BAB II

TINJAUAN PUSTAKA

2. 1 Profil Unikom

Profil Unikom berisi penjelasan yang memaparkan tentang sejarah, logo, visi, misi, tujuan, dan struktur organisasi Unikom.

2. 1. 1. Sejarah Unikom [11]

Universitas Komputer Indonesia (UNIKOM) secara resmi berdiri pada hari Selasa, tanggal 8 Agustus 2000 berdasarkan Surat Keputusan Menteri Pendidikan Nasional nomor 126/D/0/2000.

Awalnya dimulai pada bulan Juli tahun 1994 ketika didirikan Lembaga Pendidikan Komputer Indonesia Jerman, disingkat LPKIG, bertempat di jalan Dipati Ukur 102 Bandung. Dengan 1 ruang kelas berkapasitas 50 orang dan 1 laboratorium komputer dengan 25 unit komputer, Lembaga ini membuka program pendidikan 1 tahun dengan 5 program studi yaitu Ahli Komputer Aplikasi Bisnis, Ahli Komputer Keuangan & Perbankan, Ahli Komputer Akuntansi & Perpajakan, Ahli Komputer Manajemen & Pemasaran dan Sekretaris Eksekutif. Jumlah peserta pendidikan pada tahun pertama ini sebanyak 233 siswa.

Pada tahun kedua, 1995, dibuka jenjang pendidikan 3 tahun untuk memenuhi animo siswa tahun pertama yang ingin memperdalam ilmunya, disamping pemikiran jangka panjang pengembangan institusi. Pada tahun ini juga dibuka program studi baru, meliputi : Ahli Komputer Teknik Informatika, Ahli Komputer Manajemen Informatika dan Sekretaris Eksekutif. Ruang kelas

ditambah menjadi 2 buah dan laboratorium komputer menjadi 2 buah dengan jumlah siswa sebanyak 457 orang.

Pada tahun ketiga, 1996, dilakukan penambahan gedung kuliah baru bertempat di jalan Dipati Ukur 116 (gedung FISIP sekarang), sekaligus pemindahan pusat administrasi dan perkantoran. Digedung baru ini dilakukan penambahan 1(satu) Lab. Komputer, 5(lima) Ruang Kuliah, Ruang Dosen dan Ruang Kemahasiswaan. Jumlah siswa dari tahun 1996 hingga tahun 1998 bertambah dari 632 orang menjadi 1184 orang.

Pada tahun kelima, 1998, dimulai pembangunan Kampus baru (Gedung Rektorat/Kampus-1 sekarang) berlantai 6 (enam) di jalan Dipati Ukur 114. Pembangunan Kampus baru ini dapat diselesaikan pada bulan Agustus 1999, sehingga pada awal perkuliahan bulan September 1999 telah dapat digunakan.

Mencermati dinamika peserta didik dan pengembangan Institusi kedepan, pada tanggal 24 Desember 1998 dibentuklah Yayasan Science dan Teknologi dan dilanjutkan dengan pengajuan pendirian STIMIK IGI dan STIE IGI ke DIKTI.

Pada bulan Juli 1999 STIE IGI diresmikan dengan keluarnya SK Mendiknas no. 119/D/O/1999 dengan 5 program studi : Akuntansi S1, Manajemen S1, Manajemen Pemasaran D3, Keuangan Perbankan D3 serta Akuntansi D3.

Pada bulan Agustus 1999 STIMIK IGI diresmikan dengan keluarnya SK Mendiknas no. 143/D/O/1999 dengan 5 program studi : Teknik Informatika S1, Manajemen Informatika D3, Teknik Komputer D3, Komputerisasi Akuntansi D3 serta Teknik Informatika D3.

Agar Sistem Pendidikan lebih Efisien, Efektif, Produktif dengan Struktur Organisasi yang lebih baik, enam bulan kemudian dilakukan usulan ke DIKTI untuk melakukan Merger kedua Sekolah Tinggi diatas menjadi Universitas.

Pada hari Selasa, tgl. 8 Agustus 2000 keluarlah SK MENDIKNAS no. 126/D/O/2000 atas Universitas Komputer Indonesia yang disingkat dengan nama UNIKOM.Pada SK tersebut sekaligus diijinkan dibukanya 11 program studi baru: Teknik Komputer S1, Manajemen Informatika S1, Teknik Industri S1, Teknik Arsitektur S1, Perencanaan Wilayah dan Kota S1, Ilmu Hukum S1, Ilmu Komunikasi S1, Ilmu Pemerintahan S1, Desain Interior D3, Desain Komunikasi Visual S1 dan Desain Komunikasi Visual D3.

Sejak berdirinya pada tahun 2000, setiap tahunnya UNIKOM menerima ± 2.000 mahasiswa baru. Terakhir pada tahun 2009 yang lalu diterima sebanyak 3.108 mahasiswa baru. Hingga tahun akademik 2009/2010 terdapat 6 Fakultas dan 23 Program Studi di UNIKOM dengan jumlah mahasiswa sebanyak 15.000 orang yang berasal dari berbagai pelosok tanah air dan dari luar negeri yang sedang menempuh pendidikan di UNIKOM.

2. 1. 2. Visi, Misi, dan Tujuan Unikom [12]

Berikut penjabaran dari visi, misi, dan tujuan Universitas Komputer Indonesia adalah:

2. 1. 2. 1. **Visi**

Menjadi Universitas terdepan dibidang Teknologi Informasi & Komputer, berwawasan Global dan menjadi pusat Unggulan dibidang ilmu pengetahuan Teknologi dan seni yang mendukung pembangunan nasional serta berorientasi pada kepentingan masyarakat, bangsa dan Negara

2. 1. 2. 2. **Misi**

Menyelenggarakan Pendidikan tinggi kearah masyarakat Industri maju dengan sistem pendidikan yang kondusif, tenaga pengajar berkualitas dan program-program studi berbasis pada teknologi informasi & komputer dengan mengoptimalkan sumber daya yang ada, kualitas dan manajemen mutu berdasarkan prinsip *Quality Is Our Tradition*.

2. 1. 2. 3. **Tujuan**

Menghasilkan Ilmuwan dan berpikiran tinggi maju dibidangnya masing-masing, mahir menggunakan teknologi informasi & komputer dalam bekerja serta beriman dan bertakwa kepada Tuhan Yang Maha Esa.

2. 1. 3. Logo Unikom [9]

Logo Unikom (Gambar II.1) memiliki makna dari setiap bagiannya, yaitu bingkai segi lima, lingkaran dalam segi lima tempat tulisan berwarna kuning, bulatan dalam berwarna biru, komputer, stasiun relay, satelit, cakrawala, dan buku.



Gambar II.1 Logo Unikom

Adapun penjabaran makna dari setiap bagiannya adalah sebagai berikut:

1. **Binkai Segi Lima**

Melambangkan UNIKOM berlandaskan falsafah negara yakni Pancasila dan Undang-Undang Dasar 1945.

2. **Lingkaran dalam Segi Lima Tempat Tulisan Berwarna Kuning**

Melambangkan motto UNIKOM menuju kejayaan yakni Quality Is Our Tradition.

3. **Bulatan dalam Berwarna Biru**

Melambangkan UNIKOM bertujuan menghasilkan ilmuwan unggul dan berpikiran maju yang Bertaqwa kepada Tuhan Yang Maha Esa.

