

BAB II

LANDASAN TEORI

2.1 TCP/IP dan Internet

TCP/IP (*Transmission Control Protocol/Internet Protocol*) adalah protokol yang digunakan untuk melakukan komunikasi data di Internet. Antara TCP/IP dan Internet memiliki hubungan yang cukup erat, sehingga agak sulit menjelaskan Internet tanpa mengikutsertakan sedikit saja pembahasan mengenai TCP/IP.

Sebagaimana yang dilakukan oleh manusia, komunikasi antar dua pihak atau lebih memerlukan sebuah standar cara berkomunikasi, yang dalam dunia manusia dinamakan bahasa. Seperti peranan bahasa Indonesia yang menyatukan penduduk Indonesia dalam berkomunikasi, padahal terdiri dari berbagai suku bangsa yang masing-masing memiliki bahasa daerah, TCP/IP juga memiliki peranan yang hampir-hampir mirip dengan bahasa dalam pergaulan antar manusia.

Pergaulan antar komputer, yang lazimnya dinamakan komunikasi data digital, juga memerlukan bahasa. Dalam dunia jaringan komputer, istilah bahasa tidak digunakan, namun biasanya dinamakan sebagai protokol. Pada saat ini dapat dilihat seberapa besar peranan TCP/IP dalam menyatukan 'penduduk' Internet, antara lain yaitu penggunaan e-mail, chatting, dan browsing.

Protokol mengatur bagaimana sebuah komputer berkomunikasi dengan komputer lain. Dalam jaringan komputer kita dapat menggunakan banyak macam protokol tetapi agar dua buah komputer dapat berkomunikasi, keduanya perlu menggunakan protokol yang sama. Komputer-komputer yang terhubung ke

Internet berkomunikasi dengan protokol TCP/IP. Karena menggunakan bahasa yang sama, yaitu protokol TCP/IP, perbedaan jenis komputer dan sistem operasi tidak menjadi masalah. Komputer PC dengan sistem operasi Windows dapat berkomunikasi dengan komputer PowerPC dengan sistem operasi Macintosh [PUR02].

Secara sederhana dapat disimpulkan bahwa apabila sebuah komputer menggunakan protokol TCP/IP dan terhubung ke Internet, maka komputer tersebut dapat berhubungan dengan komputer di belahan dunia mana pun yang juga terhubung ke Internet.

2.1.1 Karakteristik TCP/IP

Perkembangan TCP/IP yang diterima luas dan praktis menjadi standar de-facto jaringan komputer berkaitan dengan ciri-ciri yang terdapat pada protokol itu sendiri [PUR02]:

- a. Protokol TCP/IP dikembangkan menggunakan standar protokol yang terbuka.
- b. Standar protokol TCP/IP dalam *Request for Comment* (RFC) dapat diambil oleh siapapun tanpa biaya.
- c. TCP/IP dikembangkan dengan tidak bergantung pada sistem operasi atau perangkat keras tertentu.
- d. Pengembangan TCP/IP dilakukan dengan konsensus dan tidak bergantung pada *vendor* tertentu.

- e. TCP/IP independen terhadap perangkat keras jaringan dan dapat dijalankan pada jaringan Ethernet, Token Ring, jalur telepon dial-up, jaringan X.25, dan praktis jenis media transmisi apapun.
- f. Pengalamatan TCP/IP bersifat unik dalam skala global. Dengan cara ini, komputer dapat saling terhubung walaupun jaringannya seluas Internet sekarang ini.
- g. TCP/IP memiliki fasilitas routing sehingga dapat diterapkan pada inter-network.
- h. TCP/IP Memiliki banyak jenis layanan.

