

BAB II

TINJAUAN PUSTAKA

2.1 Landasan Teori

Landasan teori menjelaskan beberapa teori-teori dan penjelasan yang berkaitan dengan aplikasi atau media yang akan dibangun. Landasan teori yang digunakan dalam penyusunan aplikasi *Home Care Online* di Kabupaten Cianjur berbasis *Mobile* Android, meliputi Pengertian Aplikasi, android, Pengertian *Mobile*, GPS, API, *Google Maps*, *Java* Android, UML, *XAMPP*, *MySQL* dan PHP.

2.1.1 Home Care

Home Care adalah pelayanan kesehatan yang berkesinambungan dan komprehensif yang diberikan kepada individu dan keluarga ditempat tinggal mereka yang bertujuan untuk meningkatkan, mempertahankan atau memulihkan kesehatan atau memaksimalkan tingkat kemandirian dan meminimalkan akibat dari penyakit [6].

Home Care merupakan sebuah pelayanan yang sebenarnya telah ada sejak awal abad 19. Pada sejarah *Home Care*, semua kegiatan merawat pasien merupakan suatu kegiatan yang lazim dilakukan oleh masyarakat pada umumnya, karena pada abad tersebut semua orang masih menangani anggota keluarganya yang sakit itu sendiri, bahkan pada zaman tersebut keluarga atau tetangga dari keluarga yang anggota keluarganya meninggal melakukan persiapan penguburannya sendiri atau bergotong royong. Akhirnya pada era sekarang, setelah 1 abad lamanya perkembangan *home care* di dunia. Ruang lingkup *home care* dibagi 2 bagian. Bagian dalam lingkup *home care* antara lain, perawat kesehatan *home care* dengan tindakan dan Non-Medis *home care*. Dari 2 ruang lingkup ini, banyak orang yang masih bingung dengan perawatan kesehatan *home care* karena kurangnya informasi [6].

2.1.2 Perawat

Perawat adalah orang yang mengasuh dan merawat orang lain yang mengalami masalah kesehatan. Namun pada perkembangannya, pengertian perawat semakin meluas. Pada saat ini, pengertian perawat merujuk pada posisinya sebagai bagian dari tenaga kesehatan yang memberikan pelayanan kepada masyarakat secara profesional [10].

2.1.2.1 Kabupaten Cianjur

Sejarah mengenai awal mula Cianjur yang paling jelas adalah adanya hubungan dengan Mataram dan *Vereeninging Oos Indische Compagnie* (VOC) karena hubungan dengan kedua belah pihak itulah mulai tercatat adanya suatu wilayah yang dinamakan Cianjur. Catatan sejarah awal tersebut berupa surat-surat perjanjian VOC. Dalam hal sejarah seperti halnya orang *Eropa* meninggalkan catatan, tradisi yang ada dalam masyarakat Indonesia pada umumnya adalah tradisi lisan. Cianjur sendiri pada dasarnya merupakan nama sebuah sungai yang terletak di wilayah pemukiman Wiratanu II. Wiratanu II adalah anak ke lima dari Wiratanu I. Dalam perkembangan Islam di Jawa Barat muncul fenomena tumbuhnya lembaga-lembaga pendidikan pesantren. Demikian pula kehidupan keagamaan di Cianjur, tidak terlepas dari peranan pesantren. Bahkan dapat dikatakan kehidupan pesantren di Cianjur lebih menonjol apabila dibandingkan dengan kabupaten-kabupaten lain di priangan.

2.2 Android

Android adalah sebuah sistem operasi untuk perangkat *mobile* berbasis linux yang mencakup sistem operasi, *middleware* dan aplikasi [3] dan android adalah sistem operasi untuk telepon selular yang berbasis linux. Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri sehingga dapat digunakan oleh bermacam peranti penggerak.

Berdasarkan pemaparan pendapat dapat diambil kesimpulan, android adalah suatu sistem operasi untuk *mobile* berbasis *linux* yang bersifat *open source* sehingga memberikan pengembangan untuk melakukan pengembangan sesuai

dengan yang diharapkan sehingga sangat cepat perkembangan sistem operasi ini [3].

2.2.1 Sejarah Android

Awalnya, *Google Inc.* Membeli *Android Inc.* Yang merupakan pendatang baru yang membuat piranti lunak untuk ponsel/*smartphone*. Untuk mengembangkann Android, dibentuklah *Open Handset Alliance*, yaitu gabungan dari 34 perusahaan piranti keras, piranti lunak, dan telekomunikasi. Pada saat perilisan Android, 5 November 2007, Android bersama *Open Handset Alliance* menyatakan mendukung pengembangan *open source* pada perangkat *mobile*. Di pihak lain, *Google* merilis kode-kode Android di bawah lisensi *Apache*, sebuah lisensi persangkat lunak dan *open platform* perangkat seluler. Terdapat dua jenis distributor sistem operasi Android. Pertama yang mendapatkan dukungan penuh *Google* atau *Google Mail Service* (GMS) dan kedua adalah yang benar-benar bebas distribusinya tanpa dukungan langsung dari *Google* atau dikenal sebagai *Open Handset Distributor* (OHD) [3].

Saat ini sebagian besar vendor-vendor *smartphone* sudah memproduksi *smartphone* berbasis android. Tidak hanya menjadi sistem operasi di *smartphone*, pada saat ini Android menjadi pesaing utama dari *Apple* pada sistem operasi Tablet PC. Pesatnya pertumbuhan Android dikarenakan Android itu sendiri adalah *platform* yang sangat lengkap baik itu dari sistem operasi, aplikasi, alat pengembangan, serta dukungan yang sangat tinggi dari komunitas *open source* di dunia. Berikut ini merupakan perkembangan OS Android sampai saat ini.