4. **Komputer**

Melambangkan ciri utama UNIKOM yang memberikan pendidikan Teknologi Informasi dan Komputasi pada seluruh Jurusan yang ada dilingkungan Universitas Komputer Indonesia, menjadi Universitas Terdepan dibidang Teknologi Informasi dan Komputer serta sebagai Universitas komputer pertama di Indonesia.

5. Stasiun Relay

Melambangkan UNIKOM menyelenggarakan Pendidikan Tinggi kearah masyarakat industri maju dengan sistem pendidikan yang kondusif dan tenaga pengajar berkualitas untuk menghasilkan lulusan-lulusan terbaik.

6. Satelit

Melambangkan UNIKOM berwawasan Global dan menjadi pusat unggulan dibidang IPTEK & seni yang mendukung Pembangunan Nasional serta berorientasi pada kepentingan masyarakat, bangsa dan negara.

7. Cakrawala

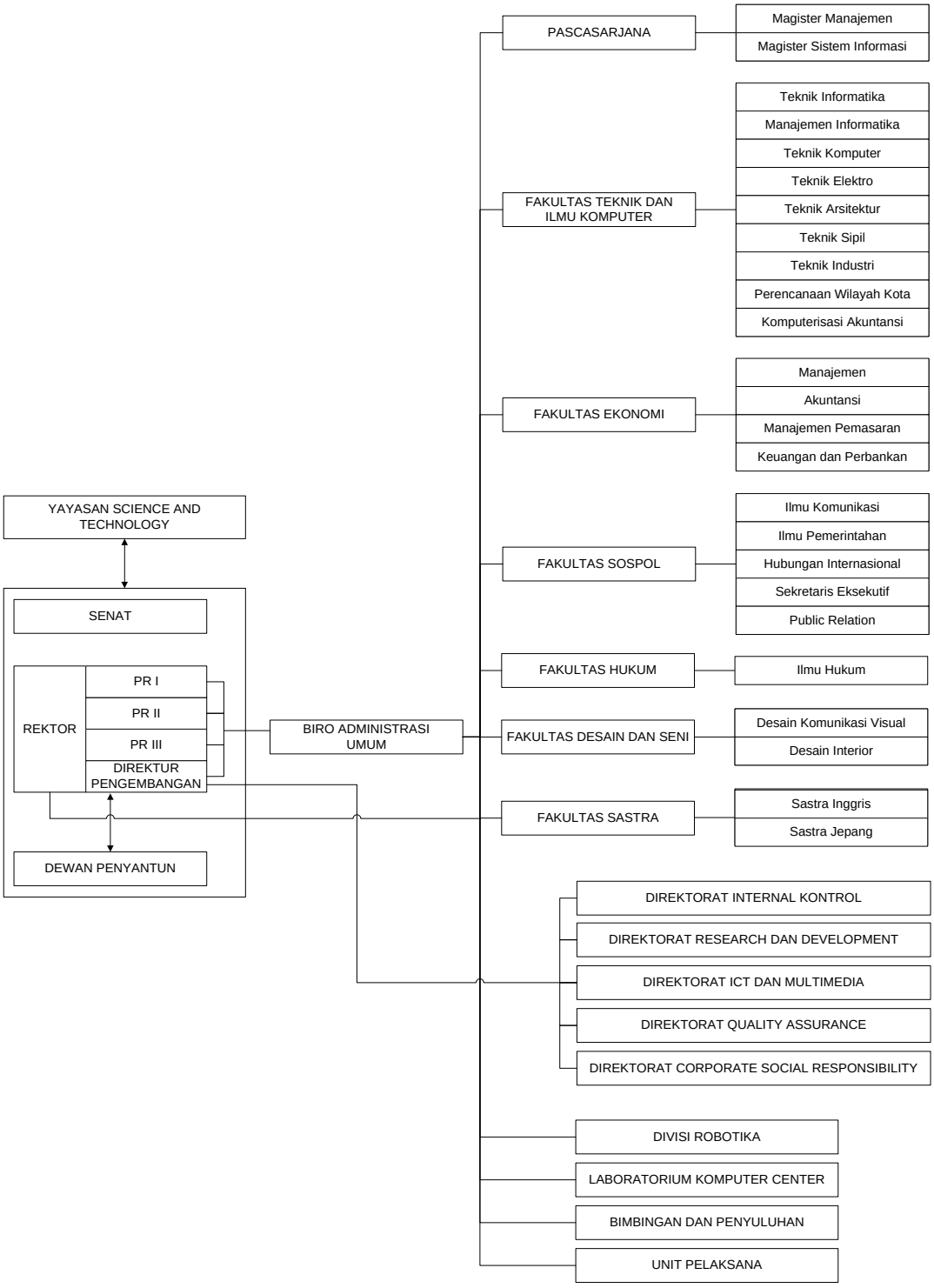
Melambangkan indahnya menggapai Cita-cita dan mengejar ilmu setinggi Langit.

8. Buku

Melambangkan sumber ilmu yang tiada habis-habisnya.

2. 1. 4. Struktur Organisasi Unikom [10]

Unikom terdiri dari bagian atau divisi yang memiliki peranannya masing-masing. Adapun susunan organisasi Unikom dapat dilihat pada Gambar II. 2.



Gambar II.2 Struktur Organisasi Unikom

Keterangan :
PR I : Pembantu Rektor Bidang Akademik
PR II : Pembantu Rektor Bidang Keuangan, Administrasi, dan Kepegawaian
PR III : Pembantu Rektor Bidang Kemahasiswaan dan Kerjasama

2. 2 Landasan Teori

Landasan teori merupakan penjelasan berbagai konsep dasar dan teori-teori yang berkaitan dalam pembangunan aplikasi *mobile* Kampus *Online* Unikom berbasis Android. Beberapa teori yang terkait dengan pembangunan aplikasi *mobile* Kampus *Online* Unikom berbasis Android adalah Kampus *Online*, sistem informasi, Android, Java, *Application Programming Interface* (API), Unikom SSO, Enkripsi MD5, *web service*, JSON, JDK, Eclipse, SQLite, dan UML.

2. 2. 1. Kampus *Online* [13]

Kampus *Online* UNIKOM merupakan sarana jejaring sosial yang sangat lengkap dengan bermacam-macam fasilitas yang dapat digunakan oleh penggunanya, dimulai dari *update status*, berbagi *link website*, *video* dengan YouTube sampai pada fasilitas blog yang dapat diatur sedemikian rupa agar dapat mencerminkan pribadi *user* Kampus *Online* itu.

Terdapat beberapa fasilitas unggulan Kampus *Online* UNIKOM yang dapat dimanfaatkan oleh penggunanya, sehingga fasilitas *online* ini dapat menjadi *One Stop Social Networking*, diantaranya:

1. Fasilitas *Real Time Chat* tanpa memerlukan aplikasi tambahan.
2. Album Foto beserta fasilitas *tagging.*
3. *Real Time Notification*, sehingga Anda akan mendapatkan notifikasi hal terbaru secara langsung tanpa melakukan *refresh.*

4. *Microblogging*/*Update Status* dengan fasilitas Komentar dan *Rating* yang mudah digunakan.
5. *User Rank* yang dapat memotifasi para *user* dalam beraktifitas di Kampus Unikom.
6. Integrasi langsung dengan Blog Unikom, dan dapat melakukan personalisasi tema tampilan, *widget*, dsb.
7. Integrasi dengan Jejaring sosial lainnya seperti Facebook dan Twitter, sehingga Anda dapat melakukan *Update Status* pada kedua jejaring sosial itu hanya dengan mengakses Kampus Unikom.
8. *Newsfeed* yang dapat digunakan untuk mengetahui berita-berita terbaru dari berbagai sumber berita *online*
9. Pertemanan dengan *Friend Suggest* yang akurat berdasarkan pada jenis akun milik Anda, baik itu mahasiswa, alumni, karyawan ataupun dosen.