2.1.2 Standarisasi Internet

Ada empat badan yang bertanggung jawab dalam mengatur, mengontrol, serta melakukan standarisasi protokol yang digunakan di Internet [PUR02]:

1. *Internet Society* (ISOC). ISOC merupakan badan profesional yang memfasilitasi, mendukung, serta mempromosikan pertumbuhan Internet, sebagai infrastruktur komunikasi global untuk riset/penelitian. Badan ini berurusan tidak hanya dengan aspek-aspek teknis, namun juga aspek politik dan sosial dari jaringan Internet.
2. *Internet Architecture Board* (IAB). IAB merupakan badan koordinasi dan penasehat teknis bagi *Internet Society*. Badan ini bertindak sebagai badan review teknis dan editorial akhir semua standar Internet. IAB memiliki otoritas untuk menerbitkan dokumen standar Internet yang dikenal sebagai RFC (*Request for Comment*). Tugas lain dari IAB adalah mengatur angka-angka dan konstanta yang digunakan dalam protokol Internet (nomor port

TCP, kode protokol IP, tipe hardware ARP dan lain-lain). Tugas ini didelegasikan ke lembaga yang disebut *Internet Assigned Numbers Authority* (IANA).

3. *Internet Engineering Task Force* (IETF). Badan ini berorientasi untuk membentuk standar Internet. IETF masih dibagi lagi menjadi sembilan kelompok kerja (misalnya aplikasi, routing dan addressing, keamanan komputer) dan bertugas menghasilkan standar-standar Internet. Untuk mengatur kerja IETF ini, dibentuk badan yang disebut *Internet Engineering Steering Group* (IESG).
4. *Internet Research Task Force* (IETF). Badan ini dibentuk untuk tujuan orientasi pada riset-riset jangka panjang.

2.2 DNS

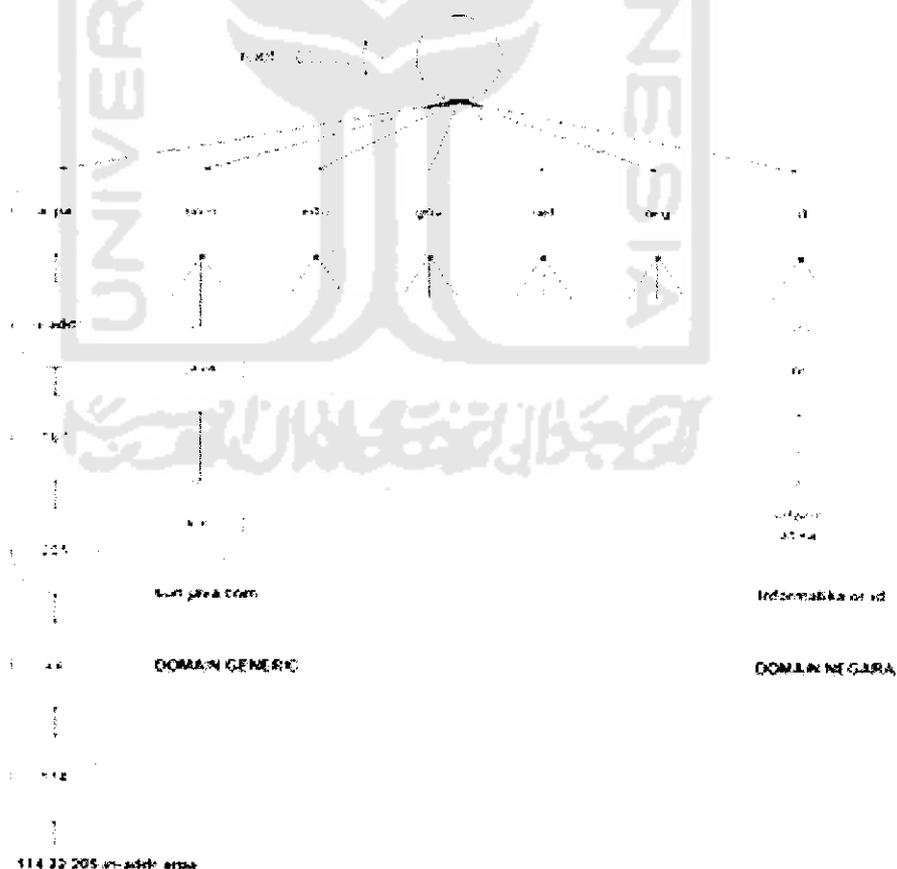
DNS (*Domain Name System*) merupakan sebuah cara untuk mengidentifikasi sebuah komputer di dalam jaringan menggunakan nama domain yang dimilikinya. Seperti yang telah diketahui bahwa pada dasarnya komputer di jaringan hanya mampu mengenali Alamat IP saja, bukan nama komputernya, satu hal yang sangat merepotkan bagi manusia tentunya untuk mengingat banyak Alamat IP untuk keperluan akses yang berbeda.

