

Sistem

Operasi

Symbian

Mauldy Laya

| Sejarah | Spesifikasi | Komponen | Arsitektur | Manajemen Memori |
| Manajemen Proses | Manajemen I/O | Komunikasi |

Sistem Operasi Symbian
versi dokumen 1.1

Hak Cipta © 2004 Mauldy Laya.

Silakan menyalin, mengedarkan, dan/atau, memodifikasi bagian dari dokumen, sesuai dengan ketentuan "GNU Free Documentation License versi 1.2" atau versi selanjutnya dari FSF (Free Software Foundation); tanpa bagian "Invariant", tanpa teks "Front-Cover", dan tanpa teks "Back-Cover".

Sistem Operasi Symbian

Abstrak:

Telepon selular (ponsel) semakin berkembang pesat dari segi fungsi dan teknologi. Perkembangan itu tidak akan jauh jika tidak ada sistem operasi yang baik dan handal yang mengatur keseluruhan sistem ponsel. Tulisan ini membahas sistem operasi populer ponsel untuk kelas menengah keatas yaitu Symbian OS. Pembahasan mengarah pada aspek teoritis dan praktis mulai dari sejarah, manajemen proses, manajemen memori, I/O, sampai pada komunikasi dan keamanan. Aspek sejarah membahas mulai dari terbentuknya perusahaan symbian sampai sejarah versi sistem operasinya. Aspek manajemen proses, memori dan I/O membahas sistem operasi sebagai pengatur proses yang terjadi, alokasi memori dan pengaturan input output. Aspek komunikasi dan keamanan membahas hal terpenting dari ponsel yaitu arsitektur komunikasi, teknologi transfer dan keamanan data.

Kata kunci: Sistem operasi, Symbian, Ponsel.

Daftar Isi

| | |
|---|-----|
| Lisensi..... | ii |
| Abstrak..... | iii |
| Daftar isi..... | iv |
| 1. Pendahuluan..... | 1 |
| 2. Symbian OS | |
| 2.1 Sejarah..... | 2 |
| 2.2 Spesifikasi..... | 3 |
| 2.3 Komponen..... | 4 |
| 2.3.1 Kernel | |
| 2.3.2 Middleware | |
| 2.3.3 Application Engine | |
| 2.3.4 UI Framework | |
| 2.3.5 Synchronization | |
| 2.3.6 JVM | |
| 2.4 Arsitektur..... | 7 |
| 2.4.1 Base | |
| 2.4.2 Application framework | |
| 2.4.3 Multimedia | |
| 2.4.4 Communication infrastructure and network stacks | |
| 2.4.5 Messaging | |
| 2.4.6 Browsing | |
| 2.4.7 Application protocol, service and engines | |
| 2.4.8 Java | |
| 2.4.9 Connectivity | |
| 2.4.10 Tools | |
| 2.5 Manajemen memori..... | 12 |
| 2.6 Manajemen proses..... | 13 |
| 2.7 Manajemen I/O..... | 14 |
| 2.8 Komunikasi..... | 14 |
| 2.8.1 Sever komunikasi | |
| 2.8.2 Modul komunikasi | |
| 2.8.3 Keamanan komunikasi | |
| 3. Kesimpulan..... | 18 |
| Daftar pustaka..... | 19 |
| Lampiran..... | 20 |

1. Pendahuluan

Telepon selular (ponsel) semakin berkembang dari hari ke hari. Perkembangan tak hanya dari sudut teknologi tetapi juga dari sudut tampilan atau *look and feel*. Selain sebagai alat untuk berkomunikasi, ponsel juga menjadi alat untuk melakukan berbagai komunikasi yang lainnya seperti *email*, *internet*, *fax* dan sebagainya. Ponsel dengan segudang *feature* dan memiliki teknologi yang tinggi dapat dikatakan sebagai sebuah *Smartphone*. Sebagai sebuah ponsel yang *smart* maka dituntut adanya sebuah sistem operasi yang stabil dan handal, tidak boleh terjadi kerusakan sistem dan proses *rebooting* walaupun mempunyai memori dan sumber daya yang terbatas.

Smartphone harus mempunyai sistem yang selalu hidup dan berjalan untuk waktu seminggu atau sebulan tanpa *restarting*, walaupun pada kenyataannya banyak orang yang mematikan ponsel pada saat naik pesawat. Sebuah *smartphone* juga dituntut mempunyai media penyimpanan data untuk data-data penting ataupun tidak penting dan tidak boleh terjadi kerusakan atau bahkan kehilangan data. Untuk itu diperlukan sebuah sistem operasi dengan manajemen memori yang handal. Kriteria lain untuk sebuah ponsel yang *smart* adalah konsumsi daya yang rendah tanpa harus melakukan pengisian ulang setiap hari.

Sistem operasi sebuah *smartphone* merupakan komponen software yang sangat kritis karena mengatur keseluruhan dari perangkat ponsel. Kebutuhan yang penting dari sebuah sistem operasi selain daripada yang telah disebutkan diatas adalah kemampuan *multitasking*, *multithreading*, *real-time operasi*, manajemen daya yang efektif, mempunyai code yang kecil, mudah menambah fungsi-fungsi baru, dapat digunakan kembali, modular, *connectivity* dan handal.

Telah ada beberapa sistem operasi yang menjadi basis beberapa tipe ponsel dan perangkat bergerak lainnya seperti PDA, diantaranya yaitu Windows CE, Palm OS, Embedded Linux, Pocket PC, dan Symbian OS. Namun untuk kalangan ponsel-ponsel kelas menengah keatas dikuasai oleh Symbian OS yang merupakan sistem operasi hasil kolaborasi dari para pembuat ponsel terbesar di dunia, seperti Nokia, Motorola, Samsung, Sony Ericsson dan Siemens.

Symbian sendiri merupakan sebuah perusahaan patungan antara Nokia, Motorola, Ericsson, Matsushita, dan Psion. Symbian didirikan untuk membuat sistem operasi yang optimum untuk perangkat bergerak yang diberi nama Symbian OS. Semenjak berdiri hingga sekarang, Symbian OS telah memimpin dalam industri perangkat bergerak seperti *communicator* dan *smartphones*.

Symbian OS dirancang untuk perangkat bergerak yang mengutamakan efisiensi penggunaan sumber daya memori dan baterai. Platform symbian terbuka untuk umum dengan pustaka API yang cukup lengkap, sehingga banyak aplikasi-aplikasi yang sudah dibuat untuk sistem operasi ini. Aplikasi dapat dibuat dengan beberapa bahasa seperti C++, Java, atau bahkan VB. Sistem operasinya sendiri dibuat dengan bahasa C++.

Tulisan ini membahas sistem operasi Symbian OS. Pembahasan mulai dari sejarah, spesifikasi, komponen, arsitektur, manajemen proses, manajemen memori, I/O, sampai pada komunikasi dan keamanan. Sejarah akan membahas dari terbentuknya perusahaan symbian sampai sejarah versi sistem operasinya. Spesifikasi akan membahas

mengenai ciri-ciri penting Symbian OS. Komponen membahas bagian-bagian penyusun sistem operasi tersebut. Arsitektur membahas modul-modul teknologi secara lebih detail. Manajemen proses, memori dan I/O membahas sistem operasi sebagai pengatur proses yang terjadi, alokasi memori dan pengaturan input output. Komunikasi dan keamanan akan membahas hal terpenting dari ponsel yaitu arsitektur komunikasi, teknologi transfer dan keamanan data.

