LAMPIRAN A
BLUETOOTH™

A.1. Teknologi *Bluetooth*™

Bluetooth™ merupakan teknologi nirkabel yang awalnya dirintis oleh *Bluetooth*™ *Special Interest Group (SIG)* yang melibatkan Ericsson, IBM, Intel, Toshiba, Lucent, 3Com, Motorola, dan Nokia. Teknologi ini bertujuan mengembangkan standar radio digital untuk hubungan jangkah pendek antara beberapa alat yang berbeda, baik dalam kantor maupun lingkungan rumah tangga, misalnya antara telepon genggam, laptop, printer, faksimili, bahkan peralatan elektronik rumah tangga juga dapat memanfaatkan hubungan nirkabel *Bluetooth*™ ini.

Gambar A.1. Logo *Bluetooth*™

Teknologi *Bluetooth*™ merupakan sebuah solusi sistem yang terdiri dari perangkat keras, perangkat lunak, dan persyaratan *interoperability* agar dapat kompatibel dengan berbagai peralatan lainnya dan dapat dioptimalkan sesuai dengan aplikasi tertentu, secara umum inti konsep *Bluetooth*™ adalah bahwa peranti yang berbeda dapat saling mengenali satu sama lain dan memulai fungsi-fungsi tingkat tinggi, dimana fungsi yang demikian dapat diimplementasikan dengan memanfaatkan *application spesific software*.

Bluetooth™ disusun dalam *microchip* yang di dalamnya memuat *baseband controller, flash memory,* dan modul *RF,* di

dalam *flash memory* terdapat *software* kontrol dan *identity coding.*

Spesifikasi dari *Bluetooth*™ antara lain:

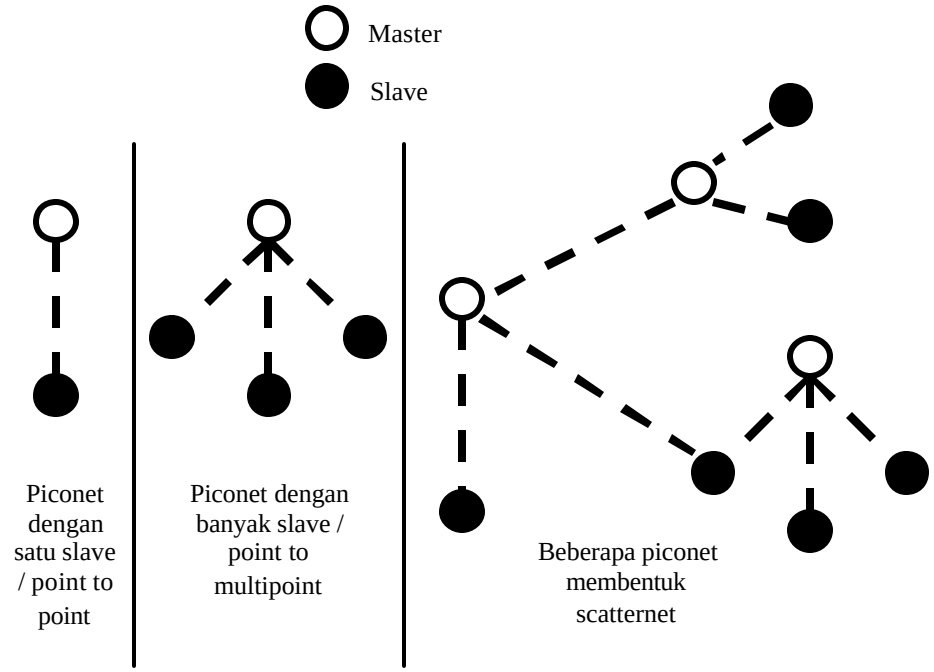
a. Frekuensi tinggi (2.402 – 2.480 GHz ISM)
b. *Frequency Hopping Spread Spectrum (FHSS)*
c. Modulasi *Gaussian Frequency Shift Keying (GFSK)*
d. Laju data 1 Mbit/s
e. Jarak jangkauan hingga 10 meter
f. Daya rendah (1 mW)
g. Penggunaan *chip* tunggal

A.2. Topologi jaringan *Bluetooth*™

Teknologi *Bluetooth*™ memiliki dua macam koneksi, yaitu koneksi *point to point* yang hanya terdiri dari dua unit *Bluetooth*™ dan koneksi *point to multipoint* dimana kanal komunikasi digunakan oleh beberapa unit *Bluetooth*™ bersama-sama.

Dua atau lebih unit *Bluetooth*™ yang menggunakan kanal bersama-sama membentuk *piconet* (dapat dianalogikan dengan sel pada teknologi telepon seluler dan *Wireless LAN*), salah satu unit *Bluetooth*™ yang memulai koneksi berlaku sebagai *master* dari *piconet* dan unit lainnya sebagai *slave*, hingga tujuh unit *slave* dapat aktif bersamaan dalam sebuah *piconet*, namun lebih banyak *slave* dapat tetap terdaftar pada *master* dalam kondisi yang disebut dengan *parked state*, unit-unit yang dalam keadaan *parked* tidak boleh aktif di dalam kanal namun mereka tetap tersinkronisasi dengan *master* baik waktu maupun pola lompatan *FHSS*-nya. Dengan demikian unit *master* tetap mengontrol akses terhadap kanal dan parameter-parameternya baik untuk unit *slave* yang aktif maupun yang dalam kondisi *parked*. *Master* mengatur lalu lintas kanal dengan metoda *polling*.