Pada awal sebelum penggunaan DNS, digunakan sistem *hosts table* (biasanya pada sistem operasi terdapat pada file *hosts*) yang lebih lazim disebut sebagai *hosts table*. Namun, di kemudian hari ternyata cara ini terbukti kurang efisien, karena jumlah host di Internet sangatlah banyak, dan hampir tidak mungkin dilakukan sistem pentabelan secara manual seperti itu. Oleh karena itu

kemudian muncul gagasan dari para pakar jaringan untuk menciptakan sebuah protokol standar, khusus untuk menangani masalah tersebut.

DNS memberikan pemecahan masalah yang masih terbukti handal hingga saat ini, dengan beberapa keuntungan dibandingkan sistem *hosts table*, antara lain:

1. Adanya distribusi tanggung jawab penggunaan domain dan *zone* (satu lingkungan domain yang sama).
2. Adanya distribusi kepemilikan data-data domain, artinya tidak setiap domain dan *zone* harus dimiliki data-datanya oleh setiap pihak, namun data-data tersebut tetap dapat diakses ketika ada pihak-pihak yang memerlukannya (misalnya ketika melakukan *resolve* DNS).



Gambar 2.1 Domain Name System

2.3 IRC

Protokol IRC telah dikembangkan selama lebih dari empat tahun sebelum akhirnya diimplementasikan untuk pertama kalinya sebagai alat untuk para pengguna pada sebuah BBS untuk melakukan komunikasi di antara mereka. Hingga saat ini IRC telah digunakan pada banyak jaringan komputer secara mendunia (dengan adanya Internet) dan dalam pemakaiannya masih terus tumbuh dan mengalami perkembangan. Protokol IRC merupakan protokol yang berbasis teks, dengan aplikasi client sederhana yang mengimplementasikan fungsi socket sudah dapat digunakan untuk terhubung ke server jaringan IRC [OIK93].

IRC (*Internet Relay Chat*) adalah protokol terbuka yang ditemukan di Finlandia pada tahun 1988. Sebuah aplikasi client (yang beberapa telah disediakan) didownload oleh para pemakai dan kemudian melalui aplikasi tersebut mereka dapat terhubung ke jaringan IRC. Administrator sistem hanya bertugas menjalankan dan memelihara jaringan saja, sehingga orang-orang bebas memilih jaringan IRC mana yang akan mereka masuki.

Jaringan yang berbeda memiliki fokus yang berbeda. Sebagai contoh, ada sebuah jaringan terbuka (irc.openprojects.net, irc.slashdot.org) dimana para penciptanya yang *open-source* dapat mengkolaborasikan proyek-proyek pengembangan perangkat lunak serta bertukar ide. Saat ini ada banyak jaringan yang sudah berjalan dengan baik, salah satunya adalah *Eris Free Network* (EFNet), yang memiliki 44076 channel aktif dan 115688 pemakai aktif [MUD03].

2.3.1 Server

Server berperan sebagai *backbone* dari IRC yang menyediakan suatu titik pertemuan untuk setiap client yang terhubung untuk berkomunikasi dengan client lainnya. Selain itu juga menjadi titik pertemuan bagi server-server lainnya sehingga membentuk sebuah jaringan IRC [OIK93].

