

IMPLEMENTASI JADWAL DAN PENGUMUMAN KULIAH MELALUI BLUETOOTH DENGAN TEKNOLOGI J2ME

Setya adji W.¹, Agus Virgono²

^{1,2}Departement Teknik Elektro Institut Teknologi Telkom, Bandung
E-mail: Adji_ega24@plasa.com, agv@ittelkom.ac.id

ABSTRAK

Penggunaan teknologi Bluetooth yang digunakan pada sejumlah handset handphone memungkinkan dibuatnya aplikasi yang mudah dan tanpa biaya seperti aplikasi jadwal midlet ini yang merupakan layanan informasi yang bersifat indoor jadi layanan ini hanya dapat diakses oleh pelanggan pada area tertentu. Layanan ini di tujukan untuk mahasiswa dan dosen yang memiliki mobilitas yang tinggi dalam proses belajar mengajar. Pada jadwal midlet akan digunakan API JSR- 82 yang dibutuhkan dalam pengontrolan Bluetooth berdasarkan API JSR-82 ,sehingga pengiriman data berlangsung dengan baik, selain itu parameter seperti delay data end to end, banyaknya data yang dikirim, biaya pembuatan sistem akan menentukan kelayakan sistem Bluetooth sebagai teknologi alternatif untuk penyampaian informasi selain teknologi yang sudah ada seperti SMS,GPRS,WAP..

Kata Kunci: Bluetooth, J2ME, Midlet

1. PENDAHULUAN

1.1 Latar Belakang

Jadwal mempunyai kedudukan yang penting dalam proses belajar mengajar antara dosen dan mahasiswa, karena dari jadwal yang telah disepakati, mahasiswa dan dosen dapat melakukan kegiatan proses belajar mengajar dengan efektif. Perubahan jadwal secara tiba-tiba membuat proses belajar mengajar tidak dapat berlangsung efektif lagi, terutama bagi mahasiswa yang terlambat memperoleh informasi. Keterlambatan memperoleh informasi dapat diatasi dengan penyampaian informasi melalui teknologi yang ada seperti SMS,WAP,GPRS,Bluetooth.

Teknologi J2ME dapat menanamkan aplikasi yang kita buat ke dalam peralatan *mobile device*. MIDP2 merupakan fitur terbaru dari J2ME. MIDP2 terdapat API JSR- 82 yang merupakan library yang digunakan untuk mengakses Bluetooth, sehingga aplikasi yang dibuat sesuai dengan standar Bluetooth yang ada.

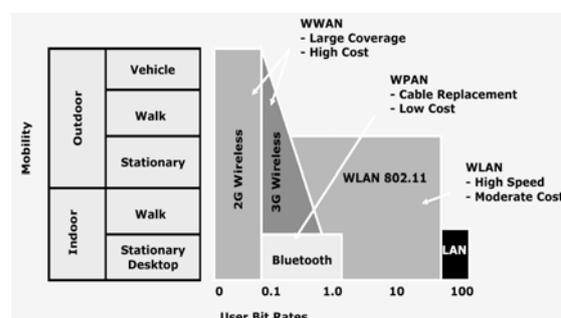
Jadwal *midlet* adalah aplikasi yang berjalan pada *mobile device* (*handphone*) untuk meminta *request* informasi jadwal kuliah kepada *server*, *server* akan memberikan *respon* tentang jadwal kuliah dan informasi tambahan pada waktu itu. Pengiriman data dari *klien* ke *server* menggunakan komunikasi Bluetooth, dengan memanfaatkan Bluetooth pada *mobile device*.

Penelitian ini akan mengembangkan dan menerapkan teknologi Bluetooth untuk layanan jadwal dan pengumuman kuliah dengan teknologi J2ME dan diterapkan pada *handphone*

1.2 Bluetooth

Bluetooth merupakan radio jarak pendek, menggunakan protokol, dan beroperasi di frekuensi 2,4GHz dari spektrum RF. Bluetooth dirancang

untuk menghubungkan perangkat yang berdaya rendah dengan jarak lebih dari 10meter dan mempunyai data transfer sebesar 1Mbps. Sebagai perbandingan dengan teknologi WiFi (802.11b) yang juga beroperasi di pita 2,4GHz, Bluetooth memiliki jarak yang lebih pendek (10m berbanding 100m), memiliki transfer data yang lebih rendah (1Mbps berbanding 10Mbps), tetapi mengkonsumsi daya yang lebih sedikit, membuat Bluetooth lebih cocok untuk perangkat yang memiliki daya rendah.



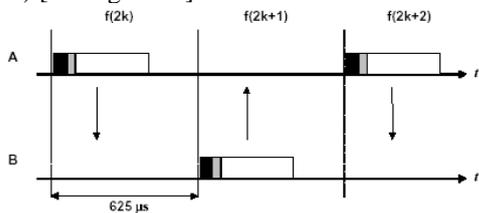
Gambar 1. Teknologi Bluetooth

Ketika sebuah perangkat Bluetooth akan menghubungi perangkat Bluetooth lain, maka ia harus memeriksa perangkat-perangkat Bluetooth yang ada di sekitarnya terlebih dahulu. Setelah menerima informasi tentang perangkat Bluetooth di sekitarnya, barulah dapat dilakukan koneksi menggunakan alamat dari perangkat-perangkat Bluetooth tersebut. Ketika dua buah perangkat Bluetooth saling terhubung, mereka membentuk suatu jaringan yang dinamakan *Piconet*.