Tabel 2.1 Perkembangan Sistem Operasi Android [3]

Sistem Operasi	Keterangan
Android Beta	Pertama kali dirilis pada tanggal 5 November 2007, kemudian pada 12 November 2007 <i>Software Development Kit</i> (SDK) dirilis oleh <i>Google</i> .
Android 1.0 (<i>Astro</i>)	Pertama kali dirilis pada 23 September 2008. Sebenarnya Android versi pertama ini akan dinamai dengan nama

Sistem Operasi	Keterangan
	"Astro" tapi karena alasan hak cipta dan <i>trademark</i> nama "Astro" tidak jadi disematkan pada versi pertama dari OS Android ini. HTC <i>Dream</i> adalah ponsel pertama yang menggunakan OS ini.
Android 1.1 (<i>Bender</i>)	Versi ini dirilis pada 9 Maret 2009, yang muncul dengan pembaruan terhadap estetika pada aplikasi, jam alarm, <i>voice search</i> (pencarian suara), pengiriman pesan dengan <i>Gmail</i> , dan pemberitahuan <i>email</i> . Namun versi Android kedua ini juga mengalami masalah penamaan yang sama dengan versi pertamanya.
Android 1.5 (<i>Cupcake</i>)	Pertama kali dirilis pada 30 April 2009. <i>Cupcake</i> atau dalam terjemahan bebas berarti kue cawan. Mulai versi Android ini penamaan menggunakan nama makan pencuci mulut (<i>dessert</i>) mulai digunakan, karena ini merupakan versi yang ketiga maka penamaan diawali dengan huruf "C" dan jadilah " <i>Cupcake</i> " menjadi nama resmi dari versi OS Android ketiga ini.
Android 1.6 (<i>Donut</i>)	Versi ini dirilis pada September 2009. <i>Donut</i> dikenal juga dengan kue berlubang alias dari versi Android 1.6.
Android 2.0/2.1 (<i>Enclair</i>)	Versi ini dirilis pada 9 Desember 2009. <i>Eclair</i> adalah kue sus yang berbentuk panjang dengan topping coklat di atasnya.
Android 2.2 (<i>Froyo</i>)	Versi ini dirilis pertamakali pada 20 Mei 2010. <i>Froyo</i> (<i>Frozen yoghurt</i>) adalah <i>yoghurt</i> (susu yang dibuat melalui fermentasi bakteri) yang dibekukan mirip seperti <i>ice cream</i> . <i>Froyo</i> dipakai sebagai nama alias dari sistem operasi Android versi 2.2.
Android 2.3 (<i>Gingerbread</i>)	Versi ini dirilis pada 6 Desember 2010. <i>Gingerbread</i> adalah kue yang terbuat dari jahe, biasanya berbentuk

Sistem Operasi	Keterangan
	boneka sering disajikan sebagai teman minum kopi. <i>Gingerbread</i> dipakai sebagai nama alias dari sistem operasi Android versi 2.3
Android 3.0 (<i>Honeycomb</i>)	Versi ini dirilis pada 22 Februari 2011. <i>Honeycomb</i> atau sarang madu adalah bagian dari hasil lebah yang dimanfaatkan selain dari madunya itu sendiri.
Android 4.0 (<i>Ice Cream Sandwich</i>)	Versi dirilis pada 19 Oktober 2011. <i>Ice Cream</i> , tentu saja kita tahu karena ini adalah minuman atau tepatnya makanan yang sangat disukai terutama oleh anak kecil. <i>Ice Cream</i> dipakai sebagai nama alias dari Android versi 4.0. Smartphone yang pertama kali menggunakan OS Android ini adalah Samsung Galaxy Nexus.
Android 4.1 (<i>Jelly Bean</i>)	Android <i>Jelly Bean</i> yang diluncurkan pada acara <i>Google I/O</i> lalu membawa sejumlah keunggulan dan fitur baru.
Android 4.4 (<i>Kitkat</i>)	Dirilis pada 14 oktober 2013 lalu . dengan mengusung konsep sebuah coklat yang sering disajikan untuk para karyawan nya di kantor <i>google</i> . Muncul sebuah ide untuk mengusung nama “ <i>KitKat</i> ” menjadi nama OS android terbaru setelah 4.1.
Android 5.0/5.1 (<i>Lollipop</i>)	Diresmikan pada 25 Juni 2014 saat <i>Google I / O</i> , dan tersedia secara resmi melalui <i>over-the-air (OTA) update</i> pada tanggal 12 November 2014, untuk memilih perangkat yang menjalankan distribusi Android dilayani oleh <i>Google</i> (seperti perangkat Nexus dan <i>Google Play edition</i>).
Android 6.0 (<i>Marshmallow</i>)	Pratayang pengembang Android "M" dirilis pada tanggal 28 Mei 2015, untuk telepon genggam Nexus 5 dan Nexus 6, tablet Nexus 9, dan <i>set-top box</i> Nexus <i>Player</i> , di bawah nomor bentukan MPZ44Q. Pratayang pengembang ketiga

Sistem Operasi	Keterangan
	(MPA44G) dirilis pada tanggal 17 Agustus 2015 untuk Nexus 5, Nexus 6, Nexus 9 dan perangkat Nexus <i>Player</i> , dan diperbarui ke MPA44I yang membawa perbaikan yang berhubungan dengan profil <i>Android for Work</i> .

2.2.2 Arsitektur Android

Secara garis besar Arsitektur Android memiliki 5 lapisan, yaitu *Application and Widget*, *Application Frameworks*, *Libraries*, *Android Run Time*, dan *Linux Kernal* [3].

1. *Application and Widget*

Application and Widget ini adalah lapisan dimana kita berhubungan dengan aplikasi saja, di mana biasanya kita mengunduh aplikasi kemudian menginstal dan menjalankan aplikasi tersebut.

2. *Application Framework*

Application Framework ini adalah lapisan di mana para pembuat aplikasi melakukan pengembangan/pembuatan aplikasi yang akan dijalankan di sistem operasi android, karena pada lapisan inilah aplikasi dapat dirancangkan dan dibuat.