Dengan beraktifitas dan melakukan *update* data terutama posting blog di Kampus *Online* Unikom ini, Anda secara langsung dapat membantu rangking Unikom pada situs-situs statistik seperti webometrics, 4icu dan alexa, selain itu Anda juga secara langsung membantu pengembangan kontent *website* Unikom, sehingga memiliki *size content* yang sangat tinggi.

Sebagai civitas UNIKOM dan belum terdaftar di Kampus *Online* UNIKOM, tidak ada salahnya untuk segera melakukan aktifasi dengan cara mengakses alamat http://kampus.unikom.ac.id/ pada *browser* Anda.

2. 2. 2. Sistem Informasi [4]

Pada dasarnya sistem informasi bukanlah sistem yang selalu terkait dengan komputer. Adapun beberapa penjelasan terkait dengan sistem informasi adalah pengertian sistem, pengertian data, pengertian informasi, dan pengertian sistem informasi itu sendiri.

2. 2. 2. 1. **Pengertian Sistem**

Pengertian sistem bisa dilihat dari dua pendekatan, yaitu berdasarkan prosedur dan komponennya. Sistem dilihat dari pendekatan prosedur berarti kumpulan prosedur yang saling bekerja sama untuk mencapai suatu tujuan tertentu sedangkan sistem dilihat dari pendekatan komponennya berarti sistem terdiri dari komponen-komponen yang saling berinteraksi untuk mencapai suatu tujuan tertentu.

2. 2. 2. 2. **Pengertian Data**

Data adalah kenyataan yang menggambarkan suatu kejadian–kejadian dan kesatuan yang nyata. Atau data adalah : representasi dunia nyata yang mewakili suatu objek seperti manusia (pegawai, mahasiswa, pelanggan), hewan, peristiwa, konsep, keadaan dll, yang direkam dalam bentuk angka, huruf, simbol, teks, gambar, bunyi atau kombinasinya.

2. 2. 2. 3. **Pengertian Informasi**

Informasi merupakan data yang memiliki nilai lebih bagi penerimanya karena telah melewati proses pengolahan tertentu sehingga bisa dimanfaatkan untuk mengambil keputusan baik itu untuk saat ini ataupun untuk masa yang akan

datang atau informasi didefinisikan sebagai data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya.

2. 2. 2. 4. **Pengertian Sistem Informasi**

Berdasarkan pemaparan sebelumnya mengenai sistem, data, dan informasi maka dapat diambil kesimpulan bahwa pengertian dari sistem informasi adalah sekumpulan komponen-komponen dan prosedur-prosedur yang saling bekerja sama untuk mencapai suatu tujuan tertentu. Berikut pengertian-pengertian lainnya dari sistem informasi:

1. *A set of people, data, procedures, hardware and software that work together to achieve the common goal of information management*.
2. Keterkaitan komponen-komponen dalam pengumpulan (atau menerima), proses, menyimpan dan mendistribusikan informasi untuk mendukung pembuatan keputusan (decision making), mengkoordinasi serta mengkontrol
3. Sistem informasi juga membantu, manajer dan para pegawai (pekerja) menganalisa masalah, menggambarkan ‘subyek’ yang kompleks dan membuat (menghasilkan) produk-produk baru
4. Dasar (pondasi) dari bisnis pada umumnya.

Ada tiga aktifitas pada sistem informasi menghasilkan informasi pada organisasi yang diperlukan untuk membuat suatu keputusan, pengkontrolan, analisa masalah-masalah dan membuat produk baru atau layanan. Aktifitas itu adalah *input*, *process* dan *output*.

Input berarti menangkap atau pengumpulkan data mentah (*raw data*) didalam organisasi atau dari luar organisasi – *environment* / lingkungan. Proses diartikan dengan menterjemahkan data *input* kedalam bentuk yang lebih berarti. Sedangkan *output* adalah mentransfer informasi yang telah diproses kepada manusia dan aktivitasnya.

2. 2. 3. Android

Android merupakan subset perangkat lunak untuk perangkat *mobile* yang meliputi sistem operasi, *middleware* dan aplikasi inti yang di-*release* oleh Google. Sedangkan Android SDK (*Software Development Kit*) menyediakan *Tools* dan API yang diperlukan untuk mengembangkan aplikasi pada *platform* Android dengan menggunakan bahasa pemrograman Java.

Dikembangkan bersama antara Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, NVIDIA yang tergabung dalam OHA (*Open Handset Alliance*) dengan tujuan membuat sebuah standar terbuka untuk perangkat bergerak (*mobile device*).

2. 2. 3. 1. **Sejarah [15]**

Pada Juli 2000, Google bekerjasama dengan Android Inc., perusahaan yang berada di Palo Alto, California Amerika Serikat. Para pendiri Android Inc. bekerja pada Google, diantaranya Andy Rubin, Rich Miner, Nick Sears, dan Chris White. Saat itu banyak yang menganggap fungsi Android Inc. hanyalah sebagai perangkat lunak pada telepon seluler. Sejak saat itu muncul rumor bahwa Google hendak memasuki pasar telepon seluler. Di perusahaan Google, tim yang dipimpin Rubin bertugas mengembangkan program perangkat seluler yang didukung oleh kernel Linux. Hal ini menunjukkan indikasi bahwa Google sedang bersiap

menghadapi persaingan dalam pasar telepon seluler. versi android terbaru yaitu versi 3.0. Android juga sudah bergabung dengan beberapa *smart mobile* seperti Nokia, Sony Ericsson, dan lainnya.

1. Produk Awal

Sekitar September 2007 sebuah studi melaporkan bahwa Google mengajukan hak paten aplikasi telepon seluler (akhirnya Google mengenalkan Nexus One, salah satu jenis telepon pintar GSM yang menggunakan Android pada sistem operasinya. Telepon seluler ini diproduksi oleh HTC Corporation dan tersedia di pasaran pada 5 Januari 2010).

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

Telepon pertama yang memakai sistem operasi Android adalah HTC Dream, yang dirilis pada 22 Oktober 2008. Pada penghujung tahun 2009 diperkirakan di dunia ini paling sedikit terdapat 18 jenis telepon seluler yang menggunakan Android.

2. Android Versi 1.1

Pada 9 Maret 2009, Google merilis Android versi 1.1. Android versi ini dilengkapi dengan pembaruan estetis pada aplikasi, jam alarm, *voice search* (pencarian suara), pengiriman pesan dengan Gmail, dan pemberitahuan *email*.

3. Android Versi 1.5 (Cupcake)

Pada pertengahan Mei 2009, Google kembali merilis telepon seluler dengan menggunakan Android dan SDK (Software Development Kit) dengan versi 1.5 (Cupcake). Terdapat beberapa pembaruan termasuk juga penambahan beberapa fitur dalam seluler versi ini yakni kemampuan merekam dan menonton video dengan modus kamera, mengunggah video ke Youtube dan gambar ke Picasa langsung dari telepon, dukungan *Bluetooth* A2DP, kemampuan terhubung secara otomatis ke *headset Bluetooth*, animasi layar, dan *keyboard* pada layar yang dapat disesuaikan dengan sistem.