Dengan adanya tulisan ini diharapkan dapat memberikan informasi baru ataupun tambahan informasi kepada para pembaca dalam topik sistem operasi.

2. Symbian OS

2.1 Sejarah

Pada awal mulanya di tahun 1998, symbian merupakan sebuah perusahaan patungan dari beberapa pemain di dunia ponsel yaitu Psion, Ericsson, Nokia, dan Motorola. Kemudian pada tahun 1999, Matsushita(Panasonic) bergabung kedalamnya. Pada tahun-tahun berikutnya banyak para pembuat ponsel yang mulai mengadopsi teknologi symbian. Tahun 2000, Sony dan Sanyo melisensi Symbian OS, tahun 2001 Fujitsu dan Siemens membeli lisensi juga. Tahun 2002 Samsung membeli lisensi, ditahun yang sama pula Siemens dan Sony Ericsson menjadi salah satu pemegang sahamnya. Samsung menjadi salah satu pemegang saham ditahun 2003. Di tahun 2004, Psion dan Motorola menjual sahamnya kepada Nokia, sekarang Symbian dikuasai sebagian besar sahamnya oleh Nokia yang merupakan pemimpin didalam penjualan ponsel-ponselnya diseluruh dunia.

Ponsel pertama yang bersistem operasi Symbian OS adalah Ericsson R380 *smartphone* yang dikeluarkan tahun 2000. Dilanjutkan dengan dikeluarkannya dan diumumkannya ponsel-ponsel dengan sistem operasi symbian, seperti Nokia 9210 *communicator* dan Nokia 7650 di tahun 2001. Dilanjutkan dengan diumumkannya ponsel P800 buatan Sony Ericsson yang mengadopsi symbian versi 7.0 ditahun 2002. Tahun 2003 mulai banyak ponsel-ponsel yang berbasis Symbian OS, diantaranya Nokia 3660, Nokia 3620, Nokia N-Gage, Nokia 6600, Nokia 7700, Sendo X, Siemens SX1, Sony Ericsson P900, BenQ P30, Foma 2102v, dan Motorola A920.

Versi Sistem Operasi Symbian bermula dari dikeluarkannya ponsel Ericsson R380 *smartphone* yang berbasis Symbian versi 5 yang merupakan versi *unicode* dari EPOC versi 5 dari Psion. Versi pertama yang menerapkan *platform* terbuka adalah Symbian OS versi 6.0 ditahun 2000 yang dipergunakan untuk ponsel Nokia 9210 dan 9290 *Communicator*. Kemudian dilanjutkan dengan versi 6.1 di awal tahun 2001 yang dipergunakan untuk ponsel *Series 60 Platform* yaitu Nokia 7650 dan 3650 *imaging phones*. Symbian OS versi 7.0 dikeluarkan pada tahun 2002 dan dipergunakan untuk ponsel Sony Ericsson P800 dan P900. Symbian OS versi 7.0s dikeluarkan tahun 2003 untuk ponsel-ponsel Nokia. Berikut merupakan daftar ponsel-ponsel yang berbasis sistem operasi Symbian.

| Nama Ponsel | Versi Symbian OS |
|-------------------------------|------------------|
| Nokia 6600 | 7.0s |
| Nokia 6620 | 7.0s |
| Nokia 7700 | 7.0s |
| Nokia N-Gage | 6.1 |
| Nokia 3660/3620 | 6.1 |
| Nokia 7650 | 6.1 |
| Nokia 3650/3600 | 6.1 |
| Nokia 9290 Communicator | 6.0 |
| Nokia 9210/9210i Communicator | 6.0 |

| | |
|--------------------|-----|
| Foma F2102V | 6.1 |
| Foma F2051 | 6.1 |
| Foma F900i | 6.1 |
| Sendo X | 6.1 |
| Siemens SX1 | 6.1 |
| Samsung SGH-D700 | 6.1 |
| BenQ P30 | 7.0 |
| Sony Ericsson P900 | 7.0 |
| Sony Ericsson P800 | 7.0 |
| Motorola A920 | 7.0 |

(Sumber: Symbian Website, <http://www.symbian.com/phones/index.html>)

2.2 Spesifikasi

Symbian OS adalah sistem operasi 32 bit, dengan konsep *little endian* dan berjalan pada beberapa tipe arsitektur mikroprocessor ARM. Symbian proses bekerja dengan prinsip *preemptive multitasking*. Dukungan terhadap *device-device* terintegrasi dalam *kernel* sebagai *kernel extension* yang ditulis dalam DLL (*dynamic linking library*) yang terpisah. *Kernel* berjalan dalam mode *privileged* dan memberikan servis ke aplikasi yang berjalan dalam mode *unprivileged* lewat *user library*. Symbian OS juga memberikan kumpulan-kumpulan *library* seperti *networking* (TCP/IP, PPP, FTP), *Communication* (*Bluetooth*, IrDA). Untuk mengakses servis-servis tersebut dengan menggunakan konsep hubungan *client-server*. *Client* menggunakan servis API yang diberikan oleh *server* untuk berkomunikasi dengan *server*. Semua hubungan komunikasi *client-server* diatur oleh *kernel*.

Symbian OS memiliki beberapa kelebihan diantaranya sebagai berikut:

- *Small*, kaya *feature*.
- *Platform* terbuka untuk aplikasi-aplikasi *third-party*.
- Konektifitas yang baik dengan perangkat lain.
- *Platform* yang berkembang.
- *High performance*, 32 bit OS dengan *preemptive multitasking*.
- *Long battery life*.
- Dukungan dan komitmen dari pembuat ponsel dunia.
- Aplikasi yang dapat dirancang sekali dan berjalan pada beberapa *device*.

Banyak *developer-developer* yang mengembangkan aplikasi-aplikasi untuk sistem operasi ini, didasarkan karena beberapa pertimbangan, diantaranya sebagai berikut:

- Symbian OS ditulis dalam C++, sistem operasi seluruhnya berbasis sistem *object oriented* sehingga *flexible*, *efisien*, *reuseability*, dan *extendability*.
- API (*Application Programming Interface*) yang jelas, memudahkan membuat aplikasi-aplikasinya.
- Mempunyai *multitasking* dan manajemen memori yang efisien.

- Proses berdasarkan *event driven* daripada *multithreaded*, sehingga menghemat memori untuk *context switching*.

Dalam masalah kehandalan, Symbian OS dirancang sedemikian rupa sehingga tidak terjadi kehilangan data dan *device* sangat jarang sekali *reboot*, karena Symbian OS mempunyai kemampuan sebagai berikut:

- Mencegah terjadinya memori *leak* dengan manajemen memori yang efektif
- Melepas sumber daya seketika sudah tidak digunakan lagi.
- Menangani dengan baik *error out of memory* dengan *error-handling framework* yang efektif

Symbian OS mempunyai beberapa *design* sesuai dengan *device family*-nya atau yang disebut sebagai DFRD(*Device Family Reference Design*). DFRD ini merupakan spesifikasi dalam hal *user interface* dan *hardware configuration*, sehingga banyak tipe-tipe ponsel berbeda yang memanfaatkan sistem operasi ini, Macam-macam DFRD yaitu:

- *Crystal*, untuk yang kaya *feature* seperti tipe *Communicator*, dimana mempunyai *full keyboard*, tampilan besar lcd mendatar dan menampung banyak informasi untuk kepentingan bisnis.
- *Pearl*, untuk *smartphone* dimana seperti ponsel standar dengan lcd kecil dan *keyboard* yang terbatas.
- *Quartz*, untuk ponsel dengan kemampuan *pen-based* atau *touch-screen* yang tidak memerlukan *keyboard*.