3. *Libraries*

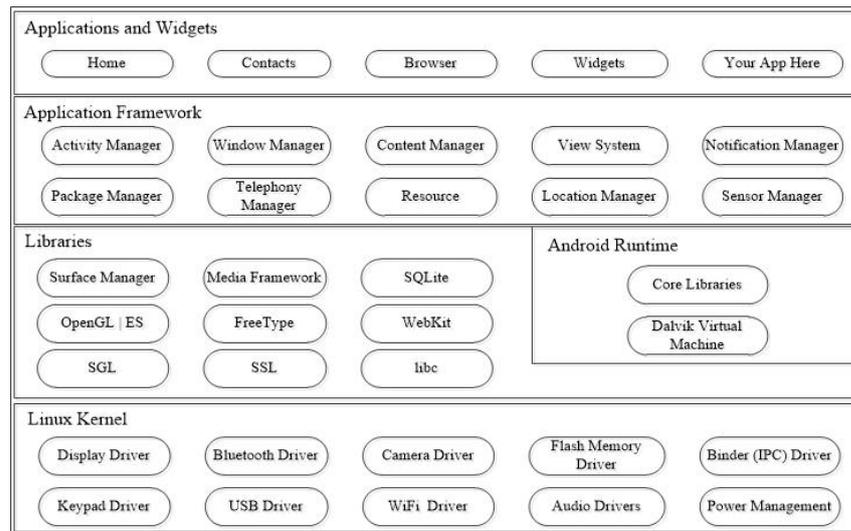
Libraries ini adalah lapisan di mana fitur-fitur android berada, biasanya para pembuat aplikasi mengakses *libraries* untuk menjalankan aplikasinya.

4. *Android Run Time*

Lapisan ini yang membuat aplikasi android dapat dijalankan di mana dalam prosesnya menggunakan implementasi linux.

5. *Linux Kernel*

Linux kernel ini adalah lapisan di mana inti dari sistem operasi android itu berada. Berisi file-file sistem yang mengatur sistem *processing*, *memory*, *resource*, *driver*, dan sistem-sistem operasi android lainnya.



Gambar 2.1 Arsitektur Android [3].

2.3 Pengertian *Mobile*

Aplikasi adalah program yang digunakan orang untuk melakukan sesuatu pada sistem komputer. *Mobile* dapat diartikan sebagai perpindahan yang mudah dari suatu tempat ke tempat yang lain, misalnya telepon *mobile* berarti bahwa terminal telepon yang dapat berpindah dengan mudah dari suatu tempat ke tempat lain tanpa terjadi pemutusan atau terputusnya komunikasi. Sistem aplikasi *mobile* merupakan aplikasi yang dapat digunakan walaupun pengguna berpindah dengan mudah dari suatu tempat ke tempat lain, tanpa terjadi pemutusan atau terputusnya komunikasi. Aplikasi ini dapat diakses melalui perangkat nirkabel seperti pager, seperti telepon seluler dan PDA.

Adapun karakteristik dari *mobile* adalah :

1. Ukuran yang kecil

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

2. Memori yang terbatas

Perangkat *mobile* juga memiliki *memory* yang terkecil, yaitu *primary* (RAM), dan sekunder (Disk).

3. Daya proses yang terbatas

Sistem *mobile* tidak setangguh desktop.

4. Mengonsumsi daya yang rendah
Perangkat *mobile* menghabiskan sedikit daya dibandingkan dengan mesin desktop.
5. Kuat dan dapat diandalkan
Karena perangkat *mobile* selalu dibawa ke mana saja, mereka harus cukup kuat untuk menghadapi benturan, gerakan, dan sesekali terkena tetesan air.
6. Konektivitas yang terbatas
Perangkat *mobile* ini memiliki *bandwidth* rendah, beberapa dari mereka kadang tidak tersambung.
7. Masa hidup yang pendek.
8. Perangkat – perangkat konsumen ini menyala dalam hitungan detik kebanyakan dari mereka selalu menyala.

2.4 Global Positioning System (GPS)

Global Positioning System (GPS) adalah suatu sistem navigasi yang memanfaatkan satelit. Penerima GPS memperoleh sinyal dari beberapa satelit yang mengorbit bumi. Satelit yang mengitari bumi pada orbit pendek ini terdiri dari 24 susunan satelit, dengan 21 satelit aktif dan 3 buah satelit sebagai cadangan. Dengan susunan orbit tertentu, maka satelit GPS bisa diterima di seluruh permukaan bumi dengan penampakan antara 4 sampai 8 buah satelit. GPS dapat memberikan informasi posisi dan waktu dengan ketelitian sangat tinggi.

Nama lengkap GPS adalah NAVSTAR GPS (*Navigational satellite Timing and Ranging Global Positioning System*), namun lebih sering dikenal sebagai GPS. GPS mulai diaktifkan untuk umum pada 17 Juli 1995. Sedangkan, *Assisted-Global Positioning System* (A-GPS) merupakan penyempurnaan dari GPS sebagai satelit penentu posisi di belahan bumi. Satelit GPS yang dimiliki bumi mempunyai konstelasi 24 satelit dalam enam orbit yang mendekati lingkaran, setiap orbit ditempati oleh 4 buah satelit dengan *interval* antara yang tidak sama. Orbit satelit GPS berinklinasi 55° terhadap bidang *equator* dengan ketinggian rata-rata dari permukaan bumi sekitar 20.200 km.

Prinsip kerja dari GPS adalah Satelit GPS memberikan informasi kepada *receiver GPS* mengenai jarak atau posisi satelit. Sehingga kita tahu bahwa kita berada pada suatu radius tertentu dari satelit. Bila ada dua satelit maka kita tahu posisi kita, berada pada 2 lokasi, yaitu perpotongan dua radius tadi. *GPS receiver* mampu menghitung tempat yg paling mungkin. Semakin banyak sinyal satelit ditangkap semakin teliti satelit menghitung posisi.

2.5 *Google Maps*

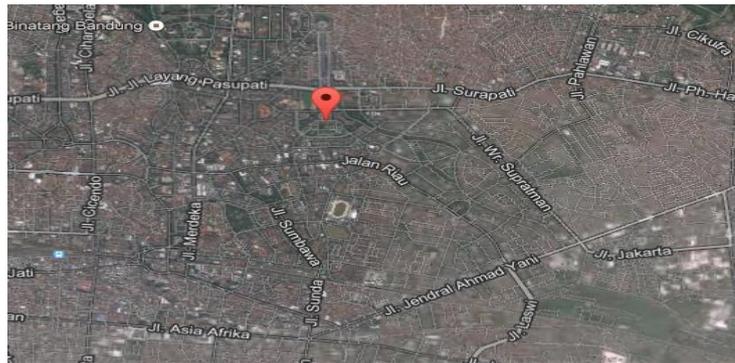
Google maps merupakan salah satu layanan yang dikeluarkan oleh Google. *Google Maps* menyediakan layanan berbasis peta yang sangat responsif dan mudah dalam penggunaannya. Dengan menggunakan *google maps*, pengguna dapat dengan mudah mencari suatu lokasi serta dapat melakukan penelusuran rute menuju lokasi yang diinginkan. Ditingkat pemrograman, *Google maps* dapat dikembangkan dengan basis data, semua data yang terkait dengan titik lokasi disimpan dalam tabel dan dapat ditampilkan sesuai keinginan pengguna. Isi tabel yang berisi data posisi peta dapat ditampilkan Dengan menyajikan informasi lokasi yang menggunakan *google maps*. Sementara ini *Google Maps* memiliki dua media untuk menampilkannya diantaranya adalah sebagai berikut :