2.3.2 Client

Client adalah segala sesuatu yang terhubung ke sebuah server yang bukan merupakan server lainnya. Setiap client dibedakan dari client lainnya melalui sebuah nickname yang bersifat unik, dengan maksimum terdiri dari sembilan karakter. Ada aturan-aturan tertentu mengenai karakter apa saja yang diperbolehkan untuk membuat sebuah nickname. Berkaitan dengan nickname, semua server harus memiliki seluruh informasi berikut: host yang digunakan oleh client, username yang digunakan oleh client pada host tersebut, dan server yang digunakan oleh client [OIK93].

2.3.2.1 Operator

Operator adalah sekelompok client yang memiliki hak akses khusus untuk dapat melakukan fungsi-fungsi umum pemeliharaan jaringan (*maintenance*). Operator harus dapat melakukan tugas-tugas mendasar jaringan seperti memutus koneksi dan melakukan koneksi ulang (*reconnecting*) terhadap server yang ditangani setiap kali diperlukan untuk mencegah terjadinya pemakaian routing jaringan yang buruk dalam jangka waktu yang lama.

2.3.3 Channel

Sebuah channel adalah sebuah kelompok yang memiliki nama tertentu yang terdiri dari satu atau lebih client yang semuanya akan dapat menerima setiap pesan yang dialamatkan ke kelompok (channel) tersebut. Channel otomatis diciptakan secara implisit ketika ada client yang memasukinya untuk pertama kali, begitu pula channel seakan menghilang ketika client yang terakhir kali meninggalkannya. Apabila channel belum diciptakan ketika memasukinya, maka channel akan otomatis terbentuk dan user yang menciptakannya kemudian menjadi seorang channel operator [OIK93].

2.3.3.1 Operator

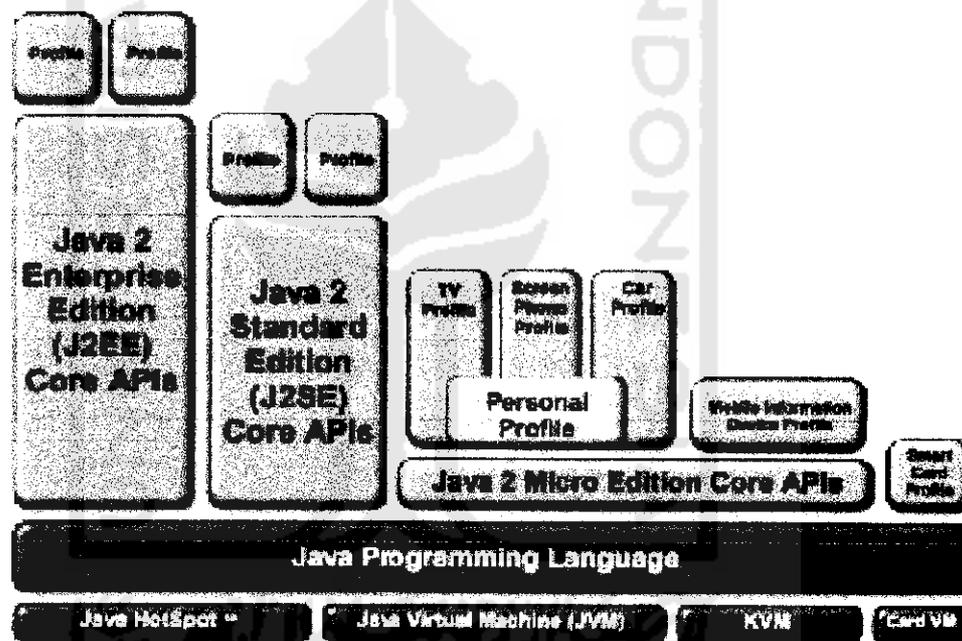
Channel operator (juga dinamakan “chop” atau “chanop”) pada sebuah channel memiliki posisi sebagai pemilik dari channel. Dalam hal ini channel operator memiliki kemampuan untuk menjaga dan mengendalikan aktivitas di dalam channel. Sebagai pemilik channel, seorang channel operator tidak diharuskan untuk memiliki alasan untuk setiap tindakan yang dilakukan, meskipun pada umumnya bersifat merugikan secara sosial kepada client-client di dalam channel.