4. Android Versi 1.6 (Donut)

Donut (versi 1.6) dirilis pada September dengan menampilkan proses pencarian yang lebih baik dibanding sebelumnya, penggunaan baterai indikator dan kontrol *applet* VPN. Fitur lainnya adalah galeri yang memungkinkan pengguna untuk memilih foto yang akan dihapus; kamera, camcorder dan galeri yang dintegrasikan; CDMA / EVDO, 802.1x, VPN, Gestures, dan *Text-to-speech engine*, kemampuan *dial* kontak, teknologi *text to change speech* (tidak tersedia pada semua ponsel), pengadaan resolusi VWGA.

5. Android Versi 2.0/2.1 (Eclair)

Pada 3 Desember 2009 kembali diluncurkan ponsel Android dengan versi 2.0/2.1 (Eclair), perubahan yang dilakukan adalah pengoptimalan *hardware*, peningkatan Google Maps 3.1.2, perubahan UI dengan *browser* baru dan dukungan HTML5, daftar kontak yang baru, dukungan *flash* untuk kamera 3,2 MP, *digital Zoom*, dan *Bluetooth* 2.1.

Untuk bergerak cepat dalam persaingan perangkat generasi berikut, Google melakukan investasi dengan mengadakan kompetisi aplikasi mobile terbaik (*killer apps* - aplikasi unggulan). Kompetisi ini berhadiah $25,000 bagi setiap pengembang aplikasi terpilih. Kompetisi diadakan selama dua tahap yang tiap tahapnya dipilih 50 aplikasi terbaik.

Dengan semakin berkembangnya dan semakin bertambahnya jumlah *handset* Android, semakin banyak pihak ketiga yang berminat untuk menyalurkan aplikasi mereka kepada sistem operasi Android. Aplikasi terkenal yang diubah ke dalam sistem operasi Android adalah Shazam, *Backgrounds*, dan *WeatherBug*. Sistem operasi Android dalam situs internet juga dianggap penting untuk menciptakan aplikasi Android asli, contohnya oleh MySpace dan Facebook.

6. Android Versi 2.2 (Froyo: Frozen Yoghurt)

Pada 20 Mei 2010, Android versi 2.2 (Froyo) diluncurkan. Perubahan-perubahan umumnya terhadap versi-versi sebelumnya antara lain dukungan Adobe Flash 10.1, kecepatan kinerja dan aplikasi 2 sampai 5 kali lebih cepat, integrasi V8 JavaScript *engine* yang dipakai Google Chrome yang mempercepat kemampuan *rendering* pada *browser*, pemasangan aplikasi dalam SD Card,

kemampuan WiFi *Hotspot portabel*, dan kemampuan *auto update* dalam aplikasi Android *Market*.

7. Android Versi 2.3 (Gingerbread)

Pada 6 Desember 2010, Android versi 2.3 (Gingerbread) diluncurkan. Perubahan-perubahan umum yang didapat dari Android versi ini antara lain peningkatan kemampuan permainan (*gaming*), peningkatan fungsi *copy paste*, layar antar muka (*User Interface*) didesain ulang, dukungan format video VP8 dan WebM, efek audio baru (*reverb*, *equalization*, *headphone virtualization*, dan *bass boost*), dukungan kemampuan *Near Field Communication* (NFC), dan dukungan jumlah kamera yang lebih dari satu.

8. Android Versi 3.0 (Honeycomb)

Android Honeycomb dirancang khusus untuk tablet. Android versi ini mendukung ukuran layar yang lebih besar. User Interface pada Honeycomb juga berbeda karena sudah didesain untuk tablet. Honeycomb juga mendukung multi prosesor dan juga akselerasi perangkat keras (*hardware*) untuk grafis. Tablet pertama yang dibuat dengan menjalankan Honeycomb adalah Motorola Xoom.

2. 2. 3. 2. ***Features*** **[7]**

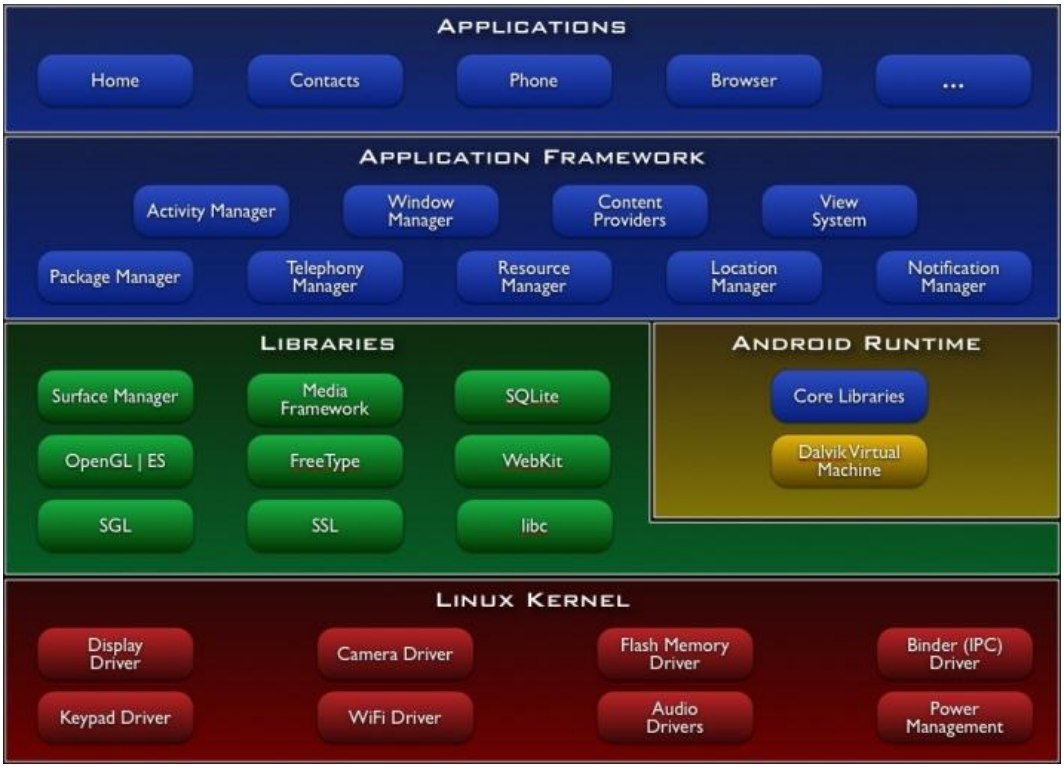
Features yang tersedia pada *platform* Android saat ini antara lain:

1. *Framework* aplikasi yang mendukung penggantian komponen dan *reusable*.
2. Mesin virtual Dalvik berjalan diatas Linux *kernel* dan dioptimalkan untuk perangkat *mobile*.
3. *Integrated browser* berdasarkan *open source engine* WebKit.