1. *Maps* menampilkan peta dalam bentuk garis contohnya dapat dilihat pada gambar :



Gambar 2.2 *Maps* yang menampilkan Peta

2. Satelit yang dapat menampilkan peta dalam bentuk citra satelit seperti yang di tunjukan pada gambar :



Gambar 2.3 Maps dalam bentuk Citra Satelit

Sampai saat ini *Google Maps* masih berada dalam tahap beta, dan masih terus dikembangkan dengan data yang selalu diperbarui secara berkala. *Google maps* merupakan hak cipta Google secara *propriety*, sehingga dalam menggunakannya memerlukan adanya perjanjian, Google membuat mekanisme untuk dapat mengakses google maps dengan koding aplikasi dan dengan sebuah kunci yang dikenal dengan API Key.

2.6 API

API adalah antarmuka yang digunakan untuk mengakses aplikasi atau layanan dari sebuah program. API memungkinkan pengembang untuk memakai fungsi yang sudah ada dari aplikasi lain sehingga tidak perlu membuat ulang dari awal. Pada konteks *web*, API merupakan pemanggilan fungsi lewat *Hyper Text Transfer Protocol* (HTTP) dan mendapatkan respon berupa *Extensible Markup Language* (XML) atau *JavaScript Object Notation* (JSON). Pemanggilan fungsi ke suatu situs tertentu akan menghasilkan respon yang berbeda kepada pengguna untuk membangun aplikasi enterprise di dalam websitenya.

Dengan API, panggilan-panggilan yang bolak-balik antaraplikasi diatur melalui *web service*. *Web service* adalah kumpulan standar teknis dan protokol, termasuk XML (*Extensible Markup Language*), bahasa umum yang digunakan oleh aplikasi-aplikasi tersebut selama berkomunikasi di internet. API sendiri merupakan sekumpulan kode *software* yang ditulis sebagai serangkaian pesan XML. Setiap pesan XML berhubungan dengan fungsi spesifik dari aplikasi yang akan diajak berkomunikasi.

Pengembang aplikasi pihak ketiga menggunakan pesan-pesan *XML* yang berhubungan dengan fungsi-fungsi spesifik dari layanan *web* yang akan diajak berkomunikasi. Pengembang bebas memilih fungsi khusus apa saja yang akan diajak berkomunikasi, dan ditampilkan pada aplikasi rancangannya.

API dan *web service* sepenuhnya bekerja di belakang layar. Para peselancar web dan pengguna *software* sama sekali tidak melihatnya. Keduanya bekerja diam-diam, menyediakan jalan bagi beberapa aplikasi untuk bekerja sama untuk memberikan informasi atau fungsionalitas yang dibutuhkan oleh pengguna aplikasi.

Dengan demikian, API adalah standar komunikasi yang dibuka oleh perusahaan *software*, agar dapat dimanfaatkan oleh pengembang pihak ketiga untuk mendesain aplikasi yang memanfaatkan layanan mereka dengan mudah.

2.7 Java

Java adalah salah satu bahasa pemrograman yang paling populer. *Java* dapat digunakan untuk berbagai hal, termasuk pengembangan perangkat lunak, aplikasi mobile, dan pengembangan sistem yang besar. Inilah yang membuat bahasa pemrograman Java sangat terkenal di lingkungan pengembang perangkat lunak.

Seperti bahasa pemrograman lain, bahasa Java memiliki struktur sendiri, aturan sintaks, dan paradigma pemrograman. paradigma pemrograman bahasa Java didasarkan pada konsep OOP. Bahasa Java merupakan turunan bahasa C, sehingga aturan sintaks yang terlihat akan seperti bahasa C. Misalnya, blok kode yang modular dalam metode dan dibatasi oleh karakter ‘{‘ dan ‘}’, dan variabel dideklarasikan sebelum digunakan. Secara struktural, bahasa *Java* diatur dengan *package*. Di Dalam *package* ada *class*, dan dalam *class* ada *method*, variabel, konstanta, dan banyak lagi [4].

2.8 PHP

PHP merupakan singkatan dari “*Hypertext Preprocessor*”. Pada awalnya PHP merupakan kependekan dari *Personal Home Page* (situs personal) dan PHP itu sendiri pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995, dan pada saat itu PHP masih bernama FI (*Form Interpreter*), yang wujudnya berupa sekumpulan script yang digunakan untuk mengolah data form dari web. PHP adalah sebuah bahasa scripting yang terpasang pada HTML. Sebagai besar sintaknya mirip dengan bahasa pemrograman C, Java, ASP, dan Perl, ditambah beberapa fungsi PHP yang spesifikasi dan mudah dimengerti.

PHP digunakan untuk membuat tampilan web menjadi lebih dinamis, dengan *PHP* anda dapat menampilkan atau menjalankan beberapa *file* dalam 1 *file* dengan cara di *include* atau *require*. PHP itu sendiri sudah dapat berinteraksi dengan beberapa database walaupun dengan kelengkapan yang berbeda, yaitu: DBM, *FilePro* (*Personic, Inc*), *Informix*, *Ingres*, *Interbase*, *Microsoft Acces*, *MSSQL*, *MySQL*, *Oracle*, *PostgreSQL*, dan *Sybase* [5].

Dari uraian di atas maka dapat diambil 4 point utama tentang PHP :

- PHP adalah singkatan dari : *Hypertrxt Preprocessor*.
- PHP adalah bahasa scripting server-side, artinya dijalankan di server, kemudian outputnya dikirim ke *client* (*browser*).
- PHP digunakan untuk membuat aplikasi *web*.
- PHP mendukung banyak database (*MySQL*, *Informix*, *Oracle*, *Sybase*, *Solid*, *PostgreSQL*, *Generic ODBC*, dll).