Beberapa *piconet* yang memiliki area jangkauan saling bertumpukan/*overlap* membentuk *scatternet*, meskipun setiap *piconet* hanya boleh memiliki sebuah *master*, namun *slave* dapat terdaftar dan aktif pada beberapa *piconet* dengan pengaturan *Time Division Multiplex*. Sebuah *master* pada sebuah *piconet* dapat menjadi *slave* pada *piconet* yang lainnya seperti tampak pada Gambar A.2. Masing-masing *piconet* yang *overlapping* tidak boleh tersinkronisasi frekuensinya, jadi setiap *piconet* harus memiliki pola loncatan frekuensi *FHSS*-nya sendiri-sendiri.



Gambar A.2. Topologi jaringan *Bluetooth*™

A.3. *Bluetooth*™ *Physical Links*

Kanal komunikasi *Bluetooth*™ dibagi menjadi slot-slot waktu dimana setiap slot waktu setara dengan sebuah lompatan frekuensi *FHSS* yang lamanya 625 µs, sehingga banyaknya lompatan adalah 1 / 0.000625 = 1600 lompatan per detik.

Antara *master* dan *slave* terdapat dua macam hubungan/*link* yang dapat dilakukan, yaitu:

1. *Synchronous Connection Oriented (SCO) link.*

   SCO link adalah hubungan *point to point* antara sebuah *master* dan sebuah *slave* dalam sebuah *piconet*. *Master* mempertahankan hubungan *SCO* dengan menggunakan slot-slot waktu yang telah dipesan pada interval tetap, sehingga dapat digolongkan sebagai hubungan *circuit switched*. Hubungan ini biasanya digunakan untuk informasi yang kritis terhadap waktu seperti suara. *Master* dapat membuat tiga buah hubungan *SCO* ke sebuah *slave* atau ke *slave* yang berlainan. Sedangkan *slave* dapat memiliki tiga buah hubungan *SCO* ke sebuah *master* atau dua buah hubungan *SCO* jika *SCO* awal berasal dari *master* yang berbeda. Paket-paket data *SCO* tidak pernah ditransmisikan ulang.

2. *Asynchronous Connection Less (ACL) link*

   ACL link adalah hubungan *point to multipoint* antara *master* dan semua *slave* yang ada dalam *piconet*. *Master* dapat membuat hubungan *ACL* ke setiap *slave* menggunakan slot yang tidak digunakan oleh *SCO link*, termasuk juga ke *slave* yang tengah menggunakan *SCO link*. Hubungan *ACL* bersifat *packet switched*, dan mendukung layanan *asynchronous* serta *isochronous*. Antara sebuah *master* dan sebuah *slave* hanya boleh ada sebuah hubungan *ACL*. Paket-paket data *ACL* dapat ditransmisikan ulang untuk menjamin integritas data. Paket data *ACL* yang tidak dialamatkan ke *slave* tertentu dianggap sebagai paket *broadcast* dan diterima oleh semua *slave*.

A.4. Protokol-protokol *Bluetooth*™

Tumpukan protokol *Bluetooth*™ dibagi menjadi empat lapisan utama, yaitu:

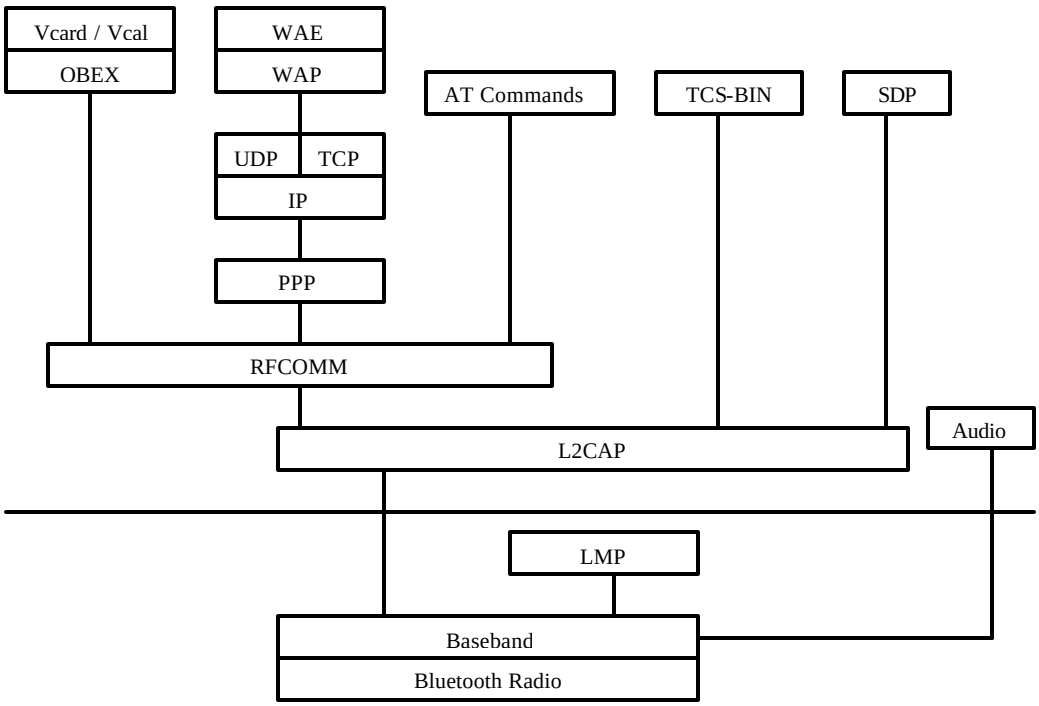
Tabel A.1 Protokol-protokol *Bluetooth*™

Lapisan protokol	Protokol-protokol dalam tumpukan
Protokol inti *Bluetooth*™	*Baseband, LMP, L2CAP, SDP*
Protokol pengganti kabel	*RFCOMM*
TCS (Telephony Control protocol Specification)	*TCS* Biner*, AT-Commands*
Protokol adopsi	*PPP, UDP/TCP/IP, OBEX, WAP, Vcard, Vcal, IrMC, WAE*

Protokol inti *Bluetooth*™ merupakan protokol spesifik *Bluetooth*™ yang dikembangkan oleh *Bluetooth*™ *SIG*. Protokol dan radio *Bluetooth*™ mutlak diperlukan dalam peranti-peranti berteknologi ini, sedangkan protokol lainnya digunakan sejauh bila diperlukan.