4. Grafis yang dioptimalkan dan didukung oleh library grafis 2D yang terkostumisasi, grafis 3D berdasarkan spesifikasi openGL ES 1,0 (Opsional akselerasi *hardware*).
5. SQLite untuk penyimpanan data.
6. *Media Support* yang mendukung audio, video, dan gambar (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF).
7. GSM *Telephony* (tergantung *hardware*).
8. *Bluetooth*, EDGE, 3D, dan WiFi (tergantung *hardware*).
9. Dukungan Perangkat Tambahan : Android dapat memanfaatkan kamera, layar sentuh, *accelerometers*, *magnetometers*, GPS, akselerasi 2D (dengan perangkat orientasi, *scalling*, konversi format piksel) dan akselerasi grafis 3D.
10. *Multi-touch* : kemampuan layaknya *handset modern* yang dapat menggunakan dua jari atau lebih untuk berinteraksi dengan perangkat.
11. Lingkungan *Development* yang lengkap dan kaya termasuk perangkat *emulator*, *tools* untuk *debugging*, profil dan kinerja memori, dan *plugin* untuk Eclipse IDE.
12. *Market* : seperti kebanyakan *handphone* yang memiliki tempat penjualan aplikasi, *market* pada Android merupakan katalog aplikasi yang dapat di-*download* dan di-*install* pada *handphone* melalui internet.

2. 2. 3. 3. **Arsitektur Android [7]**

Arsitektur Android menunjukkan komponen-komponen utama yang terdapat pada sistem operasi Android. Berikut ini arsitektur Android yang digambarkan pada Gambar II.3.



Gambar II.3 Arsitektur *Platform* Android

1. Linux *Kernel*

Android bukan Linux, akan tetapi Android dibangun diatas Linux *Kernel*, yaitu versi 2.6 sehingga kehandalannya bisa dipercaya. Untuk inti sistem *service* Linux yang digunakan seperti keamanan, manajemen memori, proses manajemen, *network*, dan *driver* model. Seperti yang terlihat digambar (Gambar II.3), Linux *Kernel* menyediakan *Driver* Layar, Kamera, *Keypad*, WiFi, *Flash Memory*, *Audio*, dan IPC (*Interprocess Communication*) untuk mengatur aplikasi dan keamanan.

Kernel juga bertindak sebagai lapisan abstrak antara *hardware* dan *software stack*-nya.

2. *Libraries*

Android menyertakan *libraries* C/C++ yang digunakan oleh berbagai komponen dari sistem Android. Kemampuan ini disediakan kepada *Developer* aplikasi melalui *Framework* Aplikasi Android. Beberapa inti *libraries* tercantum dibawah ini:

1. *System C Library* – Variasi dari implementasi BSD-berasal pelaksana sistem standar C *library* (*libc*), sesuai untuk perangkat *embedded* berbasis Linux.
2. *Media Libraries* – *PacketVideo* berdasarkan OpenCORE; *library* medukung pemutaran rekaman dan populer banyak format audio dan video, serta *file* gambar, termasuk MPEG4, H.264, MP3, AAC, AMR, JPG, dan PNG.
3. *Surface Manager* – mengelola akses ke subsistem layar, lapisan komposit 2D dan grafis 3D dari beberapa aplikasi.
4. LibWebCore – mesin *web modern* yang *powerfull* yang baik *browser* Android *embedded web*.
5. 3D *Libraries* – sebuah pelaksana berdasarkan openGL ES 1,0 API; perpustakaan baik menggunakan perangkat keras akselerasi 3D (apabila tersedia) atau yang disertakan, sangat optimal 3D *software rasterizer*.
6. FreeType – bitmap dan vektor *font rendering*.
7. SQLite – mesin *database* yang kuat dan ringan, dan penghubung tersedia untuk semua aplikasi.

3. *Android Runtime*

Android terdiri dari satu set *core libraries* yang menyediakan sebagian besar fungsi yang sama dengan yang terdapat pada *core libraries* bahasa pemrograman Java. Setiap aplikasi menjalankan prosesnya sendiri dalam Android, dengan masing-masing instan dari mesin virtual Dalvik (*Dalvik VM*). Dalvik dirancang agar perangkat dapat menjalankan multiple VMs secara efisien. Mesin Virtual Dalvik mengeksekusi *file* dalam Dalvik *executable* (.dex), sebuah format yang dioptimalkan untuk memori yang kecil.

Dalvik VM berbasis, berjalan, dan dikompilasi oleh *compiler* bahasa Java yang telah ditransformasikan ke dalam .dex format oleh tool "dx" yang telah disertakan. Dalvik VM bergantung pada *kernel* Linux untuk berfungsi seperti *threading* dan manajemen memori tingkat rendahnya.

4. *Framework* Aplikasi

Pengembang memiliki akses penuh menuju API *framework* yang sama dengan yang digunakan oleh aplikasi inti. Arsitektur aplikasi dirancang agar komponen dapat digunakan kembali (*reuse*) dengan mudah. Setiap aplikasi dapat memanfaatkan kemampuan ini (sesuai dengan batasan keamanan yang didefinisikan oleh *framework*). Mekanisme yang sama memungkinkan komponen untuk diganti oleh pengguna.

Semua aplikasi merupakan rangkaian set layanan dan sistem, termasuk:

a. *Views* yang kaya dan extensible yang dapat digunakan untuk membangun aplikasi, termasuk list, grids, kotak teks, tombol, dan bahkan sebuah *embedded web*.

b. *Content Provider* yang memungkinkan aplikasi untuk mengakses data (seperti dari daftar kontak telepon) atau membagi data tersebut.
c. *Resource Manager*, yang menyediakan akses ke kode nonsumber lokal seperti string, gambar, dan tata letak *file*.
d. *Notifications Manager* yang memungkinkan semua aplikasi menampilkan *alert* yang bisa dikostumisasi di dalam status bar.
e. *Activity Manager* yang mengelola siklus hidup aplikasi dan menyediakan navigasi umum *backstack*.

5. *Applications*

Android telah menyertakan aplikasi inti seperti *email client*, SMS, kalender, peta, *browser*, kontak, dan lainnya. Semua aplikasi tersebut ditulis dengan menggunakan bahasa pemrograman Java. Pada *layer* inilah *developer* atau kita menempatkan aplikasi yang dibuat. Yang istimewa adalah pada Android semua aplikasi baik aplikasi inti (*native*) maupun aplikasi pihak ketiga berjalan pada *layer* aplikasi dengan menggunakan *library* API yang sama. Ini berarti semua aplikasi yang dibuat untuk Android akan memiliki akses yang setara dalam mengakses seluruh kemampuan *handset*, tanpa membedakan apakah itu merupakan aplikasi inti atau aplikasi pihak ketiga. Dalam kata lain dengan *platform* Android ini, *Programmer* atau *Developer* secara penuh akan bisa mengkostumisasi perangkat Androidnya.

2. 2. 4. Java [16]

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*.

2. 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. 2. 4. 2. **Versi Awal**

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

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

2. 2. 4. 3. **Kelebihan**

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. 2. 4. 4. **Kekurangan**

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. 2. 4. 5. **Tahap Kompilasi Java**

1. Tulis / Ubah. Pemrogram menulis program dan menyimpannya di media dalam bentuk berkas '*.java'*.
2. Kompilasi. Pengkompilasi membentuk *bytecodes* dari program menjadi bentuk berkas '*.class'*.
3. Muat. Pemuat kelas memuat *bytecodes* ke memori.
4. Verifikasi. Peng-verifikasi memastikan *bytecodes* tidak mengganggu sistem keamanan Java.
5. Jalankan. Penerjemah menerjemahkan *bytecodes* ke bahasa mesin tidak bisa di pakai.