2.9 UML (*Unified Modelling Language*)

UML (*Unified Modeling Language*) adalah salah satu alat bantu yang sangat handal di dunia pengembangan sistem yang berorientasi objek. Hal ini disebabkan karena UML menyediakan bahasa pemodelan visual yang memungkinkan bagi pengembang sistem untuk membuat cetak biru atas visi mereka dalam bentuk yang baku, mudah dimengerti serta dilengkapi dengan kanisme yang efektif untuk berbagi (*sharing*) dan mengkomunikasikan rancangan mereka dengan yang lain.

UML merupakan kesatuan dari bahasa pemodelan yang dikembangkan oleh Booch, *Object Modeling Technique* (OMT) dan *Object Oriented Software Engineering* (OOSE). Metode Booch dari Grandy Booch sangat terkenal dengan nama metode *Design Object Oriented*. Metode ini menjadikan proses analisis dan design ke dalam empat tahapan interatif, yaitu: identifikasi kelas-kelas dan obyek-obyek, identifikasi semantik dari hubungan obyek dan kelas tersebut, perincian interface dan implementasi. Keunggulan metode Booch adalah pada *detail* dan kayanya dengan notasi dan elemen. Pemodelan OMT yang dikembangkan oleh Umbaugh didasarkan pada analisis terstruktur dan pemodelan *entity-relationship*. Tahapan utama dalam metodologi ini adalah analisis, *design* sistem, *design* obyek dan implementasi. Keunggulan model ini adalah penotasian yang mendukung semua konsep OO. Metode OOSE dari Jacobson lebih memberikan penekanan pada *use case*. OOSE memiliki tiga tahap yaitu membuat model *requirement* dan analisis, *design* dan implementasi, dan pengujian (*test* model). Keunggulan metode ini adalah mudah dipelajari karena memiliki notasi yang sederhana namun mencakup seluruh tahapan rekayasa perangkat lunak [2].

Terdapat diagram-diagram UML yang paling sering digunakan dalam memodelkan sebuah sistem berorientasi objek diantaranya :

1. Use Case Diagram

Use case diagram adalah deskripsi fungsi dari sebuah *system* dari perspektif pengguna. *Use case* bekerja dengan cara mendeskripsikan tipikal interaksi antara *user* (pengguna) sebuah *system* dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah *system* dipakai. Urutan langkah-langkah yang menerangkan antara pengguna disebut *scenario*. Setiap *scenario* mendeskripsikan urutan kejadian. Setiap urutan diinisialisasi oleh orang, *system* yang lain, perangkat keras atau urutan waktu. Dengan demikian secara singkat bisa dikatakan *use case* adalah serangkaian *scenario* yang digabungkan bersama-sama oleh tujuan umum pengguna.

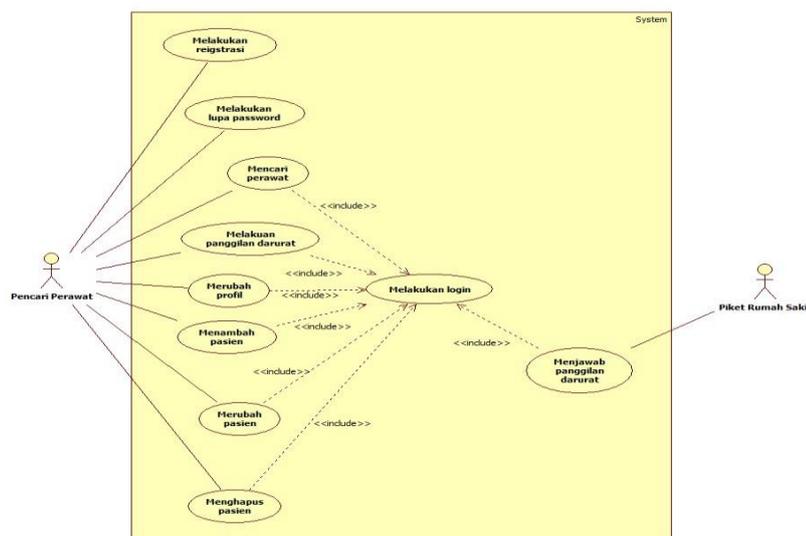
Dalam pembicaraan tentang *use case*, pengguna biasanya disebut dengan *actor*. *Actor* adalah sebuah peran yang bisa dimainkan oleh pengguna dalam interaksinya dengan *system*.

Model *use case* adalah bagian dari model *requirement* (Jacobson et al, 1992). Termasuk disini adalah problem domain object model dan penjelasan tentang user interface. *Use case* memberikan spesifikasi fungsi-fungsi yang ditawarkan oleh sistem dari perspektif user.

Use Case adalah alat bantu terbaik guna menstimulasi pengguna potensial untuk mengatakan tentang suatu sistem dari sudut pandangnya. Tidak selalu mudah bagi pengguna untuk menyatakan bagaimana mereka bermaksud menggunakan sebuah *system*. Karena *system* pengembangan tradisional seringkali ceroboh dalam melakukan analisis, akibatnya pengguna seringkali susah menjawabnya tatkala dimintai masukan tentang sesuatu.

Ide dasarnya adalah bagaimana melibatkan penggunaan *system* di fase-fase awal analisis dan perancangan *system*. Dengan demikian diharapkan akan bisa dibangun suatu *system* yang bisa membantu pengguna. Perlu diingat bahwa *use case* mewakili pandangan di luar *system*.

Use case dibuat berdasarkan keperluan *actor*. *Use case* harus merupakan ‘apa’ yang dikerjakan *software* aplikasi, bukan ‘bagaimana’ *software* aplikasi mengerjakannya. Setiap *use case* harus diberi nama yang menyatakan apa hal dicapai dari hasil interaksinya dengan *actor*. Nama *use case* boleh berdiri dari beberapa kata dan tidak boleh ada dua *use case* yang memiliki nama yang sama [2]. Berikut adalah contoh *Use Case Diagram* :



Gambar 2.4 Contoh Use Case Diagram

2. Class Diagram

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan *design* berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi).

Class diagram menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain.

