berisi deskripsi file JAR (*Java Archive*) dan pemetaan atribut MIDlet, sedangkan file JAR berisi kumpulan kelas dan *resource*.

2.6 Unified Modeling Language (UML)

Unified Modeling Language (UML) adalah bahasa spesifikasi standar untuk mendokumentasikan, menspesifikasikan, dan membangun sistem perangkat lunak. UML adalah himpunan struktur dan teknik untuk pemodelan desain program berorientasi objek (OOP) serta aplikasinya. UML adalah metodologi untuk mengembangkan sistem OOP dan sekelompok perangkat *tool* untuk mendukung pengembangan sistem tersebut. UML mulai diperkenalkan oleh *Object Management Group*, sebuah organisasi yang telah mengembangkan model, teknologi, dan standar OOP sejak tahun 1980-an. Sekarang UML sudah 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 Booch, Jim Rumbaugh, dan Ivar Jacobson. Namun demikian UML dapat digunakan untuk memahami dan mendokumentasikan setiap sistem informasi. Penggunaan UML dalam industri terus meningkat. Ini merupakan standar terbuka yang

menjadikannya sebagai bahasa pemodelan yang umum dalam industri peranti lunak dan pengembangan sistem.

2.6.1 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 sebagainya. 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 metodologinya 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*.

2.6.2 Diagram UML

Mengutip dari jurnal Andrew Bissell, (2003) "UML2.0 — a major revision of the industries de facto software modelling language", Aircraft Engineering and Aerospace Technology, Vol. 75 Iss: 2, The UML is a graphical language for specifying, visualising, constructing and documenting the artifacts of software systems. It represents a collection of the best engineering practices proven successful in modelling large and complex systems.

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.

Berikut akan dijelaskan empat macam diagram yang paling sering digunakan dalam pembangunan aplikasi berorientasi object, yaitu *use case diagram, sequence diagram, collaboration diagram, dan class diagram.*

1. *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 merepresentasikan orang yang akan mengoperasikan atau orang yang berinteraksi dengan sistem aplikasi.

Use case merepresentasikan operasi-operasi yang dilakukan oleh aktor. *Use case* digambarkan berbentuk elips dengan nama operasi dituliskan di dalamnnya. *Actor* yang melakukan operasi dihubungkan dengan garis lurus ke *usecase*.

2. 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.

3. Collaboration Diagram

Collaboration diagram dipakai untuk memodelkan interaksi antar objek 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 objek.

4. 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.

2.7 NetBeans IDE 7.0.1

NetBeans IDE merupakan Integrated Development Environment (IDE) open source untuk software developer. IDE ini berjalan pada berbagai macam platform

termasuk Windows, Linux, Solaris, dan MacOS. Selain itu, NetBeans IDE juga merupakan software yang mudah diinstal dan powerful.



Gambar 2.4. Aplikasi NetBeans IDE 7.0.1

NetBeans IDE menyediakan kemampuan untuk developer dengan semua tool yang dibutuhkan untuk membuat aplikasi desktop, aplikasi enterprise, aplikasi web dan juga aplikasi mobile yang profesional serta lintas platform. Pengguna dapat menyediakan platform Java, C/C++, PHP, JavaScript ataupun Groovy untuk mengembangkan aplikasi yang mereka inginkan.

NetBeans IDE 7.0 memperkenalkan language support untuk pengembangan pada Java SE 7 specification dengan fitur-fitur JDK 7 language. Rilis ini juga menyediakan peningkatan integrasi dengan Oracle WebLogic server, serta dukungan untuk Oracle. Database dan GlassFish 3.1.

Fitur utama lainnya termasuk Maven 3 dan HTML editing support; GridBagLayout designer baru untuk meningkatkan pengembangan Swing GUI; peningkatan pada Java editor, dan masih banyak lagi. NetBeans IDE 7.0 tersedia dalam berbagai bahasa termasuk, bahasa Inggris, Brazilian Portugese, Jepang, Rusia, dan bahasa Cina (Simplified Chinese).

Versi NetBeans IDE terbaru (7.0.1) menyertakan beberapa perubahan yang cukup signifikan, diantaranya Full JDK 7 support. Menjalankan NetBeans IDE di atas JDK 7 dan mendukung versi JDK 7 language feature final Integrasi patch-patch terbaru, serta peningkatan performa.

Untuk menginstal NetBeans IDE 7, pastikan bahwa komputer telah memenuhi persyaratan requirement yang telah ditentukan. Sebagai contoh, Java 6 dibutuhkan untuk menginstal dan menjalankan PHP dan C/C++ NetBeans Bundles. Selain itu, JDK 6 dibutuhkan untuk menginstal dan menjalankan Java SE, Java EE dan semua NetBeans Bundles.

NetBeans IDE tersedia dalam berbagai pilihan yang dapat disesuaikan dengan bahasa pemrograman yang ingin digunakan. Pilihan tersebut akan berpengaruh pada ukuran file instalasi yang akan digunakan. NetBeans IDE didistribusikan di bawah lisensi GNU General Public License (GPL) v2, sehingga dapat di download dan menggunakannya secara gratis untuk penggunaan umum.

2.8 J2ME Wireless Toolkit

J2ME Wireless Toolkit adalah apliaksi yang menyediakan lingkungan emulator, dokumentasi beserta contoh-contoh apliaksi java untuk perangkat kecil *(small device)*. J2ME WTK berbasiskan pada CLDC dan MIDP, J2ME WTK adalah program yang meniru kerja ponsel yang mendukung MIDP atau yang biasa disebut *emulator*. Oleh karena itu, belum tentu MIDlet yang berjalan di *emulator* juga berjalan pada ponsel yang sebenarnya, karena juga bergantung pada kemampuan dan kapasitas ponsel yang digunakan.