2. 2. 5. *Aplication Programming Interface* (API) [2]

Antarmuka pemrograman aplikasi (*Application Programming Interface*) adalah sekumpulan perintah, fungsi, dan protokol yang dapat digunakan oleh pengembang saat membangung perangkat lunak untuk sistem operasi tertentu. API memungkinkan pengembang untuk menggunakan fungsi standar untuk berinteraksi dengan sistem operasi.

2. 2. 5. 1. **Kampus *Online* Unikom *Application Interface***

Dalam rangka pengembangan layanan Kampus *Online* Unikom agar dapat berjalan pada *multi-platform*, dengan ini Unikom *Center* mengenalkan suatu fasilitas *Application Interface* (API) untuk memungkinkan suatu Aplikasi

melakukan interaksi dengan Kampus *Online* Unikom. Kampus API merupakan sarana berbagi data antara suatu aplikasi yang telah di daftarkan dengan Kampus *Online* Unikom. Dengan layanan ini, pengembang aplikasi baik itu web, *desktop* ataupun *mobile* dapat membuat sendiri *interface* Kampus *Online* Unikom berdasarkan *platform* masing-masing dengan data yang sama pada server utama Kampus *Online* Unikom.

2. 2. 5. 2. **Data Autentikasi API Kampus *Online* Unikom**

Untuk memanfaatkan fasilitas Kampus API ini, pengembang aplikasi Kampus Online Unikom harus mendaftarkan aplikasi yang akan di bangunnya terlebih dahulu. Silahkan hubungi Unikom Center. Setelah terdaftar, maka Anda akan mendapatkan dua variabel penting yang akan selalu digunakan ketika Anda melakukan akses pada Kampus API. Dua variabel tersebut adalah **API_KEY** dan **API_SECRET**.

1. API_KEY

API_KEY merupakan kunci unik suatu aplikasi yang terdaftar sebagai pengguna layanan Kampus API. Kunci ini merupakan kunci Unik seperti halnya username tapi digeneralisasi oleh Kampus Online secara otomatis. API_KEY memiliki panjang 32 karakter, dan hanya bisa digunakan berpasangan dengan API_SECRET. API_KEY akan selalu digunakan pada request HTTP ketika mengakses Kampus API, hal ini dimaksudkan agar Server Kampus Online Unikom dapat mengetahui request yang dimaksud benar-benar berasal dari aplikasi yang sudah terdaftar tersebut.

2. API_SECRET

API_SECRET merupakan kunci rahasia dari API_KEY, berfungsi seperti SALT pada enkripsi. Semua request modul/perintah API harus di enkripsi dengan SALT dan API_KEY yang dimaksud, sehingga request yang serupa tidak dapat dilakukan oleh pihak lainnya yang tidak memiliki API_KEY dan API_SECRET. API_SECRET juga diperlukan untuk melakukan enkripsi pada data-data riskan seperti password. Sehingga data yang di transfer tidak mungkin untuk di decrypt oleh pihak lain di tengah jalan, dan Server Kampus juga hanya dapat melakukan pencocokan password dengan salt yang dimaksud.

2. 2. 6. Unikom *Single Sign On* (SSO) [14]

Sigle Sign On merupakan fasilitas yang memberikan kemudahan untuk user yang melakukan login ketika mereka *surfing* di internet. Dengan fasilitas *Single Sign On* pengguna hanya akan diminta untuk melakukan *login* sekali saja untuk semua fasilitas *online* yang ada. Hampir semua situs besar dunia telah memanfaatkan fasilitas ini dengan implementasi yang beragam tapi dengan tujuan yang sama, diantaranya Yahoo!, Google, Microsoft, dan sebagainya.

Unikom *Single Sign On* API (USSO-API) merupakan *Application Interface* atau *Library* (pustaka) yang akan digunakan oleh pengembang aplikasi *online* berbasis web di Unikom agar dapat mengintegrasikan aplikasi yang dibuatnya dengan fasilitas USSO.

USSO-API dibuat dengan menggunakan bahasa Pemrograman PHP (*Hypertext Preprocessor*). USSO-API disusun secara *class based* dengan nama *class* "**ssoclass**" dan dapat diimplementasikan secara mudah di dalam *script* PHP

yang sedang di bangun. USSO-API selain memberikan fasilitas *single sign on* juga memberikan fasilitas transfer *encryption* tanpa *Secure Socket Layer* (SSL).

Unikom Single Sign On (**USSO**) bertujuan untuk memudahkan pengguna dalam berinteraksi *online* dan mengintegrasikan *account user* ke dalam satu *database* yang dapat digunakan oleh masing-masing aplikasi *online* di Unikom. Dengan USSO, pengembang aplikasi berbasis web di Unikom tidak perlu lagi membuat basis data dan *coding* sistem yang khusus menangani *session login*.

2. 2. 7. Enkripsi MD5 [17]

Dalam kriptografi, MD5 (Message-Digest algortihm 5) ialah fungsi hash kriptografik yang digunakan secara luas dengan hash value 128-bit. Pada standard internet (RFC 1321), MD5 telah dimanfaatkan secara bermacam-macam pada aplikasi keamanan, dan MD5 juga umum digunakan untuk melakukan pengujian integritas sebuah file.

MD5 didesain oleh Ronald Rivers pada tahun 1991 untuk menggantikan hash function sebelumnya, MD4. Pada tahun 1996, sebuah kecacatan ditemukan dalam desainnya, walau bukan kelemahan fatal, pengguna kriptografi mulai menganjurkan menggunakan algoritma lain seperti SHA-1. Pada tahun 2004, kecacatan-kecacatan yang lebih serius ditemukan menyebabkan penggunaan algoritma tersebut dalam tujuan untuk keamanan jadi makin dipertanyakan.

2. 2. 8. *Web Service* [1]

Web Service adalah sekumpulan *application logic* beserta objek-objek dan metode-metode yang dimilikinya yang terletak di suatu *server* yang terhubung ke internet sehingga dapat diakses menggunakan *protocol* HTTP dan SOAP (*Simple*

Object Access Protocol). Dalam penggunaannya, *web service* dapat digunakan dari hanya untuk memeriksa data *user* yang *login* ke sebuah *website* ataupun untuk digunakan pada transaksi perbankan *online* yang rumit.

Tujuan dari teknologi ini adalah untuk memudahkan beberapa aplikasi atau komponennya untuk saling berhubungan dengan aplikasi lain dalam sebuah organisasi maupun diluar organisasi menggunakan standar yang tidak terikat *platform* (*platform-neutral*) dan tidak terikat akan bahasa pemrograman yang digunakan (*language-neutral*). Hal tersebut dapat terjadi karena penggunaan XML standar yang didukung oleh banyak perusahaan besar di dunia, yang digunakan untuk bertukar data. Selain daripada itu, penggunaan SOAP menjadikan metode-metode dari objek-objek yang ada dalam sebuah *web service* dapat di akses dari aplikasi lain seperti halnya aplikasi tersebut mengakses metode lokal.

2. 2. 8. 1. ***Web Services Messaging*** **[5]**

W3C mendefinisikan *web service* sebagai "sebuah *software* aplikasi yang dapat teridentifikasi oleh URI dan memiliki *interface* yang didefiniskan, dideskripsikan, dan dimengerti oleh XML dan juga mendukung interaksi langsung dengan *software* aplikasi yang lain dengan menggunakan *message* berbasis XML melalui protokol internet"

Web service adalah sebuah *sofware* aplikasi yang tidak terpengaruh oleh *platform*, ia akan menyediakan *method-method* yang dapat diakses oleh *network*. Ia juga akan menggunakan XML untuk pertukaran data, khususnya pada dua *entities* bisnis yang berbeda.