Class memiliki tiga area pokok :

1. Nama memiliki fungsi merepresentasikan identitas dari sebuah *class*.
2. Atribut memiliki fungsi merepresentasikan atribut – atribut yang ada di dalam suatu *class*.
3. Metoda memiliki fungsi merepresentasikan metode-metode yang ada pada suatu *class*.

Atribut dan metoda dapat memiliki salah satu sifat berikut :

1. *Private*, tidak dapat dipanggil dari luar *class* yang bersangkutan.
2. *Protected*, hanya dapat dipanggil oleh *class* yang bersangkutan dan anak-anak yang mewarisinya.
3. *Public*, dapat dipanggil oleh siapa saja.

Setiap *class* pasti memiliki hubungan dengan *class* lain. Terdapat 4 cara menghubungkan satu *class* dengan *class* lain tergantung hubungan *class – class* tersebut. Hubungan tersebut direpresentasikan dengan sebuah panah diantara *class* tersebut. Adapun 4 hubungan antar *class* pada *class diagram* sebagai berikut :

1. Asosiasi

Asosiasi adalah hubungan statis antar *class*. Umumnya menggambarkan *class* yang memiliki atribut berupa *class* lain, atau *class* yang harus mengetahui eksistensi *class* lain. Panah *navigability* menunjukkan arah *query* antar *class*.

2. Agregasi

Agregasi adalah hubungan yang menyatakan bagian “terdiri atas”.

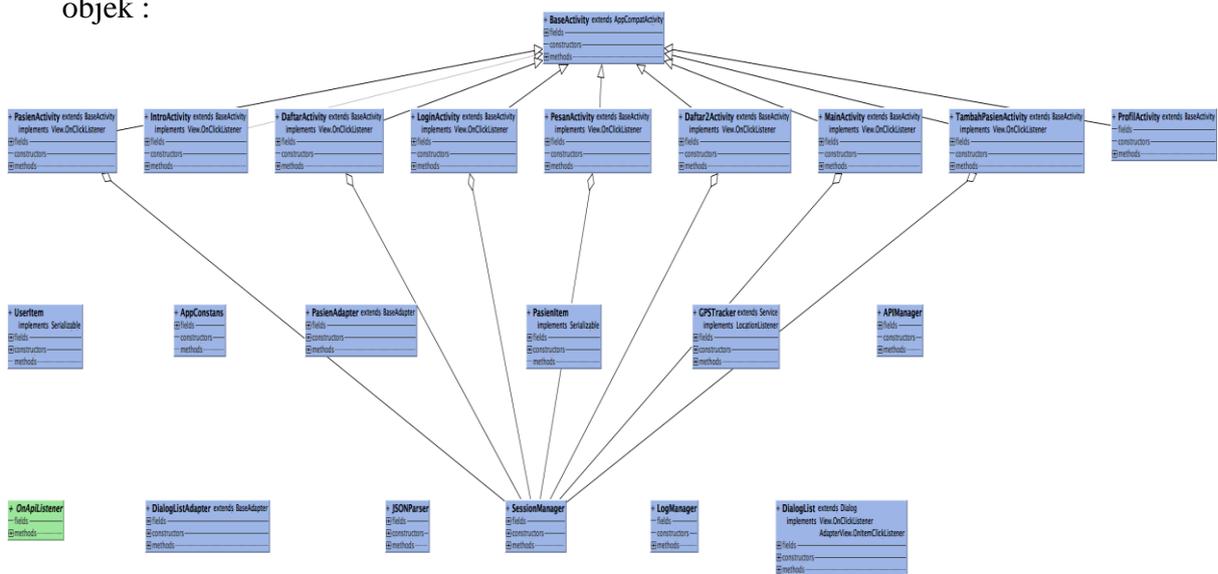
3. Pewarisan

Pewarisan adalah hubungan hirarki antar *class*. *Class* dapat diturunkan dari *class* lain dan mewarisi semua atribut dan metoda *class* asalnya dan menambahkan fungsionalitas baru sehingga ia disebut anak dari *class* yang diwarisinya. Kebalikan dari pewarisan adalah generalisasi.

4. Hubungan dinamis

Hubungan dinamis adalah rangkaian pesan (*message*) yang di-*passing* dari satu *class* kepada *class* lain. Hubungan dinamis dapat digambarkan dengan menggunakan *sequence diagram* yang akan dijelaskan kemudian.

Berikut adalah contoh dari *class diagram* pada sebuah pemodelan berbasis objek :



Gambar 2.5 Contoh *Class Diagram*

3. Activity Diagram

Activity Diagram adalah teknik untuk mendiskripsikan logika prosedural, proses bisnis dan aliran kerja dalam *flowchart* akan tetapi perbedaannya dengan *flowchart* adalah *activity diagram* bisa mendukung perilaku paralel sedangkan *flowchart* tidak bisa [2].

Activity diagram merupakan *state diagram* khusus, di mana sebagian besar *state* adalah *action* dan sebagian besar transisi di-*trigger* oleh selesainya *state* sebelumnya (*internal processing*). Oleh karena itu *activity diagram* tidak

menggambarkan behaviour internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum.

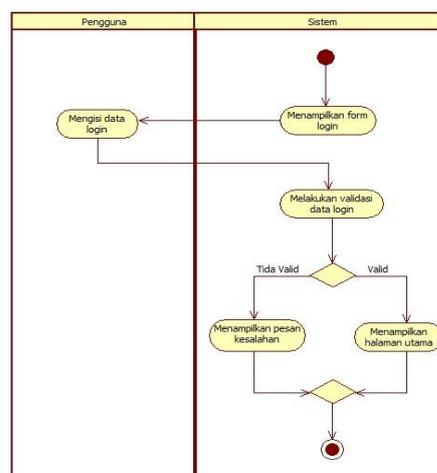
Sebuah aktivitas dapat direalisasikan oleh satu *use case* atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara *use case* menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas.

Standar UML menggunakan segiempat dengan sudut membulat untuk menggambarkan aktivitas. *Decision* digunakan untuk menggambarkan *behaviour* pada kondisi tertentu. Untuk mengilustrasikan proses-proses paralel (*fork* dan *join*) digunakan titik sinkronisasi yang dapat berupa titik, garis horizontal atau vertikal.

Activity diagram dapat dibagi menjadi beberapa *object swimlane* untuk menggambarkan objek mana yang bertanggung jawab untuk aktivitas tertentu.