Sebagai sistem operasi untuk perangkat komunikasi bergerak, Symbian OS merupakan awal untuk ponsel masa depan. Berikut merupakan kemampuan-kemampuan penting yang dimiliki Symbian OS yang dapat menangani kebutuhan akan masa depan:

- Integrasi menyeluruh antar *contact info*, *messaging*, *browsing* dan telepon *wireless*.
- *Messaging* (*internet mail* dengan POP3,IMAP4,SMTP,MHTML), standar *attachment* termasuk *Microsoft word doc*, *Fax*, *Text messaging* dengan SMS.
- Protokol telepon bergerak (2G *voice* dan *circuit-switched data*, 2.5G *packet-switched data*, 3G, dan SMS).
- Protokol komunikasi (TCP/IP,WAP,Bluetooth,IrDA,serial)
- *Security* (enkripsi, *secure* protokol komunikasi termasuk HTTPS,WTLS dan SSL), *certificate-base install* aplikasi.
- *Engine* aplikasi (*contact,schedule,messaging,browsing,voice,office,utility* dan *system control*)
- *Object exchange* (OBEX untuk *appointment* dan *business card*)
- *Multimedia* (mendukung beberapa format audio,video dan image).
- Internasional lokalisasi (*unicode* karakter,*handwriting recognition*).
- *Programming* dan *content development* (C++,Java,WAP dan web)
- Sinkronisasi dengan PC
- Support beberapa tipe tampilan (*keyboard base* dan/atau *pen-base*)

2.3 Komponen

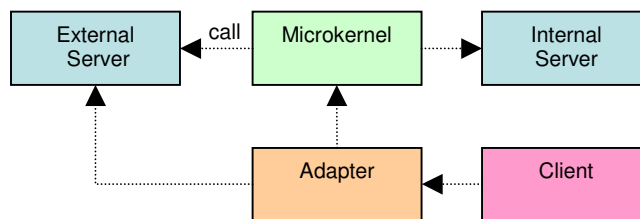
Symbian OS merupakan sebuah sistem operasi yang mengatur seluruh sumber daya yang ada didalam ponsel. Symbian OS disusun dari enam komponen, yaitu:

1. *Kernel*
2. *Middleware*
3. *Application Engine*
4. *UI Framework*
5. *Synchronization*
6. *JVM*

2.3.1 Kernel

Merupakan inti dari sistem operasi yang terdiri dari *device driver*, *data table*, dan program yang memungkinkan user berinteraksi dengan perangkat keras. *Kernel* merupakan program yang berjalan setiap waktu dan mengatur layanan-layanan yang diberikan ke user.

Symbian OS merupakan sistem operasi berbasis *microkernel* yaitu hanya elemen-elemen yang penting yang berada di *kernel* sedangkan fungsi yang lain ada dalam *middleware*, sehingga membuat *kernel* sangat ringkas dan arsitekturnya menjadi lebih modular. Dari awalnya Symbian OS merupakan sistem operasi 32-bit yang mendukung *multitasking* dan *multithreading*. Ukuran *microkernel* sekitar 5% dari keseluruhan sistem operasi, yang berkisar antara 500kB sampai 15MB tergantung pada ada tidaknya dukungan java dan aplikasi-aplikasi lain yang ikut diinstal. Pemisahan inti dan komponen lain membuat sistem sangat modular, yang akan meningkatkan *portabilitas platform* dan membuat proses *upgrade* dan perubahan *platform* lebih mudah dilakukan. Berikut merupakan arsitektur *microkernel* Symbian OS.



(Sumber: Digia Inc., "Programming for the Series 60 Platform and Symbian OS", p37)

Komponen *microkernel* menerapkan *atomic service* yang diperlukan keseluruhan aplikasi di sistem, mengontrol sumber daya seperti memori, proses, *thread*, dan IPC. Fungsi yang tidak mungkin dimasukkan ke *kernel* karena alasan kompleksitas dan besar, dipisah ke *internal server*. *Internal server* mengembangkan fungsi inti, misalnya untuk menangani *graphic* dan media penyimpanan, dapat mempunyai proses sendiri atau *share library*. *External server* menggunakan servis dari *microkernel* dan *internal server* untuk memberikan servis ke *client*. *External server* menangani komunikasi (*serial comm server*, *socket server*, *message server*, *telephony server*), *graphics* (*window*, *font*, *bitmap server*), *audio*(*media server*), *storage media* (*file server*). *External server* dieksekusi pada proses tersendiri, setiap *external server* memberikan *client side API* yang membungkus IPC

antara *client* dengan *server*. Sedangkan *Adapter* memberikan *interface* yang transparan untuk *client* seperti detail komunikasi.

2.3.2 Middleware

Merupakan kumpulan *library*, *data storage*, dan program yang mengimplementasikan sistem servis. Kesemuanya itu tidak perlu diletakkan dalam *kernel*. Manajemen data, komunikasi dan *graphics* termasuk servis sistem tersebut. Sebagai contoh *window system* yaitu yang mengatur bagaimana user berinteraksi dengan perangkat keras, hal ini tidak cukup penting untuk diletakkan didalam *kernel*.

Symbian OS menggunakan *server* untuk implementasi *middleware*. Idenya adalah *server* yang dapat mengatur servis dari beberapa *client* dan merespon permintaan-permintaan tersebut. Dengan membuat sebuah *layer* baru untuk *middleware*, *designer* dapat dengan mudah merancang sistem servis baru tanpa harus merubah *kernel*.

2.3.3 Application Engine

Aplikasi *user level* dapat memanfaatkan servis yang diberikan pada *level middleware* yaitu dengan menggunakan *application engine*. *Application engine* yang melakukan koordinasi untuk mengakses sumber daya yang tidak begitu penting. *Application engine* diantaranya adalah sebagai berikut:

Agenda engine, *Contact engine*, *Sheet engine*, *Alarm server and WorldTime engine*, *Spell engine*, dan *Help engine*.

2.3.4 User Interface Framework

User interface merupakan faktor utama untuk sebuah ponsel, yang menyebabkan ponsel mudah digunakan, mudah dirubah dan diprogram. Symbian OS menerapkan *framework* untuk *user interface* sehingga mudah diadaptasi. Dalam Symbian OS ada dua *user interface* dalam *frameworknya* yang menggunakan komponen yang umum seperti kontrol dan dialog yaitu Uikon dan Eikon.