Lapisan pengganti kabel, lapisan kontrol telepon, dan protokol adopsi membentuk protokol yang berorientasi pada aplikasi untuk memungkinkan aplikasi bekerja di atas protokol inti *Bluetooth*™*.*

Spesifikasi *Bluetooth*™ bersifat terbuka sehingga protokol-protokol tambahan seperti *HTTP, FTP,* dll dapat diimplementasikan di dalamnya agar *interoperable* di atas protokol transport spesifik *Bluetooth*™, atau di atas *aplication oriented protocol.* Tumpukan berbagai protokol *Bluetooth*™ tampak pada Gambar A.3.



Gambar A.3. Tumpukan protokol *Bluetooth*™

A.4.1. Protokol inti *Bluetooth*™

a. *Baseband* (pita dasar)

Lapisan *baseband* dan *link control* memungkinkan hubungan *physical RF link* antara unit-unit *Bluetooth*™ sehingga membentuk *piconet*. Karena sistem *RF Bluetooth*™ merupakan sistem *FHSS* dimana paket-paket data ditransmisikan dalam slot waktu tertentu pada frekuensi tertentu, maka lapisan tersebut menggunakan prosedur *inquiry* (pemeriksaan) dan *paging* (prosedur untuk membentuk *physical link* bertipe *ACL* pada aras *baseband* yang terdiri dari sebuah aksi panggilan dari *initiator* dan sebuah aksi dari peranti dalam menanggapi panggilan itu) guna mensinkronisasikan transmisi frekuensi lompatan dan *clock* peranti-peranti *Bluetooth*™. Lapisan tersebut menyediakan 2 macam *physical link* dengan paket-paket *baseband* yang sesuai yaitu *SCO* dan *ACL* yang dapat ditransmisikan dengan cara *multiplex*

pada *RF link* yang sama. Paket *ACL* digunakan hanya untuk data, sedangkan paket *SCO* dapat berupa *audio* saja atau kombinasi dari *audio* dan data. Semua paket *audio* dan data dapat tersedia dengan aras *Forward Error Correction (FEC)* atau *Error Correction Cyclic Redundancy Check (CRC)* yang berbeda, serta dapat dienkripsi lebih jauh lagi, tipe-tipe data yang berbeda termasuk pesan-pesan *link* manajemen dan kontrol masing-masing dialokasikan sebuah kanal khusus.

Data *audio* dapat ditransfer antara dua atau lebih peranti *Bluetooth*™, sehingga memungkinkan berbagai model penggunaan dan data *audio* dalam paket *SCO* dikirimkan langsung dari atau menuju *baseband* dan tidak melalui *L2CAP.* Model *audio* cukup sederhana dalam *Bluetooth*™, dua buah peranti *Bluetooth*™ apapun dapat saling mengirimkan dan menerima data *audio* hanya dengan membuka sebuah *audio link*.

b. *LMP (Link Manager Protocol)*

LMP bertanggung jawab terhadap *link set-up* antara peranti-peranti *Bluetooth*™ dan mencakup aspek-aspek keamanan dan enkripsi dengan cara membentuk, menukar, dan memeriksa *link* dan kunci-kunci enkripsi serta kontrol dan negosiasi ukuran paket *baseband.* Selain itu juga mengendalikan *mode* daya dan *duty cycle* dari peranti radio *Bluetooth*™*,* juga status koneksi sebuah unit *Bluetooth*™ dalam sebuah *piconet.*

c. *L2CAP (Logical Link Control and Adaptation)*

L2CAP Bluetooth™ menyesuaikan protokol-protokol lapisan atas *baseband. L2CAP* dapat bekerja sejajar dengan *LMP,* hanya bedanya *L2CAP* menyediakan layanan-layanan bagi lapisan atas ketika *payload* data tidak pernah dikirimkan pada pesan-pesan *LMP. L2CAP* menyediakan layanan data

connection oriented dan *connectionless* kepada protokol-protokol lapisan atas dengan kemampuan protokol *multiplexing*, pembagian dan penggabungan data, serta *group abstractions*. *L2CAP* membolehkan protokol-protokol pada aras yang lebih tinggi dan aplikasi–aplikasi untuk mengirim dan menerima paket data *L2CAP* sampai sepanjang *64 Kilobytes*. Meskipun protokol *baseband* menyediakan tipe *link SCO* dan *ACL, L2CAP* ditentukan hanya untuk hubungan *ACL*.

d. *SDP (Service Discovery Protocol)*

Service Discovery (SD) merupakan bagian yang sangat penting dalam kerangka *Bluetooth™. SD* adalah protokol untuk mengetahui kemampuan dari peranti-peranti yang saling berhubungan atau peranti-peranti induk.