Perintah-perintah yang sering digunakan oleh channel operator:

- a. KICK. Mengeluarkan seorang client dari channel.
- b. MODE. Mengubah mode channel.
- c. INVITE. Mengundang client ke channel (untuk channel dengan mode +i).
- d. TOPIC. Mengubah topik channel (mode +t).

2.4 Java

Java merupakan teknologi yang cukup fenomenal, dan mampu memberikan model pengembangan perangkat lunak yang inovatif. Bahasan mengenai teknologi Java akan selalu melibatkan dua bagian, yaitu Java sebagai bahasa pemrograman (*programming language*) dan Java sebagai lingkungan dijalankannya aplikasi (*platform*) [WIC02a].



Gambar 2.2 Arsitektur Java

2.4.1 Java Programming Language

Sebagai bahasa pemrograman, Java memiliki beberapa ciri khas yang menjadi keunggulan dibandingkan dengan bahasa pemrograman lain [WIC02a]:

1. *Simple*. Java dirancang untuk mudah dipelajari, dengan *syntax* yang sangat mirip dengan bahasa pemrograman C/C++ yang sudah sangat populer.

2. *Object-oriented*. Java merupakan bahasa pemrograman yang sangat konsisten dalam mengimplementasikan konsep OOP (*Object Oriented Programming*). Hal ini banyak memberikan keuntungan kepada para pengembang yang sudah mulai mengubah paradigmanya ke arah konsep OOP.
3. *Robust and secure*. Java menawarkan penggunaan sumber daya sistem operasi dengan lebih aman, karena dijalankan di atas Java platform, bukan di atas sistem operasi secara langsung. Selain itu Java juga menghilangkan operasi pointer yang sudah dikenal luas dalam bahasa C/C++. Sudah tidak terhitung banyaknya insiden keamanan seperti *web defacing* bahkan *host compromise* yang diakibatkan oleh kelemahan pemrograman yang dikenal dengan nama *buffer overflow*, yang disebabkan oleh kecerobohan dalam melakukan operasi pointer. Hal semacam itu tidak akan terjadi di aplikasi Java karena operasi pointer bahkan telah dihilangkan.
4. *Neutral architecture and portable*. Dengan semboyan *write once run anywhere*, Java dirancang untuk dapat dijalankan di semua platform, tanpa peduli apakah arsitekturnya berbasis Intel, AMD, Sun SPARC, atau PowerPC, aplikasi Java tetap mampu berjalan di berbagai platform tanpa perlu dikompilasi ulang.
5. *High performance*. Java dirancang untuk menghasilkan aplikasi-aplikasi dengan performansi yang terbaik. Contoh nyata aplikasi besar saat ini yang mungkin bisa dilihat adalah aplikasi database database Oracle 8i/9i yang *core*-nya dibangun menggunakan bahasa pemrograman Java.

2.4.2 Java Platform

Sebagai sebuah platform, Java terdiri atas dua bagian utama, yaitu *Java Virtual Machine* (JVM) dan *Java Application Programming Interface* (Java API) [WIC02a].

2.4.2.1 Java Virtual Machine (JVM)

Pada dasarnya, ada berbagai macam platform tempat aplikasi-aplikasi perangkat lunak dijalankan, seperti Windows, Unix, Linux, Netware, Macintosh, dan OS/2. Namun aplikasi-aplikasi yang berjalan pada suatu platform (misalnya Windows) tidak akan bisa dijalankan di platform yang lain (misalnya Linux) tanpa dilakukan perubahan kode dan proses kompilasi ulang.