2.3.5 Synchronization

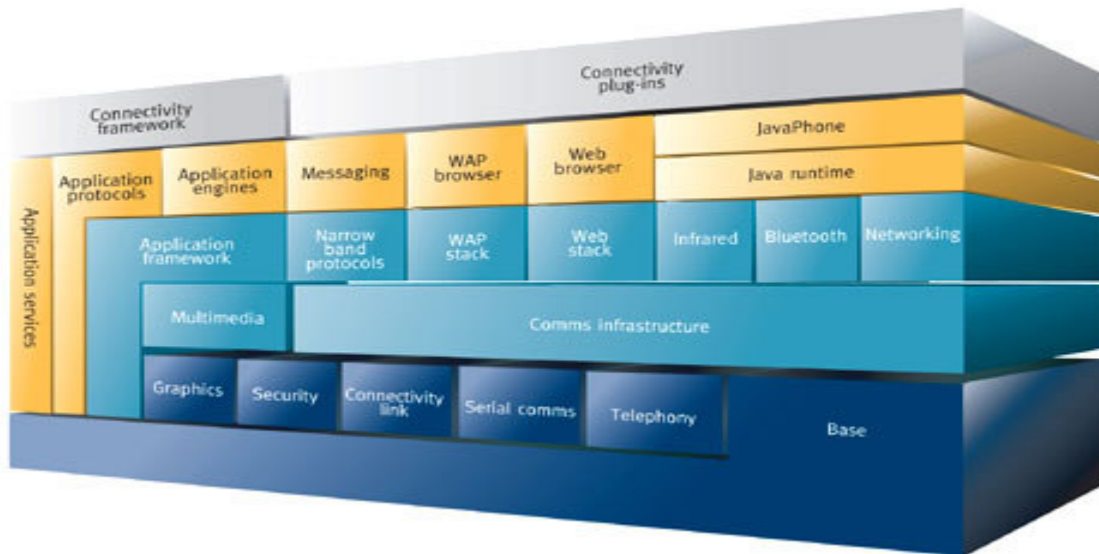
Sinkronisasi dengan peralatan lain ataupun dengan komputer desktop menjadi perhatian dalam teknologi ponsel. Symbian OS menerapkan 3 bagian untuk sinkronisasi yaitu *Connection manager* yang melakukan inialisasi koneksi antar *device* dan mendeteksi jikalau ada *device* yang ingin melakukan koneksi. *Connectivity server* yang mengimplementasi sesi sinkronisasi seperti *file browsing*, *file sinkronisasi*, *back and restoration*. *File Converter* yang melakukan transfer data antar format yang berbeda.

2.3.6 JVM

Symbian OS mengimplementasikan teknologi java yang dikenal dengan J2ME. JVM merupakan salah satu komponen dalam Symbian OS yang untuk perangkat ponsel

dikenal dengan KVM (*kilo virtual machine*). *JavaPhone* dan *PersonalJava* yang merupakan bagian dari J2ME *specification* juga diimplementasikan dalam Symbian OS.

2.4 Arsitektur



(Sumber: Symbian Website, <http://www.symbian.com/technology/symbos-v6x-det.html>)

Gambar diatas menunjukkan Arsitektur Symbian OS versi 6.x. *Subsystem* yang lebih atas mempunyai ketergantungan dengan beberapa *subsystem* yang dibawahnya, walaupun tidak selalu demikian. Sebagai contoh, *WAP stack*, *Bluetooth*, *Infrared* dan *Narrow band* merupakan protokol-protokol yang ada dalam *comms infrastructure*. *Messaging* juga bergantung pada *WAP stack*, tetapi java tidak bergantung dengan *infrared*.

Komponen-komponen dalam arsitektur diatas selengkapnya sebagai berikut:

- *Base*, Terdiri atas Sistem *runtime* yang sangat dasar, *low-level security*.
- *Application framework*, Terdiri atas API untuk manajemen data, *text*, *clipboard*, *graphics*, *internationalization*, dan inti komponen GUI.
- *Multimedia*, Terdiri atas *audio recording* dan *playback*, fungsi-fungsi yang berhubungan dengan *image*.
- *Communication infrastructure and network stacks*, Terdiri atas stack komunikasi yang luas termasuk TCP/IP, GSM, GPRS dan WAP. Komunikasi Personal seperti *infrared*, *Bluetooth* dan *serial*.
- *Messaging*, Terdiri atas *internet mail*, SMS dan Fax.
- *Browsing*, Terdiri atas WML dan HTML *browsing engine*.
- *Application protocols, service and engines*, Terdiri atas *engine* untuk manajemen *contact*, *schedule* dan *to-do list* manajemen dan aplikasi-aplikasi yang lain.

- *Java*, Terdiri atas PersonalJava 3.0 spec JVM-based java runtime system dengan JavaPhone 1.0 APIs.
- *Connectivity*, Terdiri atas *converter* dan *viewer* untuk format data *foreign* termasuk *attachment mail* Microsoft word. *Framework* komunikasi untuk berhubungan dengan PC.
- *Tools*, Terdiri atas *tool* untuk membuat aplikasi, ROMs dan untuk *debug target* aplikasi.

2.4.1 Base

Kernel dan User Library

Komponen E32 menggabungkan *kernel* ekern.dll dan *user library* euser.dll. *Kernel* berjalan dalam mode *privileged*, memiliki *device driver*, melakukan manajemen daya, alokasi memori untuk dirinya dan mode user yang mempunyai proses *unprivileged*. *Kernel* berjalan secara dasar pada ARM *microprocessor*.

User library memberikan servis ke program user berupa:

- Proses, *thread*, program dan manajemen memori.
- *Error handling* dan *cleanup framework*.
- *Descriptors*: string dan buffer data biner.
- *Class container*: array dan lists.
- *Active object*: untuk *even driven multitasking* tanpa memerlukan *multithreading*.
- *Client server architecture*: untuk simple dan efisien IPC.
- HAL (*Hardware abstraction layer*): memberikan *interface* yang konsisten untuk tipe perangkat keras yang berbeda.
- Lokalisasi: *currency*, *time* dan *date format*.
- *Miscellaneous*: seperti timer.

Base peripherals

File server yang mendukung VFAT, ROM dan Sistem file Flash (*log Flash file system*) dalam *internal chip*, CF card dan MMC card. VFAT *file sistem* dengan mode 'rugged' yang memberikan proteksi terhadap daya *loss*.

Security

Modul dasar *security* adalah *cryptography module* dan *certificate management module*. *Security* termasuk algoritma standar *cryptography*, *hash key generation*, *random number generation* dan *certificate management*.

Cryptography module termasuk didalamnya sbb:

- *Raw cryptography algorithm*: untuk enkripsi dan dekripsi simetris seperti DES, 3DES, RC2, RC4, RC5 dan asimetris seperti RSA, DSA, DH.
- *Hash function*: MD5, SHA, HMAC.
- *Random number generator* sebagai basis untuk *key cryptography*.

2.4.2 Application framework

Text

Mendukung *Unicode* standar versi 3.0, semua 16-bit karakter *unicode* dapat digunakan dalam text. Semua karakter disimpan dan diambil sesuai aslinya, *diimport* dan *diexport* sebagai *plain text*. Mendukung perubahan warna text seperti pada PC. Gambar dapat diletakkan disamping text dengan pengaturan *transparency* dan *background scrolling*. Dua level *undo/redo* yang dapat disimpan. Mendukung perubahan *size*, *shape*, *color* dan *blink rate* kursor.