Dengan kata lain *SD* merupakan prosedur untuk menanyakan dan melihat berbagai layanan yang ditawarkan oleh atau melalui peranti *Bluetooth™* lain. Layanan *service discovery* ini menyediakan basis bagi semua model penggunaan. Dengan menggunakan *SDP*, informasi peranti, layanan-layanan dan karakteristik layanan dapat ditanyakan dan setelah itu dapat diciptakan koneksi antara dua atau lebih peranti *Bluetooth™.*

A.4.2. Protokol penggantian kabel (*RFCOMM*)

RFCOMM adalah protokol emulasi kabel/jalur serial *RS-232* yang sesuai dengan spesifikasi *European Telecommunications Standards Institute (ETSI).*

A.4.3. Protokol kontrol *telephony (TCS)*

a. *Telephony Control Binary*

Protokol ini berorientasi bit yang mengontrol pensinyalan panggilan wicara dan data antar peranti *Bluetooth*™, protokol ini juga mengatur manajemen mobilitas untuk mengatur sekumpulan peranti *TCS Bluetooth*™. Protokol ini direkomendasikan oleh *International Telecommunication union (ITU).*

b. *Telephony control AT Commands*

Bluetooth™ menspesifikasikan set perintah-perintah *AT* yang digunakan untuk mengendalikan telepon seluler dan *modem* didalamnya, bisa juga untuk layanan faksimili. Protokol ini juga direkomendasikan oleh *ITU* dan *ETSI.*

A.4.4. Protokol-protokol adopsi

a. *Point to Point Protocol (PPP)*

PPP dirancang pada *RFCOMM* agar dapat melakukan koneksi *point to point*. Protokol *PPP* berasal dari *Internet Engineering Task Force (IETF).*

b. *Transport Control Protocol (TCP)/User Datagram Protocol (UDP)/Internet Protocol(IP)*

Protokol-protokol ini distandarisasi oleh *IETF* dan digunakan untuk komunikasi internet. Dengan adanya protokol-protokol ini memungkinkan peranti *Bluetooth*™ untuk dapat mengakses internet, dan juga sebagai media transpor bagi *Wireless Application Protocol(WAP).*

c. *Protokol OBject EXchange (OBEX)*

IrOBEX adalah protokol yang dikembangkan *oleh Infrared Data Association (IrDA),* protokol ini dapat mentransfer sebuah *object* menggunakan operasi *Put* dan *Get* yang dilakukan dengan menggunakan model *client* dan *server*. Protokol ini dapat menggunakan media transpor *RFCOMM*

maupun *TCP/IP*, sedangkan aplikasi yang menggunakan protokol ini dapat berupa sinkronisasi, transfer *file*, dan *push object.*

d. *Wireless Application Protocol (WAP)*

WAP adalah protokol untuk berkomunikasi melalui internet antara *web server* dan sebuah telepon seluler. Tujuannya adalah untuk membawa isi internet dan layanan teleponi ke dalam telepon seluler digital dan terminal nirkabel lainnya.

A.5. Profil penggunaan *Bluetooth*™

Dari protokol-protokol yang telah disebutkan diatas kemudian akan terbentuk model penggunaan *Bluetooth*™ yang disebut dengan profil, yaitu pemilihan format pesan dan protokol yang digunakan untuk antarmuka layanan dan penggunaan tertentu.

Pada spesifikasi *Bluetooth*™ versi 1.1 didefinisikan tiga belas profil:

1. Profil *Generic Access*

   Profil ini mendeskripsikan bagaimana peranti-peranti *Bluetooth*™ berjalan (baik dalam keadaan *standby* atau aktif), untuk menjamin bahwa kanal komunikasi dan hubungan antar peranti selalu terjaga dengan baik, sehingga operasi multi profil dapat dilakukan. Profil ini difokuskan pada prosedur *discovery, link establishment* dan keamanan.

   Profil ini menggunakan protokol-protokol *baseband, LMP, L2CAP, SDP, RFCOMM, TCS* dan *OBEX.*

2. Profil *Service Discovery Application*

   Karena kemungkinan akan perkembangan layanan yang disediakan melalui *Bluetooth*™ akan meningkat dengan cepat dan tidak dapat diduga, maka dibutuhkan prosedur

untuk membantu pengguna *Bluetooth*™ mencari dan memilih layanan dari berbagai variasi yang tersedia. Profil ini dapat mencari layanan dengan tiga cara, berdasarkan kelas layanan, berdasarkan atribut layanan, dan dengan menjelajah.

Pencarian layanan sendiri dapat dilakukan dengan sepengetahuan pengguna pada peranti yang telah terhubung, dan tanpa sepengetahuan pengguna dengan menghubungi peranti *Bluetooth*™ yang ditemukan disekitarnya.

Untuk bisa menggunakan profil ini peranti-peranti *Bluetooth*™ terlebih dahulu perlu ditemukan, dihubungkan untuk kemudian diketahui layanan apa yang didukungnya.

Profil ini menggunakan protokol-protokol *baseband, LM, L2CAP, SDP,* dan aplikasi *service discovery.*

3. Profil *Cordless Telephony*

Profil ini mendefinisikan protokol dan prosedur yang dibutuhkan untuk implementasi peralatan yang disebut dengan *"3-in-1 phone"* pada telepon seluler, dimana selain dapat menghubungi telepon lain melalui jaringan seluler juga dapat digunakan sebagai telepon tanpa kabel yang terhubung ke *base station* lokal/pribadi.

Profil ini menggunakan protokol-protokol *baseband, LMP, L2CAP, TCS binary, SDP, Call Control (CC),* dan aplikasi *telephony.*

4. Profil *Intercom*

Profil ini melengkapi fungsi dari *"3-in-1 phone"* diatas, dimana telepon seluler berfungsi sebagai *walkie talkie*, yang terhubung langsung satu sama lain.