Pada bahasa pemrograman lain (misalnya bahasa C), untuk beberapa program, seperti program Hello World atau Menghitung Faktorial, mungkin tidak perlu dilakukan modifikasi sama sekali, dan karena kebetulan *compiler*-nya banyak tersedia di berbagai platform, kode program dapat dikompilasi di platform masing-masing. Tetapi begitu kode pemrograman menyentuh bagian yang spesifik, seperti operasi I/O, komunikasi jaringan, atau multithreading, akan ada banyak keunikan pada masing-masing platform dalam menangani hal-hal tersebut, yang kemudian mengharuskan dilakukannya beberapa modifikasi kode program untuk masing-masing platform [PUR01].

Dalam hal ini, JVM berfungsi untuk menjalankan aplikasi-aplikasi Java yang telah dikompilasi menjadi kode biner Java (*byte-code*). Saat ini JVM tersedia secara luas untuk berbagai platform, yaitu antara lain Sun Solaris, Microsoft Windows, dan Linux.

2.4.2.2 Java Application Programming Interface (Java API)

Java API adalah sekumpulan pustaka dasar yang disediakan oleh vendor Java yang menjadi pondasi utama untuk melakukan pengembangan perangkat lunak berbasis teknologi Java. Pada setiap versi dan rilis, biasanya terdapat beberapa perbedaan pustaka serta adanya beberapa penyempurnaan implementasi Java API. Perkembangan Java API terutama dipengaruhi oleh permintaan dari para pengembang aplikasi yang menggunakan Java, kemudian juga adanya tuntutan perkembangan teknologi komputer yang ada.

2.5 J2ME

Platform Java 2 dibagi menjadi beberapa edisi oleh Sun Microsystems. Fungsionalitas utama yang memiliki dukungan minimal yang diperlukan untuk tiap lingkungan pengembangan Java dirilis dengan nama J2SE (*Java 2 Standard Edition*).

Sebagaimana yang terjadi pada sebagian besar perangkat lunak, kebutuhan sumber daya komputer untuk menjalankan Java semakin meningkat pada tiap rilis. Meskipun telah memiliki pangsa pasar pada produk-produk elektronik, J2SE semakin lama memerlukan semakin banyak memori dan kemampuan prosesor untuk tetap dapat memberikan solusi yang layak pakai. Ironisnya, ketika Sun mengembangkan Java untuk Internet dan pemrograman komersial, permintaan mulai berkembang untuk Java pada perangkat yang lebih kecil termasuk pada *smart-card*. Oleh karena itu, Sun kemudian menanggapi dengan menciptakan platform Java yang bersifat minimalis dalam fungsionalitasnya agar dapat berjalan

pada perangkat dengan sumber daya yang terbatas. Platform tersebut seluruhnya didasarkan pada JDK 1.1, yang merupakan teknologi pendahulu platform Java 2.

J2ME merupakan platform untuk *small devices* yang dimaksudkan untuk menggantikan produk-produk berbasis JDK1.1 dengan solusi berbasis Java 2. Tidak seperti pada lingkungan pengembangan desktop dan server yang menjadi target J2SE dan J2EE, lingkungan pengembangan mikro berisi perangkat yang sangat banyak dengan *capability* yang sangat berbeda sehingga tidak memungkinkan untuk menciptakan sebuah produk perangkat lunak yang cocok dengan semua produk tersebut. Subset dari lingkungan pemrograman Java secara keseluruhan didefinisikan oleh satu atau lebih *profile*, yang mengembangkan kapabilitas dasar dari sebuah *configuration* [TOP02].

2.5.1 J2ME Configuration

Dalam teknologi J2ME, J2ME Configuration berfungsi untuk mendefinisikan J2ME *runtime* (lingkungan kerja J2ME). Perlu diketahui bahwa setiap *handheld devices* memiliki fitur yang berbeda-beda. J2ME Configuration ini dirancang untuk menyediakan pustaka standar yang mengimplementasikan fitur standar dari sebuah *handheld devices*.