Internationalization

Mendukung Chinese dan Japanese karakter untuk text input menggunakan *handwriting recognition* atau *keyboard*. Perubahan antara *unicode* dan karakter set lain melalui mekanisme *plug-in*, karakter yang diimplementasikan diantaranya: UTF-7, UTF-8, *modified UTF-7*, *modified UTF-8*, 7 bit SMS (atau 7-bit GSM), *Code Page 1252*, dan ISO 8859-1. Nama file VFAT *filename* mendukung *8.3 name format*.

Uikon

Inti Komponen GUI Symbian OS termasuk didalamnya *dialog framework*, *concrete control*, *GUI environment*, perubahan *look and feel*. GUI ini fungsi-fungsinya diberikan oleh EIKON dalam versi 5 Symbian OS.

2.4.3 Multimedia

Memberikan kemampuan *recording*, *playback audio* dan fungsi-fungsi yang berhubungan dengan *image*. *Audio framework* terdiri atas *share library* yang dapat melakukan pembacaan dan penulisan terhadap *format audio* yang umum seperti WAV, AU, WVE dan RAW dalam format yang berlainan. *Image framework* terdiri atas *share library* yang dapat melakukan pembacaan dan penulisan terhadap format-format gambar yang umum seperti JPEG, BMP, MBM and GIF (*read only*), WBMP (*read only*) dan *Smart Messaging images (read only)*. *Server interface* memberikan *plug-in* yang *generic*, *plug-in audio local* dan *telephony 8/16 bit PCM*, ALaw, DTMF, *tone* dan *tune generator*. Format baru audio dan image dapat ditambah pada saat *runtime* dengan menambah *plug-in library*-nya.

2.4.4 Communication infrastructure and network stacks

Networking

Protokol TCP/IP untuk dapat melakukan koneksi lewat internet yang digunakan oleh aplikasi seperti email dan web.

GSM telephony & communications

Telephony framework yang memberikan *interface* dasar untuk GSM voice, data dan fax. Symbian OS versi 6.1 juga mempunyai kemampuan mendukung GSM *phase 2+*

SIM application toolkit, Class 3 (ETSI 11.14 R98), dengan kombinasi class 'a' (*mobile phones* yang hanya mendukung satu SIM) dan class 'b'.

GPRS data communications

Versi 6.1 memperkenalkan dukungan GPRS (*General Packet Radio Service*) class *B phones*. Dengan fungsi kelas B ini maka ponsel dapat melakukan hubungan telepon lewat GSM bersamaan dengan penggunaan GPRS, jika Paket data protokol aktif, servis GPRS akan otomatis *suspend* dan *resume*. Class B sekarang ini didukung oleh banyak jaringan GPRS.

WAP stack

Peningkatan yang penting dalam versi 6.1 adalah dukungan untuk WAP 1.2.1, fungsi *push* dan GPRS sebagai *bearer*. *WAP stack* mendukung protokol spesifikasi versi 1.1 dan 1.2.1 class C dari WAP Forum. *WAP stack* dapat menggunakan *bearer* GSM CSD dan GPRS UDP untuk koneksi *browsing*, GSM CSD, GPRS UDP, GSM SMS dan GPRS SMS untuk *connectionless push*. *WAP stack* mempunyai layer WSP (*session protocol for WAP*), WTP (*transaction protocol for WAP*), WTLS (*transport layer security protocol for WAP*) dan WDP (*datagram protocol for WAP*).

Bluetooth stack

Bluetooth diimplementasikan sesuai spesifikasi versi 1.0 *Bluetooth system architecture*. *Bluetooth stack* mengimplementasi penuh *Generic Access Profile*, *Serial Port Profile* dan *General Object Exchange Protocol*. *Stack* terdiri atas *protocol module*, *security manager*, *communications server module* dan *Service Discover Protocol server module*.

Infrared

Infrared IrDA stack berada dalam modul protokol soket *server* (*irda.prt*) yang menerapkan IrDA layer IrLAP v1.1, IrLMP v1.1 dan IrTinyTP.v1.1. Symbian OS Versi 6.0 dan yang terbaru, meningkatkan fungsi *infrared* dengan menambahkan *feature* seperti *slow infrared* (SIR) dengan *throughput* 9.6 Kbps to 115.2 Kbps, IrOBEX v1.0 (*object exchange*), IrTRANP v1.0 (varian dari fungsi GET/PUT IrOBEX). APIs IrCOMM v1.0 mendukung fungsi fax/modem yang diimplementasikan dalam *serial communications server module*.

2.4.5 Messaging

Messaging framework mendukung pengiriman dan penerimaan pesan SMS, email dan fax. *Framework* memanfaatkan *polymorphic MTMs* (*message type modules*) untuk menangani tipe pesan yang spesifik. Perubahan besar pada symbian 6.0 adalah penambahan *watchers* yang menangani pesan masuk dan *BIO messaging* yang mendukung pengiriman pesan ke system daripada ke user. Symbian versi 6.1 menambahkan dukungan GPRS dan 2D kompresi fax.

2.4.6 Browsing

Web engine

Arsitektur *web engine* dibagi dalam beberapa komponen inti yaitu: *Rendering engine* yang bertanggung jawab untuk *rendering* dokumen dan menjaga struktur pages. *Services engine* yang memberikan fungsi umum, seperti *network status monitoring*, *bookmark*, *history list*, *proxy and authentication support*. Web control yang bertanggung jawab menampilkan *actual pages*.

Web engine mendukung HTTP/1.1 dalam RFC2068, juga dapat mengakses *secure web sites* HTTPS dengan SSL 3.0 and TLS 1.0. Dengan Symbian OS Version 6.1, Web browser berjalan secara transparan melalui koneksi jaringan GPRS.

WAP browsing engine

Browse engine mendukung WML 1.1 dan WML 1.2.1 juga WMLScript. Symbian OS Versi 6.1 juga mendukung PRE element. Engine tidak mendukung fungsi pilihan seperti *access key* sebuah elemen, mode *connectionless* HTTP ke WSP, WTA, vCard dan vCalendar, *class attributes*, *fieldset*.

2.4.7 Application protocols, services and engines

Application engines

Yang menjadi bagian *application engine* adalah: *agenda engine*, *contacts model*, *sheet engine*, *alarm and world server*, *spell engine* dan *help engine*.

Application services

Application service merupakan gabungan komponen-komponen yang digunakan oleh *application engine*. Ada beberapa servis, diantaranya: *Task scheduler*: menjadwalkan pemanggilan aplikasi atau melakukan inisialisasi *feature application*. *System agent*: memberikan informasi status *mobile phone* dan *interface* fisiknya. *log engine*: menyimpan penggunaan phone terutama *telephony* dan *messaging*. *Alarm and world server*: memberikan fungsi alarm, *sound playing*, *country codes*, *world country and city information database*.

Software installation

Memberikan proses instalasi yang cepat dan mempunyai keamanan yang baik. Instalasi bisa berasal dari PC ataukah dari ponselnya dengan melakukan instal paket file dalam bentuk .sis. Paket dapat berupa aplikasi program, *engine*, dan *user interface*. Symbian OS mempunyai daftar aplikasi yang telah terinstal sehingga dapat dengan mudah untuk dilakukan proses *reinstall* ataupun *uninstall*.