Profil ini menggunakan protokol-protokol *baseband, LMP, L2CAP, TCS binary, SDP, CC,* dan aplikasi *telephony.*

5. Profil *Serial Port*

Profil ini mendefinisikan protokol dan prosedur yang dibutuhkan untuk menggunakan *Bluetooth*™ sebagai emulasi kabel serial *RS-232* atau komunikasi serial sejenisnya, yang akan tampak sebagai *virtual serial port.*

Profil ini menggunakan protokol-protokol *baseband, LMP, L2CAP, RFCOMM, SDP, Serial Port Emulation Application Programming Interface (API)*, dan aplikasi serial.

6. Profil *Headset*

Profil ini mendefinisikan protokol dan prosedur yang dibutuhkan untuk implementasi *headset* yang terhubung secara nirkabel dengan tujuan untuk input dan output audio yang menyediakan kanal audio *full duplex*, dengan *headset* ini akan meningkatkan mobilitas pengguna namun mempertahan privasi pengguna.

Profil ini menggunakan protokol-protokol *baseband, LMP, L2CAP, RFCOMM, SDP, kontrol headset*, dan aplikasi *headset.*

7. Profil *Dial-Up Networking*

Profil ini mendefinisikan protokol dan prosedur yang dibutuhkan untuk implementasi model *"Internet Bridge"*, dimana telepon seluler dan *modem* berfungsi sebagai *modem* nirkabel agar komputer dapat mengakses internet menggunakan layanan *dial-up*. Profil ini juga memungkinkan komputer untuk menerima panggilan data melalui telepon seluler dan *modem*.

Profil ini menggunakan protokol-protokol *baseband, LMP, L2CAP, RFCOMM, SDP, dialing dan kontrol*, serta aplikasi emulasi *modem*.

8. Profil Fax

Profil ini mendefinisikan protokol dan prosedur yang dibutuhkan peranti *Bluetooth*™ untuk dapat digunakan oleh komputer sebagai *modem* faksimili nirkabel untuk mengirim dan menerima pesan faksimili.

Profil ini menggunakan protokol-protokol *baseband, LMP, L2CAP, RFCOMM, SDP, dialing* dan kontrol, serta aplikasi emulasi *modem*.

9. Profil *LAN Access*

Profil ini mendefinisikan cara akses ke *LAN* dengan menggunakan *PPP* melalui *RFCOMM*, akses dapat dilakukan baik oleh peranti *Bluetooth*™ tunggal maupun jamak, juga dapat dilakukan langsung dari *PC* ke *PC* menggunakan emulasi kabel serial *PPP.*

Profil ini menggunakan protokol-protokol *baseband, LMP, L2CAP, RFCOMM, SDP, ME, PPP, IP, TCP & UDP,* aplikasi *LAN.*

10. Profil *Generic Object Exchange*

Profil ini mendefinisikan protokol dan prosedur yang dibutuhkan aplikasi peranti *Bluetooth*™ untuk dapat bertukar *object*, dalam hal ini *object* dapat berupa sinkronisasi, transfer *file*, atau *object push.*

Profil ini menggunakan protokol-protokol *baseband, LMP, L2CAP, RFCOMM, SDP, OBEX,* dan aplikasi *client server.*

11. Profil *Object Push*

Profil ini mendefinisikan protokol dan prosedur yang dibutuhkan aplikasi peranti *Bluetooth*™ untuk dapat menggunakan model *object push*, dicontohkan adalah

mengirim dan/atau meminta *object* yang berupa kartu nama elektronis atau jadwal ke *inbox* peranti *Bluetooth*™ lainnya.

Profil ini membutuhkan profil *Generic Object Exchange Profile (GOEP)* untuk dapat beroperasi.

Profil ini menggunakan protokol-protokol *baseband, LMP, L2CAP, RFCOMM, SDP, OBEX,* dan aplikasi *push client server.*

12. Profil *File Transfer*

Profil ini mendefinisikan protokol dan prosedur yang dibutuhkan agar peranti *Bluetooth*™ dapat bertukar *file*, dengan begitu sebuah peranti *Bluetooth*™ dapat menjelajah dan melihat *file* atau *object* yang ada pada peranti *Bluetooth*™ lainnya, kemudian dapat mengkopi dan memanipulasi (menghapus, membuat direktori baru, mengedit) *file* tersebut.

Profil ini membutuhkan profil *GOEP* untuk dapat beroperasi.

Profil ini menggunakan protokol-protokol *baseband, LMP, L2CAP, RFCOMM, SDP, OBEX*, dan aplikasi *file transfer client server.*

13. Profil *Synchronization*

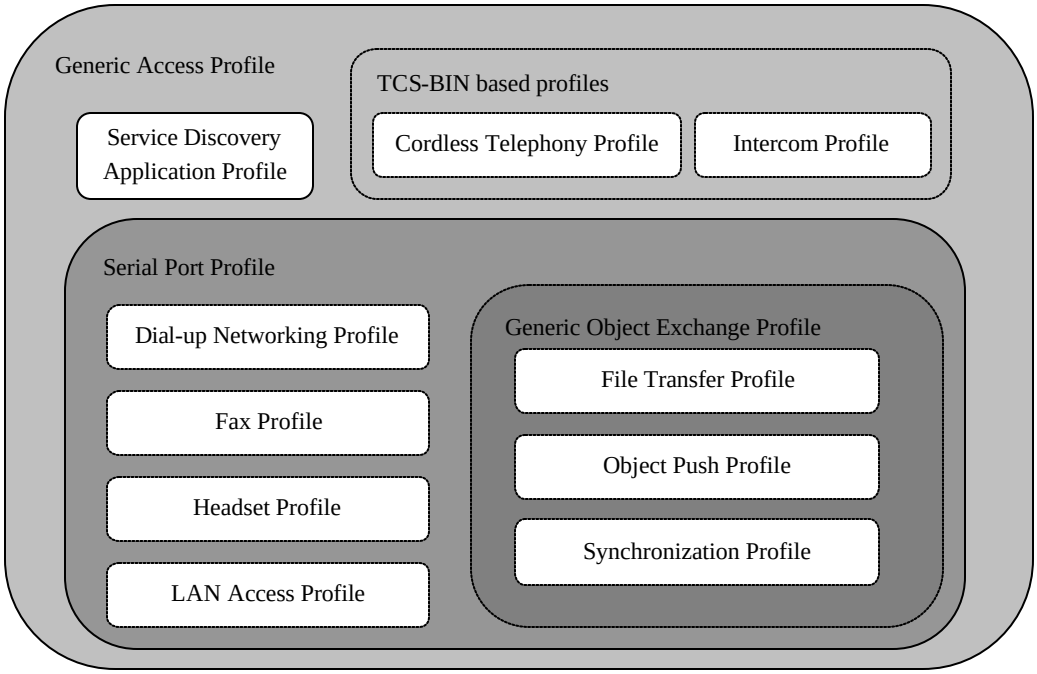
Profil ini mendefinisikan protokol dan prosedur yang dibutuhkan agar berbagai peranti *Bluetooth*™ dapat saling melakukan sinkronisasi data, profil ini juga membutuhkan *GOEP* untuk beroperasi.