Saat ini telah ada dua kategori J2ME Configuration, yaitu [WIC02b]:

1. CLDC (*Connected Limited Device Configuration*)

Pada umumnya digunakan untuk aplikasi Java pada handphone seperti Nokia, Samsung, Motorola i85s, organiser dan PDA (*Personal Digital Assistant*). Umumnya perangkat-perangkat tersebut hanya memiliki memori berukuran 160-512 KiloBytes.

2. CDC (*Connected Device Configuration*)

Pada umumnya digunakan untuk aplikasi Java pada perangkat-perangkat *handheld devices* dengan ukuran memori paling tidak 2 MegaBytes.

Contohnya adalah Internet TV dan Nokia Communicator.

2.5.2 J2ME Profile

Konfigurasi standar yang disediakan oleh J2ME Configuration masih tergolong sangat umum dan masih memerlukan spesifikasi tambahan secara lebih detail. Contohnya adalah sebuah perangkat telepon seluler Nokia/Siemens/Sony Ericsson dapat digunakan untuk menelepon dan SMS. Hal ini sama-sama dapat dilakukan karena memang fitur-fitur tersebut adalah memang fitur standar sebuah telepon seluler [WIC02b].

Pada perkembangannya, perangkat Nokia dapat menyimpan dan memainkan file-file MP3 dan Real Media, yang mungkin tidak dapat dilakukan oleh sebagian perangkat dari vendor lain. J2ME Profile menyediakan implementasi-implementasi tambahan yang sangat spesifik dari sebuah *handheld devices*.

Saat ini ada lima kategori J2ME Profile, yaitu:

- a. *Mobile Information Device Profile* (MIDP)
- b. *Foundation Profile*
- c. *Personal Profile*
- d. *RMI Profile*
- e. *Personal Digital Assistance Profile*

2.5.3 MIDlet

MIDlet adalah aplikasi Java yang dijalankan pada lingkungan MIDP yang merupakan sebuah kelas dalam paket *java.microedition.midlet*. Setiap aplikasi MIDP yang dibuat harus merupakan penurunan (*extends*) dari kelas MIDlet.

2.5.3.1 MIDlet Suites

Sebuah MIDlet Suites adalah sekumpulan MIDlet yang dibungkus dalam sebuah file JAR (*Java Archives*). Setiap MIDlet dalam MIDlet Suites dapat melakukan pemakaian bersama (*sharing*) terhadap *class* dan *resources* yang ada dalam file JAR tersebut.

Sebuah file JAR terdiri dari komponen-komponen berikut:

- a. File *class* yang mengimplementasikan MIDlet.
- b. Berbagai file *resources* yang digunakan MIDlet, misalnya file-file *icon* atau *image*.
- c. Sebuah *manifest* yang mendeskripsikan isi file JAR.

Berikut ini adalah informasi atribut yang mendeskripsikan isi file JAR, yang harus ada dalam sebuah *manifest*:

- a. *MIDlet-Name*
- b. *MIDlet-Version*
- c. *MIDlet-Vendor*
- d. *MIDlet-<n>* untuk tiap MIDlet
- e. *MicroEdition-Profile*
- f. *MicroEdition-Configuration*

2.5.3.2 Tahap-tahap Pembuatan MIDlet

Proses pembuatan MIDlet terdiri dari beberapa tahapan, yaitu sebagai berikut:

1. Membuat MIDlet dan berbagai resources yang diperlukan.
2. Membuat MIDlet Suites.
3. Membuat Application Descriptor.
4. Menjalankan MIDlet di Emulator.
5. Melakukan perbaikan dan membuat ulang MIDlet (langkah 1-3).
6. Melakukan *deploy* MIDlet.