2.4.8 Java

Java

Java pada Symbian OS versi 6.x merupakan implementasi *PersonalJava application environment specification 1.1.1a*. Spesifikasi tersebut merupakan *PersonalJava 3.0.x hybrid reference* yang berdasarkan pada Java 1.1.6. Symbian OS memberikan implementasi lengkap PersonalJava, namun tanpa RMI dan JDBC.

JavaPhone

Komponen JavaPhone memberikan kumpulan APIs yang mengembangkan runtime PersonalJava untuk mengakses fungsi *native* yang penting terutama *telephony*, *agenda*, *contact* dan *power monitoring* dan juga serial komunikasi. Symbian OS menerapkan *JavaPhone 1.0 reference*.

2.4.9 Connectivity

Onboard converters

Symbian OS Version 6.x memberikan fungsi *converter* antara Symbian OS and *Windows formats* untuk *spreadsheet and word documents*, dan dari Symbian OS *Rich Text* ke HTML.

Connection manager

Mengatur koneksi antara PC dan Symbian ponsel. Koneksi dapat melewati serial, infrared dan pada versi 6.1 dapat digunakan bluetooth.

Symbian Connect

Merupakan program symbian pada PC untuk melakukan komunikasi dengan Symbian OS, pada versi 6.1 dapat mendukung Microsoft Windows ME.

2.4.10 Tools

Software development

Tool dasar untuk membuat program-program dengan C++ dan Java dan untuk membuat ROMs dengan tujuan mikroprosesor ponsel yang menggunakan ARM4 atau *Thumb binaries* yang efisien memproses *unicode*. Dapat pula membuat program untuk target *ARMI binary format* dimana dapat berjalan baik pada Thumb atau ARM4 ROMs

In-target debugging

Pada Symbian OS Version 6.1 untuk melakukan debug pada target mesin dengan menggunakan *GNU Debugger GDB*. Debug ini hanya bisa untuk *user-mode programs*.

2.5 Manajemen Proses

Proses merupakan unit memori yang terproteksi, sedangkan *Thread* merupakan unit eksekusi. Satu proses tidak dapat mengakses secara langsung memori dari proses lain, kecuali dibuat memori yang global. Setiap aplikasi dan server di symbian mempunyai prosesnya sendiri-sendiri, tetapi user juga dapat membuat proses baru. Semua proses dapat mengakses *share library* dan sistem ROM. Dalam mode user, dapat dibuat 4 prioritas proses, yaitu:

- EPriorityLow=150
- EPriorityBackground=250
- EPriorityForeground=350
- EPriorityHigh=450

Prioritas tersebut digunakan untuk menghitung keseluruhan prioritas sebuah thread yang dibuat dalam proses. Developer dapat juga menggunakan nilai absolut prioritas dalam hal ini prioritas proses tidak digunakan. Ada 5 level prioritas yang tersedia untuk thread yang dieksekusi di mode user, yaitu:

- EPriorityMuchLess
- EPriorityLess
- EPriorityNormal
- EPriorityMore
- EPriorityMuchMore

Kernel mempunyai prioritas tertinggi untuk proses dan thread. *Thread* dijadwalkan berdasar pada prioritasnya atau *round robin* jika thread mempunyai prioritas yang sama. Penjadwalan bersifat *preemptive* yang berarti yang mempunyai prioritas tinggi dapat menginterupsi thread yang lebih rendah. Pada beberapa kasus *preemptive multitasking* tidak diperlukan, dan dapat digantikan dengan *multitasking* yang *cooperative* yaitu prinsip *active object*.

Seperti telah disebutkan diatas ada dua mode eksekusi yaitu *user mode* dan *kernel mode*. *Kernel mode* mempunyai prioritas yang tertinggi dibandingkan proses dan thread dalam *user mode*. Kedua mode tersebut diatur dalam dua *library* yaitu EUser dan EKern. Kedua *library* tersebut menyediakan kumpulan servis-servis untuk aplikasi. EUser digunakan untuk menangani proses dan thread, manajemen memori, *active object*, dan sebagainya. EKern digunakan untuk mengakses *device driver* dan sebagainya.

2.6 Manajemen Memori

Setiap proses harus mempunyai setidaknya satu thread. Proses yang kosong tidak dapat dieksekusi, tetapi thread dapat dieksekusi. Thread yang baru dibuat dengan fungsi Create(), dimana prototipenya sebagai berikut:

```
TInt Create ( const TDesC& aName, TThreadFunction aFunction, TInt
             aStackSize, TInt aHeapMinSize, TInt aHeapMaxSize,
             TAny *aPtr, TOwnerType aType=EOwnerProcess);
```

Untuk membuat thread diperlukan memori stack sebesar 8kB defaultnya, sedangkan untuk memori *heap* minimum 256Bytes sampai batas maksimum memori

yang tersedia. Ketika thread dibuat, *chunk* memori baru dialokasikan untuk thread tersebut. *Chunk* merupakan area memori di *virtual* memori yang mempunyai alamat yang berdekatan. Secara fisik, *chunk* terdiri atas alokasi memori yang dibutuhkan tergantung pada arsitektur, 4kB jika pada arsitektur ARM. Bagian bawah dari *chunk* adalah *stack* dan di atasnya adalah *heap*. *Stack* selalu tumbuh kebawah, jadi tidak akan pernah mencapai memori *heap*. *Heap* dapat *dishare* antara thread dengan menggunakan versi fungsi `Create()` yang berbeda.

Heap dan *stack* memainkan peranan yang berbeda dalam penyimpanan object yang dibuat dan data yang lain. Yang harus diperhatikan dalam penggunaan *stack* adalah *stack* mudah terjadi *overflow*. Semua alokasi di *stack* secara otomatis di hapus ketika tidak lagi dibutuhkan, namun untuk object yang dialokasikan di *heap* harus ada *pointer* untuk mengalokasikannya dan untuk menghapusnya. *Class RHeap* memberikan beberapa fungsi untuk menghitung banyaknya alokasi di *heap* atau menentukan keseluruhan yang teralokasi di *heap*.

Semua object yang berada dalam *heap* harus mempunyai tipe class C, yang berarti berasal dari turunan kelas yang umum yaitu CBase. Semua yang berasal dari CBase dapat ditaruh dalam *cleanup stack* dan dihapus dengan fungsi `PopAndDestroy()`. Dapat juga digunakan `User::Alloc()` untuk melakukan alokasi di memori *heap*. Dikarenakan *heap* tidak secara otomatis dilakukan proses penghapusan, maka ada resiko *heap* akan menyebabkan memori *leak*. Untuk itu dalam Symbian OS digunakan *cleanup stack* untuk menyimpan variabel secara otomatis menangani penghapusannya sehingga tidak terjadi memori *leak*.

2.7 Manajemen I/O

Dalam permasalahan input dan output, baik itu berupa *keyboard input*, *pointer input*, dan *I/O request* yang lain, semua berjalan dalam mode *asynchronous* yang diberikan oleh *asynchronous service providers*. Symbian OS memberikan mekanisme yang mudah untuk menggunakan *asynchronous service* tersebut.