Contoh penggunaannya yaitu antar komputer, *Personal Digital Assistant (PDA)*, dan telepon seluler dapat saling bertukar data *Personal Information Management (PIM)* serta catatan perubahannya agar semua informasi *PIM* sama persis disetiap peranti, data *PIM* dapat berupa buku telepon dan kalender.

Profil ini menggunakan protokol-protokol *baseband, LMP, L2CAP, RFCOMM, SDP, OBEX,* dan aplikasi *Infrared Mobile Communications (IrMC) client server.*

A.5.1. Struktur ketergantungan profil-profil *Bluetooth*™

Dari penjelasan diatas tampak bahwa beberapa protokol akan membentuk sebuah profil, dan sebuah protokol dapat digunakan pada lebih dari satu profil, ini menyebabkan terjadinya ketergantungan profil yang satu dengan yang lainnya baik secara langsung maupun tidak langsung, pada Gambar A.4 tampak struktur ketergantungan profil satu dengan yang lainnya, sebuah profil memiliki ketergantungan pada profil yang ada diluarnya.



Gambar A.4. Struktur ketergantungan profil-profil *Bluetooth*™

A.6. Keamanan *Bluetooth*™

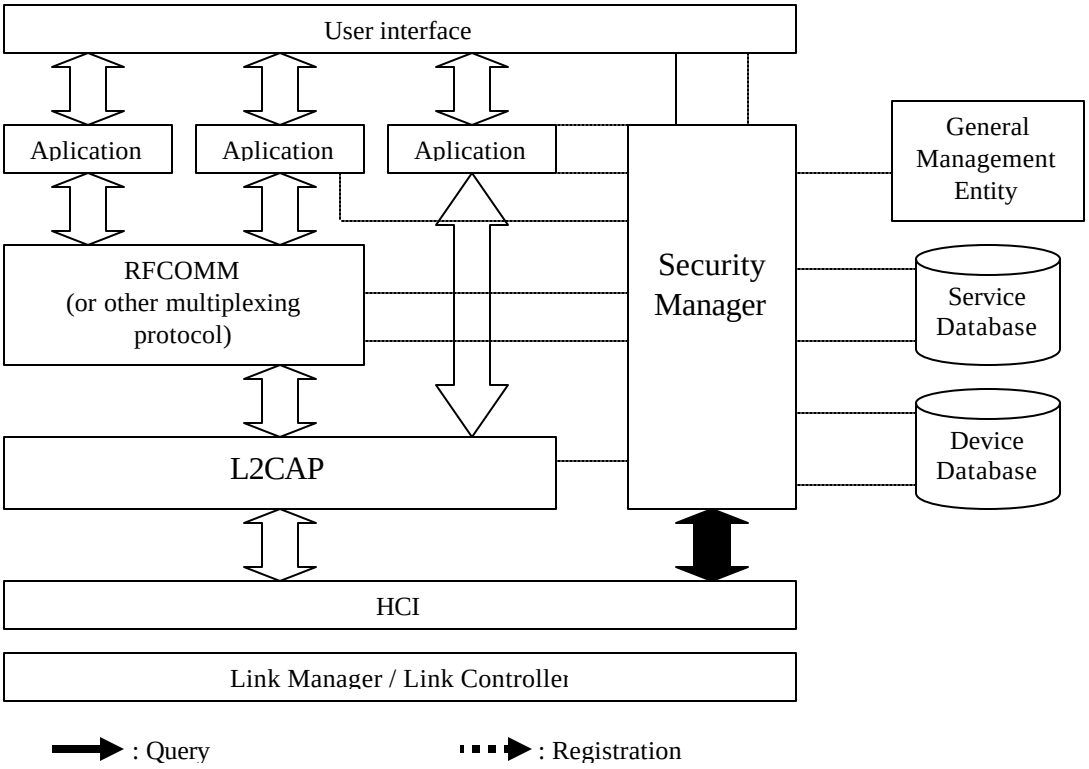
Bluetooth™ memiliki teknik keamanan pada *link level* untuk otentikasi peranti dan/atau pengguna serta enkripsi data.

Bluetooth™ memiliki tiga jenis *mode* keamanan antar peranti:

1. *Mode* 1 (*non-secure*), pada *mode* ini sebuah peranti tidak mengajukan prosedur keamanan apapun.
2. *Mode* 2 (*service-level enforced security*), sebuah peranti tidak mengajukan prosedur keamanan sebelum pemilihan kanal pada tingkat *L2CAP*. Mode ini memperbolehkan akses yang fleksibel untuk berbagai aplikasi, khususnya aplikasi-aplikasi yang bekerja dengan persyaratan keamanan yang berbeda.
3. *Mode* 3 (*link-level enforced security*), sebuah peranti mengajukan prosedur keamanan sebelum hubungan pada tingkat *LMP* terpenuhi.