Activity Diagram seperti sebuah *flow chart*. *Activity* Diagram menunjukkan tahapan, pengambilan keputusan dan percabangan. Diagram ini sangat berguna untuk menunjukkan *operation* sebuah obyek dan proses bisnis. Kelebihan *activity* diagram dibandingkan *flow chart* adalah kemampuannya dalam menampilkan aktivitas paralel.

Activity Diagram bisa digunakan untuk menunjukkan siapa mengerjakan apa dengan teknik *partition* [2]. Berikut adalah contoh dari *activity* diagram :



Gambar 2.6 Contoh Activity Diagram

4. *Sequence Diagram*

Sequence diagram digunakan untuk menggambarkan perilaku pada sebuah *scenario*. Diagram ini menunjukkan sejumlah contoh obyek dan *message* (pesan) yang diletakkan diantara obyek-obyek ini di dalam *use case*.

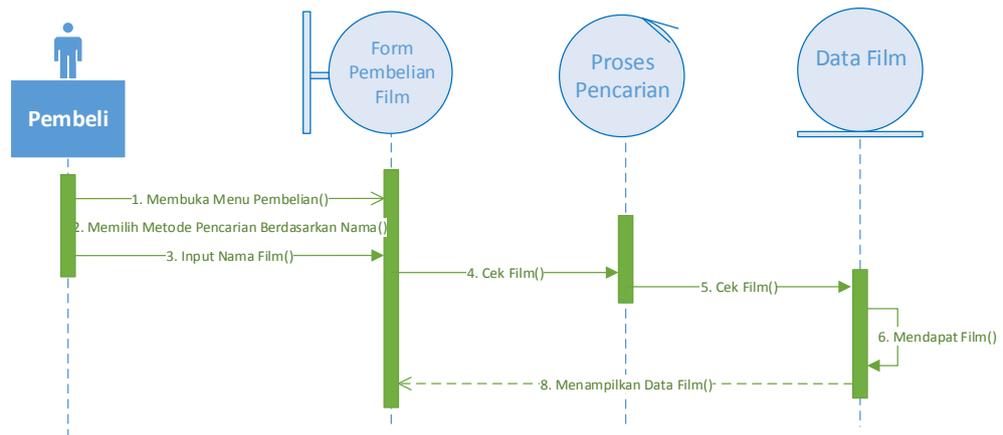
Komponen utama *sequence* diagram terdiri atas obyek yang dituliskan dengan kotak segiempat bernama. *Message* diwakili oleh garis dengan tanda panah dan waktu yang ditunjukkan dengan progress vertical [2] (Munawar, Bab 9).

Sequence diagram biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai *respons* dari sebuah *event* untuk menghasilkan *output* tertentu. Diawali dari apa yang men-*trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara *internal* dan *output* apa yang dihasilkan. Masing-masing objek, termasuk aktor, memiliki *lifeline* vertikal. *Message* digambarkan sebagai garis berpanah dari satu objek ke objek lainnya. Pada fase desain berikutnya, *message* akan dipetakan menjadi operasi/metoda dari *class*. *Activation bar* menunjukkan lamanya eksekusi sebuah proses, biasanya diawali dengan diterimanya sebuah *message*. Untuk objek-objek yang memiliki sifat khusus, standar UML mendefinisikan *icon* khusus untuk objek *boundary*, *controller* dan *persistent entity*.

Sequence Diagram menambahkan dimensi waktu pada interaksi diantara obyek. Pada diagram ini *participant* diletakkan di atas dan waktu ditunjukkan dari atas ke bawah. *Life line participant* diurutkan dari setiap *participant*. Kotak kecil pada *lifeline* menyatakan *activation* yaitu, menjalankan salah satu *operation* dari *participant*. State bisa ditambahkan dengan menempatkannya sepanjang *life line*.

Message (sederhana, *synchronous*) adalah tanda panah yang menghubungkan suatu *life line* ke *life line* yang lain. Lokasi *life line* dalam dimensi vertikal mewakili urutan waktu dalam *sequence* diagram. *Message* yang pertama terjadi adalah yang paling dekat dengan bagian atas diagram dan yang terjadi belakangan adalah yang terdekat dengan bagian bawah.

Pada beberapa sistem, operasi bisa dilakukan kepada dirinya sendiri. Hal ini diaebut dengan rekursi. Untuk melukiskannya digunakan anak panah dari *activation* kembali ke dirinya sendiri, dan sebuah kotak kecil diletakkan pada bagian atas dari *activation* [2]. Berikut adalah contoh dari *Sequence Diagram* :



Gambar 2.7 Contoh *Sequence Diagram*

2.10 MySQL

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (*Database Management System*) atau DBMS yang *multithread*, *multi-user*, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL AB (perusahaan asal Finlandia) membuat MySQL sebagai perangkat lunak gratis dibawah lisensi *GNU General Public License* (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL. *Relational Database Management System* (RDBMS). MySQL merupakan *Relational Database Management System* (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (*General Public License*) dimana setiap orang bebas untuk menggunakan MySQL, namun tidak boleh dijadikan produk turunan yang bersifat komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam database sejak lama, yaitu SQL (*Structured Query Language*).

Dalam penggunaan database MySQL, setiap perintah yang diketikkan disebut *query*. Perintah MySQL dapat dikategorikan menjadi 3 sub perintah, yaitu DDL (*Data Definition Language*), DML (*Data Manipulation Language*), dan DCL (*Data Control Language*).

1. DDL (*Data Definition Language*)

Perintah dalam SQL yang pertama adalah perintah DDL. DDL sendiri merupakan kependekan dari apa yang dikenal dengan nama *Data Definition Language*. Apabila diartikan secara harafiah, *Data Definition Language* berarti merupakan sebuah bahasa SQL yang digunakan untuk mendefinisikan suatu data. Secara teoritis, DDL dapat berarti sebuah perintah yang berhubungan dengan pendefinisian dari suatu struktur database. Terdapat beberapa perintah DDL pada MySQL sebagai berikut :

1. *CREATE* berfungsi untuk membuat database baru, tabel baru, view baru dan kolom.
2. *ALTER* berfungsi untuk mengubah struktur tabel. Seperti mengganti nama tabel, menambah kolom, mengubah kolom, menghapus kolom maupun memberikan atribut pada kolom.
3. *DROP* berfungsi untuk menghapus database dan tabel.
4. *TRUNCATE* berfungsi untuk Menghapus semua catatan dari tabel.
5. *COMMENT* berfungsi untuk Menambahkan komentar pada data.
6. *RENAME* berfungsi untuk mengubah nama obyek.