Ada 2 level *asynchronous* yaitu: *Low-level* dan *High-level asynchronous*.

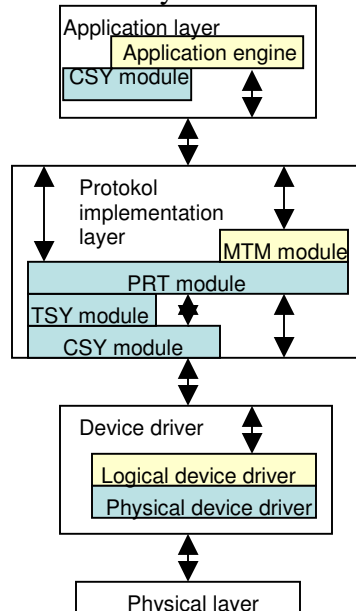
Pada *low-level*, *asynchronous service provider* memberikan servis dengan fungsi *request* dan *cancel*. Status *request* akan disimpan kedalam *request status*, dan *service provider* akan memberikan signal bahwa *request* telah selesai dengan menggunakan *thread request semaphore*.

Pada *high-level*, menggunakan *active scheduler* yang melakukan *wait loop* yang digunakan dalam *multiple asynchronous*, dan *active object* yang melakukan fungsi *request* dan *cancel*, sekaligus menangani penyelesaian permintaan.

2.8 Komunikasi

Ponsel tidak akan bermakna tanpa ada pertukaran informasi, untuk melakukan hal tersebut diperlukan suatu teknologi komunikasi, baik berupa teknologi *telephony* maupun teknologi pertukaran data yang lain.

Struktur *microkernel* Symbian OS mempunyai efek terhadap arsitektur komunikasi. Service komunikasi harus melewati sistem *server*, yang melakukan penambahan atau pengurangan tergantung perangkat keras yang didukung oleh *smartphone*. Arsitektur komunikasi terdiri atas server komunikasi dan modul tambahan. Modul tambahan dapat ditambah dan dibuang kapanpun pada saat *runtime* tanpa harus melakukan *rebooting* OS. Berikut ini adalah arsitektur komunikasinya.



(Sumber: Digia Inc., "Programming for the Series 60 Platform and Symbian OS", p310)

2.8.1 Server komunikasi

Server komunikasi dalam Symbian OS memberikan *client-side* API, ada empat server komunikasi, yaitu:

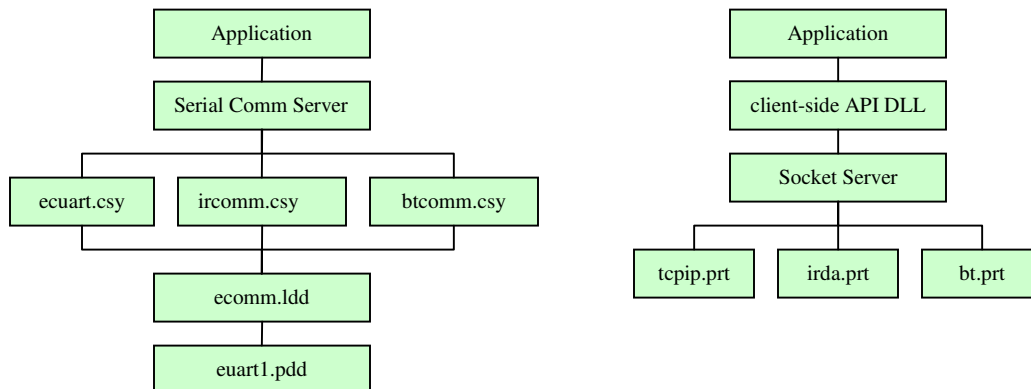
- Server serial komunikasi (C32) yang memberikan interface umum untuk komunikasi serial.
- Server soket (ESock) yang memberikan interface umum untuk komunikasi *endpoint* yang lebih dikenal dengan soket.
- Server *Telephony* (ETel) yang memberikan interface umum untuk inisialisasi, kontrol dan memutuskan *telephone call*.
- Server *Message* (MTMs) yang memberikan akses ke *data message*.

Penggunaan server komunikasi tergantung pada aplikasi, jika aplikasi menghendaki pengiriman bit sederhana maka dapat digunakan *serial server*. *Serial server* menggunakan protokol *dependen* yang mempunyai ekstensi *.csy*. Hal ini yang membedakan antara modul protokol RS-232 dengan infrared. CSY mempergunakan *device driver* di *kernel* untuk mengakses perangkat keras. *Device driver* pun merupakan modul tambahan yang dapat di *load* dan *unload* kapanpun.

2.8.2 Modul komunikasi

Device driver dibagi menjadi dua bagian dalam Symbian OS yaitu *physical* dan *logical device driver* untuk meningkatkan *modularity* dan *reusability*. PDD (*physical device driver*) berhubungan langsung dengan perangkat keras, sedangkan LDD (*logical device driver*) bertanggung jawab terhadap *buffer data*, *control flow*, DFC (*delay function call*) dan *interrupt handling*. Interupsi *device* ditangani dalam dua tahap, ketika interupsi terjadi, pertama akan menjalankan servisrutin yang akan mengenali *device* kemudian akan men-set flag di *kernel* untuk memanggil DFC. Selanjutnya DFC akan dipanggil dalam *user mode*.

Untuk serial komunikasi dalam symbian, PDD-nya adalah `euart1.pdd` an LDD-nya adalah `ecomm.ldd`. Perubahan hanya terjadi pada protokol modul yaitu `ecuart.csy` untuk RS-232, `ircomm.csy` untuk infrared, dan `btcomm.csy` untuk bluetooth. Protokol digunakan sebagai fungsi untuk *error detection*, *error correction*, efisiensi dan adaptasi *flow control*, dan mendukung beberapa koneksi simultan. Implementasi protokol dalam Symbian OS di akses melalui socket server ESocket, yang mirip dengan interface BSD socket. Protokol high-level mulai dari network hingga *application layer* di terapkan dalam modul terpisah yang dinamakan *protocol modules* dan mempunyai ekstensi `.prt`, misalkan `tcpip.prt` adalah modul protokol TCP/IP, `irda.prt` adalah modul protokol IrDa, `bt.prt` adalah modul protokol Bluetooth dan `wapprot.prt` adalah untuk WAP. Berikut ini merupakan gambar level modul komunikasi dan modul protokol dalam Symbian OS.



(Sumber: Digia Inc., "Programming for the Series 60 Platform and Symbian OS", p314)

Modul komunikasi untuk telepon adalah TSY modul yang memberikan standar fungsi telepon seperti *establishing*, *controlling* dan *terminating call*. Modul TSY digunakan oleh *server telephony* ETel. Modul `phonetsy.tsy` yang memberikan semua service untuk keperluan telephony. Selain standar juga ada modul yang lain seperti `hayes`, `gsmbsc`, dan `gprstsy`.

Modul komunikasi untuk menangani *creation*, *sending*, *receiving* dan *editing message* adalah MTMs. Modul message ini dapat menangani *email message* seperti (SMTP, POP3 dan IMAP4), dapat pula menangani FAX, SMS dan MMS. Untuk mengirimkan message dalam aplikasinya digunakan `CSendAs` class yang melakukan pengelompokan data message tanpa menggunakan interface MTM secara langsung.