Arsitektur umum keamanan *Bluetooth*™ tampak pada Gambar A.5 dimana komponen utama adalah *security manager*.

Arsitektur keamanan ini sifatnya fleksibel, keamanan dapat dilakukan baik pada tingkat aplikasi atau *link level*, atau keduanya, juga kapan dibutuhkan tambahan keamanan menggunakan *Personal Identification Number (PIN*) pengguna diatur disini.



Gambar A.5. Arsitektur keamanan *Bluetooth*™

Security manager memiliki fungsi-fungsi:

1. Menyimpan informasi keamanan pada layanan.
2. Menyimpan informasi keamanan pada peranti.
3. Menjawab permintaan akses yang membutuhkan implementasi protokol atau aplikasi, apakah akses akan diterima atau ditolak.
4. Menjalankan otentikasi dan/atau enkripsi sebelum terhubung ke aplikasi.
5. Melaksanakan proses input dari sebuah *ESCE (External Security Control Entity)*, misalnya pengguna peranti lunak untuk membuat hubungan terpercaya.
6. Memulai *pairing* dan menanyakan *PIN* masuk pengguna, permintaan *PIN* masuk dapat juga dilakukan oleh sebuah aplikasi.

Security manager pada *Bluetooth*™ tersentralisasi, sehingga memungkinkan implementasi yang mudah dan pengaturan akses yang fleksibel.

A.6.1. *Pairing*

Pairing merupakan prosedur dimana hubungan dibuat antara dua peranti *Bluetooth*™ yang sebelumnya saling tidak mengenal, ini dilakukan dengan membuat *link key* saat pertamakali kedua peranti dipasangkan. *Pairing* ini sendiri menggunakan kunci tersendiri yang disebut dengan *initialization key*, kunci ini dibuat dengan memperhitungkan alamat *Bluetooth*™ kedua peranti, bilangan acak, dan kunci rahasia/*PIN* yang digunakan bersama-sama.

A.6.2. Otentikasi

Otentikasi merupakan proses pemeriksaan/pembuktian siapa/apa yang berada pada ujung hubungan komunikasi.

Otentikasi dilakukan dengan menggunakan kunci rahasia yang disebut dengan *link key* antara dua buah peranti, *link key* dibuat saat prosedur *pairing*, semua peranti yang telah dipasangkan akan memiliki *link key* yang sama. Terdapat dua macam *link key*, yaitu *unit keys* dan *combination keys.*

Peranti yang menggunakan *unit keys* menggunakan kunci rahasia yang sama untuk semua koneksinya, ini bertujuan untuk menghemat memori dan antarmuka. Selama prosedur *pairing*, *unit keys* dienkripsi lalu dikirim ke peranti lainnya.

Combination keys adalah kunci yang unik untuk setiap pasangan peranti, kunci kombinasi ini digunakan hanya untuk mengamankan komunikasi antar kedua peranti tersebut.

Jelas bahwa peranti yang menggunakan *unit keys* tidak seaman peranti yang menggunakan *combination keys*, ini disebabkan karena *unit keys* diketahui oleh semua peranti yang

telah di-*paired* sehingga mereka dapat ikut mendengarkan komunikasi.

Otentikasi dilakukan dengan prosedur *challenge response* yang menggunakan algoritma *E1* (*E1* merupakan modifikasi dari *block cipher Secure And Fast Encryption Routine (SAFER+)),* pertama-tama pengotentikasi mengirimkan *challenge* sepanjang 128 bit, peranti kemudian memperhitungkan algoritma E1 pada *challenge*, alamat *Bluetooth*™-nya yang sepanjang 48 bit, dan *link key*, kemudian 32 *Most Significant Bit (MSB)* dari hasilnya yang sepanjang 128 bit dikirim kembali ke pengotentikasi untuk diverifikasi, dengan demikian otentikasi telah berhasil. Sedangkan bit sisa perhitungan yang disebut dengan *Authentication Ciphering Offset (ACO)* digunakan sebagai kunci *ciphering* untuk mengenkripsi data.

Sebuah peranti diverifikasi sebagai “terpercaya” jika sebuah respon otentikasi positif diberikan dan *flag* terpercaya diset.

Otentikasi peranti “tidak terpercaya” dilakukan dengan cara yang sama seperti peranti “terpercaya”, hanya saja *flag* “terpercaya” peranti tersebut tidak diset pada *database* internal.

Otentikasi dapat dilakukan dua arah, *master* mengotentikasi *client* dan sebaliknya.

A.6.3. Otorisasi

Otorisasi merupakan proses penentuan apakah sebuah peranti boleh mendapatkan akses terhadap layanan tertentu. Karena itu terdapat apa yang disebut dengan konsep “keterpercayaan”, dimana peranti yang terpercaya (dalam hal ini bisa terotentikasi atau dianggap “terpercaya”) diperbolehkan untuk mengakses layanan. Sedangkan peranti yang tidak terpercaya atau peranti yang tidak dikenali membutuhkan

otorisasi dari pengguna untuk mengakses layanan yang ada, tetapi ini bukan sesuatu yang mutlak, karena otorisasi dapat diberikan oleh aplikasi secara otomatis. Otorisasi selalu mencakup otentikasi.