Beberapa karakteristik dari *web service* adalah:

1. *Message-based*
2. *Standards-based*
3. *Programming language independent*
4. *Platform-neutral*

Beberapa *key standard* didalam *web service* adalah:

1. XML,
2. SOAP,
3. WSDL, dan
4. UDDI.

SOAP (*Simple Object Access Protocol*) adalah sebuah XML-*based mark-up language* untuk pergantian pesan diantara aplikasi-aplikasi. SOAP berguna seperti sebuah amplop yang digunakan untuk pertukaran data object didalam *network*. SOAP mendefinisikan empat aspek didalam komunikasi: *Message envelope*, *Encoding*, RPC *call convention*, dan bagaimana menyatukan sebuah *message* didalam protokol *transport*.

Sebuah SOAP message terdiri dari SOAP *Envelop* dan bisa terdiri dari *attachments* atau tidak memiliki *attachment*. SOAP *envelop* tersusun dari SOAP *header* dan SOAP *body*, sedangkan SOAP *attachment* membolehkan non-XML data untuk dimasukkan kedalam SOAP *message*, di-*encoded*, dan diletakkan kedalam SOAP message dengan menggunakan MIME-*multipart*.

2. 2. 8. 2. ***Web Services Description*** **[5]**

WSDL (*Web Services Description Language*) adalah sebuah XML-*based language* untuk mendeskripsikan XML. Ia menyediakan *service* yang mendeskripsikan *service request* dengan menggunakan protokol-protokol yang berbeda dan juga *encoding*. Ia akan memfasilitasi komunikasi antar aplikasi. WSDL akan mendeskripsikan apa yang akan dilakukan oleh *web service*, bagaimana menemukannya dan bagaimana untuk mengoperasikannya.

Spesifikasi WSDL mendefinisikan tujuh tipe *element*:

1. *Types* - elemen untuk mendefinisikan tipe data. Types akan mendefinisikan tipe data (seperti string atau integer) dari elemen didalam sebuah message.
2. *Message - abstract*, pendefinisian tipe data yang akan dikomunikasikan.
3. *Operation* - sebuah deskripsi abstract dari sebuah action yang didukung oleh *service*.
4. *Port Type* - sebuah koleksi *abstract* dari *operations* yang didukung oleh lebih dari satu *endpoints*.
5. *Binding* - mendefinisikan penyatuan dari tipe port (koleksi dari operasi-operasi) menjadi sebuah protokol transport dan data format (contohnya SOAP 1.1 pada HTTP). Ini adalah sebuah protokol konkret dan sebuah spesifikasi data format didalam tipe port tertentu.

6. *Port* - mendefinisikan sebuah komunikasi *endpoint* sebagai kombinasi dari *binding* dan alamat *network*. Bagi protokol HTTP, ini adalah sebuah bentuk dari URL sedangkan bagi protokol SMTP, ini adalah sebuah *form* dari *email address*.
7. *Service* – satu set *port* yang terkorelasi atau suatu *endpoints*.

2. 2. 8. 3. ***Web Service Discovery*** **[5]**

UDDI (*Universal Description, Discovery and Integration*) adalah sebuah *service registry* bagi pengalokasian *web service*. UDDI mengkombinasikan SOAP dan WSDL untuk pembentukan sebuah *registry* API bagi pendaftaran dan pengenalan *service*. Ia menyediakan sebuah area umum dimana sebuah organisasi dapat mengiklankan keberadaan mereka dan *service* yang mereka berikan (*web service*).

UDDI adalah sebuah *framework* yang mendefinisikan sebuah XML-*based registry* dimana sebuah organisasi dapat meng-upload informasi mengenai *service* yang mereka berikan. XML-*based registry* berisi nama-nama dari organisasi tsb, beserta *service* dan deskripsi dari *service* yang mereka berikan.

2. 2. 9. JSON [6]

JSON (*JavaScript Object Notation*) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (*generate*) oleh komputer. Format ini dibuat berdasarkan bagian dari Bahasa Pemprograman JavaScript, Standar ECMA-262 Edisi ke-3 – Desember 1999. JSON merupakan format teks yang tidak bergantung pada bahasa

pemprograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh *programmer* keluarga C termasuk C, C++, C#, Java, JavaScript, Perl, Python dll. Oleh karena sifat-sifat tersebut, menjadikan JSON ideal sebagai bahasa pertukaran-data.

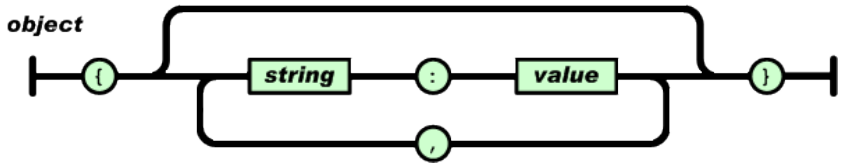
JSON terbuat dari dua struktur:

1. Kumpulan pasangan nama/nilai. Pada beberapa bahasa, hal ini dinyatakan sebagai objek (object), rekaman (record), struktur (struct), kamus (dictionary), tabel hash (hash table), daftar berkunci (keyed list), atau associative array.
2. Daftar nilai terurutkan (an ordered list of values). Pada kebanyakan bahasa, hal ini dinyatakan sebagai larik (array), vektor (vector), daftar (list), atau urutan (sequence).

Struktur-struktur data ini disebut sebagai struktur data universal. Pada dasarnya, semua bahasa pemprograman moderen mendukung struktur data ini dalam bentuk yang sama maupun berlainan. Hal ini pantas disebut demikian karena format data mudah dipertukarkan dengan bahasa-bahasa pemprograman yang juga berdasarkan pada struktur data ini. JSON menggunakan bentuk sebagai berikut:

1. Objek

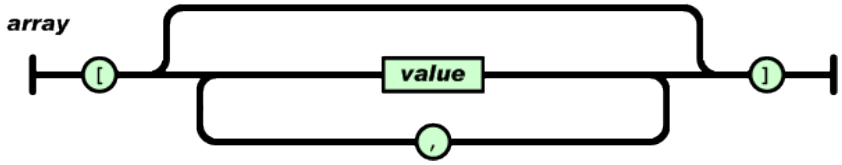
Objek adalah sepasang nama/nilai yang tidak terurutkan. Objek dimulai dengan { (kurung kurawal buka) dan diakhiri dengan } (kurung kurawal tutup). Setiap nama diikuti dengan : (titik dua) dan setiap pasangan nama/nilai dipisahkan oleh , (koma).