Object hasil dari CSendAs akan membuat session ke *message server* dan mengambil registry MTM sehingga dapat memanfaatkan servis-servis-nya.

Teknologi komunikasi dan protokol yang didukung Symbian OS

| Layer | Serial | IrDA | TCP/IP | WAP | Bluetooth | Telephony |
|--------------|--------|--------------|--------------|---------|--------------|------------------------|
| Application | | | | WAE | | |
| Presentation | | | | WSP | | |
| Session | | | | WTP | | |
| Transport | | TinyTP/IrMux | UDP,TCP | WTLS | | |
| Network | | IrLMP | IP,ICMP | WDP | L2CAP/RFCOMM | |
| Data link | RS-232 | IrLAP | Ethernet,PPP | Bearers | Baseband | Telephony |
| Physical | UART | Infrared | | | BT radio | Wired,mobile telephone |

(Sumber: Digia Inc., "Programming for the Series 60 Platform and Symbian OS", p333)

2.8.3 Keamanan komunikasi

Ada tiga jenis ukuran keamanan dalam berkomunikasi yaitu *confidentiality*, *integrity* dan *availability*. *Confidentiality* berarti tidak ada data yang bisa didapat oleh orang yang tidak berkepentingan. *Integrity* berarti tidak ada data yang bisa diubah oleh orang yang tidak mempunyai hak akses. *Availability* berarti data dan service selalu tersedia untuk yang menginginkannya, tidak boleh terjadi serangan *denial of service*.

Mekanisme proteksi dapat dilakukan dengan beberapa cara, diantaranya untuk meningkatkan tingkat *confidentiality* dapat dilakukan dengan melakukan enkripsi, walaupun tidak ada algoritma enkripsi yang tidak bisa dipecahkan, tetapi tingkat kesulitan memecahkannya yang perlu dipertimbangkan. Untuk meningkatkan *integrity*, dapat digunakan *message certificate* seperti *digital signature*. *Availability* dapat ditingkatkan dengan pengecekan keaslian dengan pengecekan *password*.

Symbian OS mendukung berberapa algoritma enkripsi diantaranya yaitu RSA, DES, 3DES, RC2, RC4, SHA-1, MD5, dan Diffie-Hellman. Keamanan dalam tingkat protokol, dapat digunakan *secure shell* (SSH) untuk *remote system* yang akan melakukan enkripsi *password* dan data. Dalam WAP (*wireless application protocol*) diberikan layer terpisah untuk keamanan yaitu WTLS (*Wap transfer layer security*). Ketika menggunakan socket, dapat dipergunakan TLS (*transfer layer security*) dan SSL (*secure socket layer*).

Dalam tool Symbian OS, ada *generator* untuk membuat *private-public key* yang merupakan *asymetric cryptography* dan dapat mengeluarkan permintaan *certificate*. *Private key* dipakai untuk tanda digital instalasi file, sehingga *system installer* dapat mengenalinya. Proses instal yang aman pertama dengan mengecek tanda pada file yang telah terinstal dengan menggunakan *public key* dalam *certificate developer* untuk meyakinkan bahwa paket instalasinya telah tertanda *private key developer*. Kemudian mengecek tanda pada *certificate developer* dengan organisasi *public key* dari organisasi *certificate* untuk meyakinkan bahwa pasangan kunci tersebut adalah kepunyaan seseorang yang ada dalam *certificate* tersebut.

Ada permasalahan dalam keamanan, bahwa tidak akan ada metode keamanan yang berguna jika salah satu software tidak menerapkannya. Misalnya dalam penggunaan protokol bluetooth untuk koneksi ke *service provider* yang tidak menerapkan keamanan, maka dapat terjadi data-datanya tidak bisa dipercaya keasliannya. Dalam symbian untuk mengecek keabsahan suatu *password* dari *input external* harus dicek isinya dan besarnya dengan menggunakan software, dapat diterapkan dengan menggunakan *ASSERT macro*.

3. Kesimpulan

Ponsel yang *smart* atau *smartphone* dituntut adanya sebuah sistem operasi yang stabil dan handal, tidak boleh terjadi kerusakan sistem dan proses *rebooting* walaupun mempunyai memori dan sumber daya yang terbatas. Symbian OS menjadi sistem operasi yang diperhitungkan untuk ponsel-ponsel generasi sekarang dan yang akan datang, selain karena kekayaan *feature*-nya namun juga didukung oleh para industri pembuat ponsel.

Symbian OS memiliki arsitektur yang modular dengan tingkat *dependency* yang rendah sehingga mempunyai tingkat *reusability* dan *extendability* yang tinggi. Model *microkernel* yang diterapkannya menjadikan Symbian OS lebih sederhana namun *powerfull*. *Platform* terbuka, bahasa pemrograman yang berbeda untuk membuat aplikasi-aplikasi yang berjalan di atasnya, membuat Symbian OS banyak didukung oleh para *developer-developer* aplikasi dunia.

Manajemen proses, memori, dan I/O yang efisien dan sistematis membuat sistem operasi ini dapat mengatasi kehilangan data, kekurangan memori dan sumber daya lainnya, juga dapat menangani banyak proses dan *request input/output* secara simultan. Penerapan keamanan dalam berbagai macam komunikasi yang didukungnya sehingga dapat mengikuti perkembangan teknologi menjadikan Symbian OS dapat dipakai untuk berbagai keperluan yang memerlukan tingkat *privately*, *confidentiality*, dan *integrity* data yang tinggi ataupun tidak.

Daftar Pustaka

- Michael J Jipping, "Symbian OS Communications Programming", John Wiley & Sons, 2002
- Digia Inc., "Programming for the Series 60 Platform and Symbian OS", John Wiley & Sons, 2003
- Nokia, "Symbian FAQ versi 2.0", Nokia, April 2002
- Symbian: Symbian OS phones, <http://www.symbian.com/phones/index.html>, 23 februari 2004
- Symbian: Technology: Symbian OS v6.x Functional Description, <http://www.symbian.com/technology/symbos-v6x-det.html>, 23 februari 2004
- Symbian: Technology: Symbian OS v6.x product sheet, <http://www.symbian.com/technology/symbos-v6x.html>, 23 februari 2004
- Symbian: About us: Symbian OS, <http://www.symbian.com/about/symb-os.html>, 23 februari 2004
- Symbian: About us: Company history, <http://www.symbian.com/about/history.html>, 23 februari 2004

Lampiran

Gambar Ponsel-ponsel dengan Symbian OS

| | | | |
|---|---|--|---|
|  |  |  |  |
| Nokia 6600 | Nokia 6620 | Nokia 7700 | Nokia N-Gage |
|  |  |  |  |
| Nokia 3660/3620 | Nokia 7650 | Nokia 3650/3600 | Nokia 9290 |
|  |  |  |  |
| Nokia 9210 | Foma F2102V | Foma F2051 | Foma F900i |
|  |  |  |  |
| Sendo X | Siemens SX1 | Samsung SGH-D700 | BenQ P30 |

| | | |
|---|---|---|
|  |  |  |
| Sony Ericsson P900 | Sony Ericsson P800 | Motorola A920 |

(Sumber: Symbian Website, <http://www.symbian.com/phones/index.html>)