Terdapat tiga jenis tingkat keterpercayaan peranti:

1. Peranti terpercaya (*trusted device*), peranti ini sebelumnya telah diotentikasi, sebuah kunci *link* disimpan dan peranti diberi tanda “terpercaya” dalam *database* peranti.
2. Peranti tidak terpercaya (*untrusted device*), peranti ini sebelumnya telah diotentikasi, sebuah kunci *link* disimpan, namun peranti tidak diberi tanda “terpercaya” dalam *database* peranti.
3. Peranti tidak dikenal (*unknown device*), tidak terdapat informasi keamanan pada peranti dan termasuk dalam golongan peranti tidak terpercaya.

A.6.4. Registrasi

Security manager juga mempertahankan informasi keamanan untuk layanan-layanan yang ada dalam *database*, aplikasi harus mendaftar/registrasi kepada *security manager* sebelum dapat diakses.

Bila tidak dilakukan registrasi digunakan tingkat keamanan dasar, yaitu semua koneksi masuk perlu otorisasi dan otentikasi, sedangkan semua koneksi keluar perlu otentikasi.

A.6.5. Penggunaan kunci eksternal

Arsitektur keamanan *Bluetooth*™ tidak melarang penggunaan kunci eksternal, aplikasi dapat dibuat untuk langsung membagikan kunci atau *PIN* secara langsung (namun tidak dimungkinkan untuk menyediakan kunci enkripsi dari luar).

Dalam hal ini, pengguna harus memperhatikan *interoperability* antar piranti *Bluetooth*™ yang digunakannya.

A.6.6. Tingkat keamanan layanan

Tingkat keamanan layanan digolongkan menjadi tiga jenis:

1. Perlu otorisasi, akses hanya diberikan secara otomatis bagi piranti terpercaya atau peranti yang tidak terpercaya namun telah melalui prosedur otorisasi.
2. Perlu otentikasi, sebelum hubungan antar aplikasi dilakukan, peranti harus diotentikasi.
3. Perlu enkripsi, *mode* hubungan harus diubah menjadi *mode* terenkripsi sebelum akses terhadap layanan diberikan.

Informasi tingkat keamanan layanan tersebut disimpan dalam *database* layanan pada *security manager.*

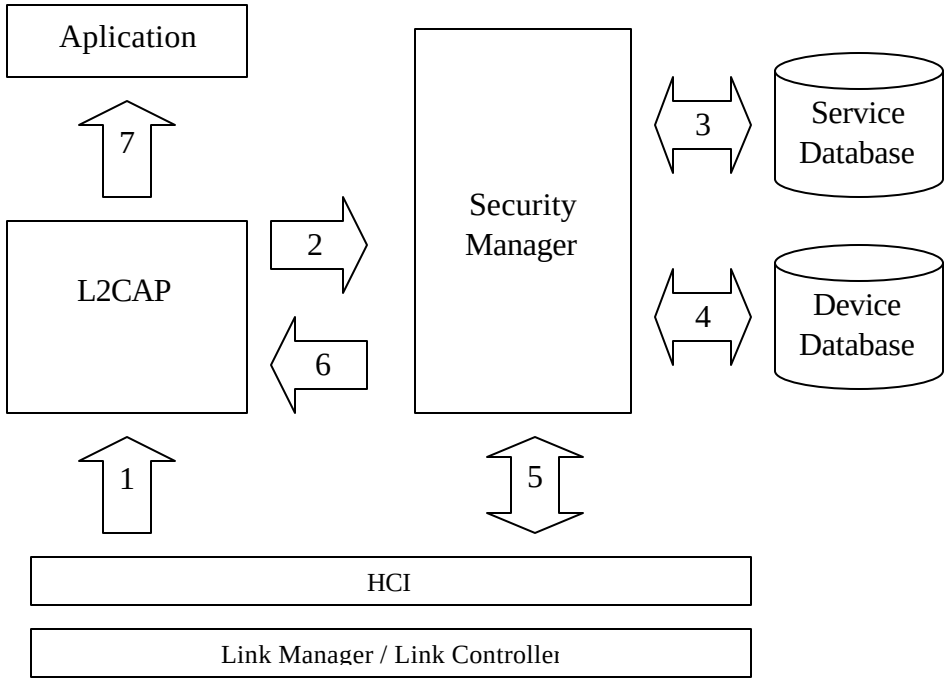
A.6.7. Enkripsi data

Untuk enkripsi data digunakan algoritma *stream cipher E0* yang didesain khusus untuk jaringan nirkabel *Bluetooth*™, sebuah kunci enkripsi dipakai untuk setiap sesi komunikasi yang juga digunakan untuk kunci rahasia setiap paket data.

E0 juga merupakan algoritma pengembangan kunci rahasia menjadi aliran *ciphertext*, ini dilakukan dengan memperhitungkan algoritma tersebut dengan angka acak, *link key*, dan sisa perhitungan algoritma *E1 (ACO).*

A.6.8. Prosedur umum pembuatan koneksi

Agar layanan dapat tersedia bagi peranti tanpa interferensi pengguna, harus dilakukan otentikasi setelah tingkat keamanan layanan ditentukan. Otentikasi dilakukan ketika ada permintaan koneksi dari layanan, Gambar A.6 menjelaskan tentang aliran informasi untuk akses piranti terpercaya.



Gambar A.6. Prosedur umum pembuatan koneksi

Prosedur-prosedur yang dilakukan:

1. Permintaan koneksi kepada *L2CAP*.
2. *L2CAP* meminta akses dari *security manager*.
3. *Security manager* mencari dalam *database* layanan.
4. *Security manager* mencari dalam *database* peranti.
5. Jika diperlukan, *security manager* menjalankan otentikasi dan enkripsi.
6. *Security manager* memberikan akses.
7. *L2CAP* membuat hubungan/koneksi.