Gambar II.4 Objek JSON

2. Larik

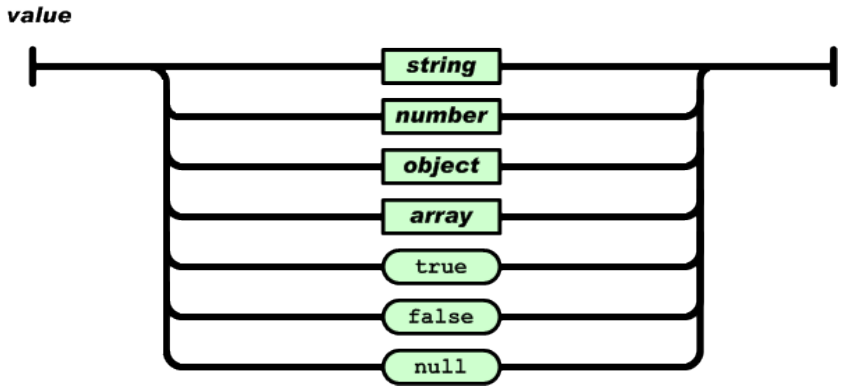
Larik adalah kumpulan nilai yang terurutkan. Larik dimulai dengan [(kurung kotak buka) dan diakhiri dengan] (kurung kotak tutup). Setiap nilai dipisahkan oleh , (koma).



Gambar II.5 Larik JSON

3. Nilai

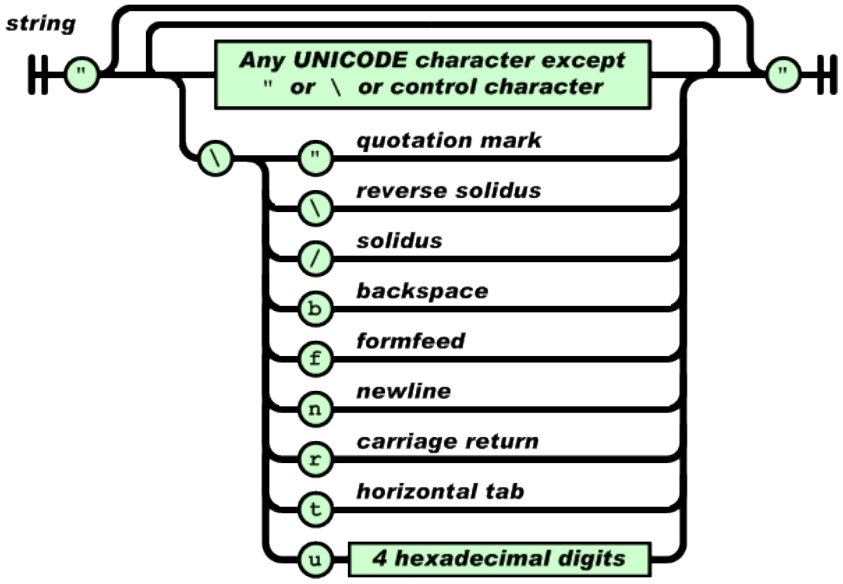
Nilai (*value*) dapat berupa sebuah string dalam tanda kutip ganda, atau angka, atau *true* atau *false* atau *null*, atau sebuah objek atau sebuah larik. Struktur-struktur tersebut dapat disusun bertingkat.



Gambar II.6 Nilai JSON

4. String

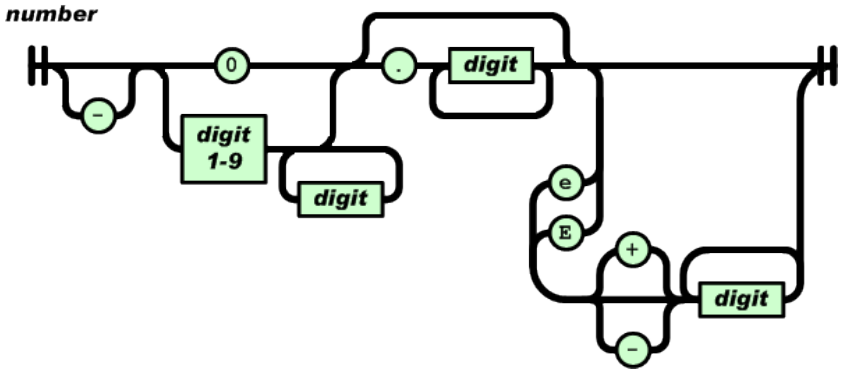
String adalah kumpulan dari nol atau lebih karakter *Unicode*, yang dibungkus dengan tanda kutip ganda. Di dalam string dapat digunakan *backslash escapes* "\" untuk membentuk karakter khusus. Sebuah karakter mewakili karakter tunggal pada string. String sangat mirip dengan string C atau Java.



Gambar II.7 String JSON

5. Angka

Angka adalah sangat mirip dengan angka di C atau Java, kecuali format oktal dan heksadesimal tidak digunakan.



Gambar II.8 Angka JSON

2. 2. 10. *Tools* Implementasi

Tools implementasi merupakan peralatan-peralatan yang dibutuhkan dalam membangun perangkat lunak. Adapun *tools* implementasi yang digunakan untuk membuat aplikasi *mobile* jejaring sosial Kampus *Online* Unikom berbasis Android adalah JDK sebagai perangkat pembangun Java, Eclipse IDE sebagai lingkungan pengembangan perangkat lunak, dan SQL sebagai sistem manajemen basis data.

2. 2. 10. 1. **JDK [8]**

JDK (*Java Development Kit*) adalah sebuah perangkat peralatan yang digunakan untuk membangun perangkat lunak dengan menggunakan bahasa pemrograman Java. JDK berjalan diatas sebuah *virtual machine* yang dinamakan JVM (Java *Virtual Machine*).

Dokumentasi JDK berisi spesifikasi API, deskripsi fitur, panduan pengembang, referensi halaman untuk perkakas JDK dan utilitas, demo, dan link ke informasi terkait. Dokumentasi ini juga tersedia dalam bundel yand dapat diunduh dan diinstal pada komputer.

2. 2. 10. 2. **Eclipse IDE [3]**

Eclipse adalah sebuat komunitas proyek *open source* yang difokuskan pada membangun sebuah *platform* pengembangan yang dapat diperluas, *runtime* dan kerangka aplikasi untuk membangun, menyebarkan dan mengelola perangkat lunak di seluruh daur hidup perangkat lunak.

Komunitas Eclipse memiliki lebih dari 60 proyek *open source*. Proyek-proyek ini secara konsep terbagi menjadi 7 kategori :

1. *Enterprise Development*
2. *Embedded and Device Development*
3. *Rich Client Platform*
4. *Rich Internet Applications*
5. *Application Frameworks*
6. *Application Lifecycle Management* (ALM)
7. *Service Oriented Architecture* (SOA)

Komunitas Eclipse juga didukung oleh ekosistem besar dan dinamis dari penyedia solusi teknologi informasi utama, *start-up* inovatif, universitas dan lembaga penelitian, dan individu yang memperluas, mendukung, dan melengkapi *platform* Eclipse.

2. 2. 11. UML [18]

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. 2. 11. 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. 2. 11. 2. **Diagram UML**

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

1. *Use Case Diagram* untuk memodelkan proses bisnis.
2. *Conceptual Diagram* untuk memodelkan konsep-konsep yang ada di dalam aplikasi.
3. *Sequence Diagram* untuk memodelkan pengiriman pesan (*message*) antar *objects*
4. *Collaboration Diagram* untuk memodelkan interaksi antar *objects*.
5. *State Diagram* untuk memodelkan perilaku *objects* di dalam sistem.
6. *Activity Diagram* untuk memodelkan perilaku *Use Cases* dan *objects* di dalam *system.*
7. *Class Diagram* untuk memodelkan struktur kelas.
8. *Object Diagram* untuk memodelkan struktur *object*.
9. *Component Diagram* untuk memodelkan komponen *object*.
10. *Deployment Diagram* untuk memodelkan distribusi aplikasi.

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.