Adapun contoh sintaks dari perintah DDL pada MySQL sebagai berikut :

```
CREATE DATABASE NILAI;  
ALTER TABLE Mahasiswa ADD (NoTelp char(8));  
DROP INDEX nama_index ;  
TRUNCATE TABLE table_barang;  
RENAME table_barang to table_barang_gudang;
```

Gambar 2.8 Sintax Perintah DDL pada MySQL

2. *Data Manipulation Language (DML)*

Data Manipulation Language (DML) ialah perintah yang digunakan untuk mengelola/memanipulasi data dalam database. Terdapat beberapa perintah DML pada MySQL sebagai berikut :

1. *SELECT* berfungsi untuk mengambil/menampilkan data dari database.
2. *INSERT* berfungsi untuk memasukkan data ke dalam tabel.
3. *UPDATE* berfungsi untuk memperbarui data dalam tabel.
4. *DELETE* berfungsi untuk menghapus data dari tabel.
5. *CALL* berfungsi untuk memanggil subprogram PL / SQL atau Java.
6. *EXPLAIN PLAN* berfungsi untuk menjelaskan jalur akses ke data.
7. *LOCK TABLE* berfungsi untuk mengunci tabel.

Adapun contoh sintaks dari perintah DML pada MySQL sebagai berikut :

```
INSERT INTO mahasiswa VALUES ("08052926",
"Frenky", "70");
SELECT nama_mahasiswa FROM mahasiswa;
DELETE FROM mahasiswa;
UPDATE mahasiswa SET nama_mahasiswa='ujang'
WHERE nim='08052926';
```

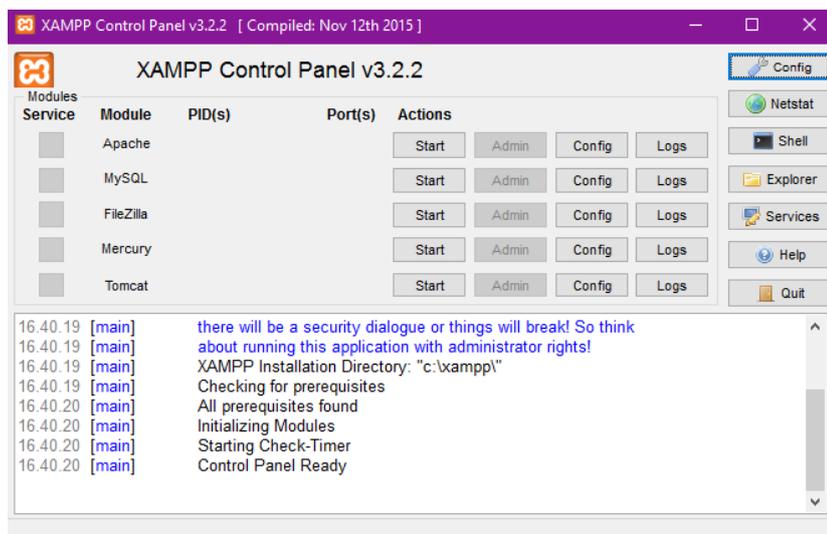
Gambar 2.9 Sintax Perintah DML pada MySQL

2.11 XAMPP

XAMPP adalah sebuah *software web server apache* yang didalamnya sudah tersedia database *server MySQL* dan dapat mendukung pemrograman PHP. XAMPP merupakan *software* yang mudah digunakan, gratis dan mendukung instalasi di Linux dan Windows. Keuntungan lainnya adalah cuma menginstal satu kali sudah tersedia *Apache Web Server, MySQL Database Server, PHP Support* (PHP 4 dan PHP 5) dan beberapa module lainnya.

Fungsi XAMPP sendiri adalah sebagai server yang berdiri sendiri (*localhost*), yang terdiri beberapa program antara lain : *Apache HTTP Server, MySQL database*, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan *Perl*. Nama XAMPP sendiri merupakan singkatan dari X

(empat sistem operasi apapun), *Apache*, *MySQL*, *PHP* dan *Perl*. Program ini tersedia dalam GNU (*General Public License*) dan bebas, merupakan *web server* yang mudah untuk digunakan yang dapat menampilkan halaman *web* yang dinamis.



Gambar 2.10 Aplikasi *XAMPP*

2.12 *Web Server*

Web Server adalah sebuah perangkat lunak yang bertugas menerima permintaan *client* melalui *port* HTTP maupun HTTPS dan merubahnya ke dalam format HTML. Terdapat beberapa format selain HTML yaitu PHP atau ASP, tetapi format – format tersebut hanyalah berfungsi untuk menghubungkan HTML dengan database.

Web server yang umum digunakan adalah *Apache*. Tugas utama dari *Web server* adalah menerjemahkan permintaan ke dalam respon yang cocok untuk keadaan pada saat itu, ketika klien membuka komunikasi dengan *Apache*, *Apache* mengirimkan permintaan untuk sumber daya. *Apache* menyediakan sumber daya yang baik atau memberikan respon alternatif untuk menjelaskan mengapa permintaan tidak dapat terpenuhi. Dalam banyak kasus, sumber daya adalah *Hypertext Markup Language* (HTML) halaman *web* yang berada pada *disk* lokal, tetapi ini hanya pilihan sederhana. Hal ini dapat berupa *file* gambar, hasil dari sebuah *script* yang menghasilkan *output* HTML, *applet* *Java* yang diunduh dan

dijalankan oleh klien, dan seterusnya, *Apache* menggunakan HTTP untuk berbicara dengan klien. Permintaan / tanggapan protokol ini, yang berarti bahwa ia mendefinisikan bagaimana membuat permintaan klien dan bagaimana *server* menanggapi mereka. Setiap komunikasi HTTP dimulai dengan permintaan dan berakhir dengan jawaban. *Executable Apache* mengambil nama dari protokol, dan pada sistem *Unix* umumnya disebut *httpd*, kependekan *daemon* HTTP.