1.3 Akses Jamak [Kleingsheim]

Bluetooth berdasarkan pada FH-CDMA. pada band 2.45GHz ISM, satu set dari 79 frekuensi *carrier hop* telah didefinisikan dengan spasi 1MHz.

Kanal *Bluetooth* adalah kanal *hopping* dengan waktu *hop* nominal 625µs. Komunikasi *full duplex* dicapai dengan menggunakan *time-dision duplex* (TDD). [Kleingsheim]

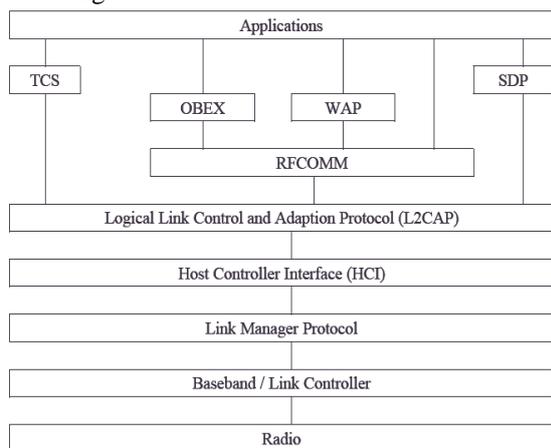


Gambar 2. Ilustrasi dari kanal FH / TDD yang diaplikasikan pada *Bluetooth* [Kleingsheim]

1.4 *Bluetooth* Protocol Stack

Bluetooth Protocol Stack adalah *Software* pendukung untuk mengimplementasikan *protocol Bluetooth* dan mengontrol *hardware Bluetooth* secara terprogram.

Layer RFCOMM mensimulasikan fungsi dari *port standard* komunikasi serial dan merupakan protokol pengganti kabel. *Service Discovery Protocol* (SDP) digunakan untuk menemukan servis *Bluetooth* yang ditawarkan oleh perangkat yang terhubung.



Gambar 3. *Bluetooth Protocol Stack*

2. PERANCANGAN

Aplikasi jadwal dan pengumuman kuliah akan diimplementasikan pada gedung yang mempunyai radius 35 meter, didalam gedung terdapat beberapa faktor yang dapat melemahkan sinyal *Bluetooth* sehingga mengurangi *coverage* seperti redaman gedung dan interferensi. Pada aplikasi ini *server* akan mengirimkan *respon* data setiap ada *request* dari *client*, ada kemungkinan beberapa *client* akan mengakses bersamaan. Agar kondisi diatas dapat terpenuhi maka perlu dipertimbangkan beberapa kekurangan *Bluetooth* seperti jarak jangkauan yang terbatas, akses jamak dan keamanan data.

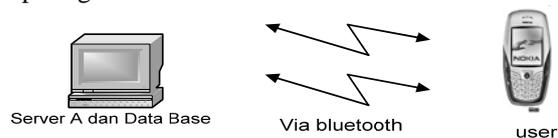
Jarak jangkauan yang terbatas dapat diatasi dengan memasang *server Bluetooth* pada setiap area tertentu.

Akses Jamak tiap *user* dapat diatasi dengan teknik FH-TDD sehingga beberapa pengguna dapat menggunakan aplikasi ini dengan membagi kanal dalam *sequence frekuensi hopping* yang berbeda, untuk *uplink* dan *downlink* menggunakan teknik TDD dimana *uplink* dan *downlink* dalam waktu yang berbeda

Keamanan data dilakukan dengan pemakaian *password* untuk *login update* jadwal. Teknologi *Bluetooth* susah untuk di *jamming* karena menggunakan teknologi frekuensi *hopping* jadi data yang dikirim dari pengirim ke penerima tidak dapat dicuri orang lain.

Pada spesifikasi *Bluetooth* diijinkan *server* diakses secara bersamaan maksimal tujuh *client* [Rustandi], jadi perancangan *server* akan diutamakan ke aplikasi program agar *server* dapat mengolah data dan memberikan *respon* sesuai yang diinginkan, meskipun diakses satu *client* maupun beberapa *client*. Pada keamanan datanya diperlukan *password* untuk aplikasi *login update*, maka pada *server* juga diperlukan perancangan *database* untuk menyimpan *password*. Pada perancangan *client* akan diutamakan pada koneksi dan menampilkan hasil *respon* pada layar *handphone*

Perancangan blok sistem yang akan di buat seperti gambar 4 dibawah ini



Gambar 4. Blok sistem

Prinsip kerja dari diagram blok diatas dapat diterangkan sebagai berikut, *Server Bluetooth* memberikan nama layanan yang akan dilayani pada *service record*, kemudian *server* menunggu adanya permintaan dari *client*.

Client mencari perangkat *Bluetooth* yang dapat dijangkau dan terdeteksi, setiap perangkat *Bluetooth* yang terdeteksi akan ditampilkan ke layar *handphone*, perangkat *Bluetooth* yang dipilih akan diperiksa layanannya, jika layanan sesuai maka data dikirim ke *server Bluetooth*. Pada saat itu *server Bluetooth* akan bekerja pada mode *threading* dimana *server* membagi dua tugas secara bersamaan

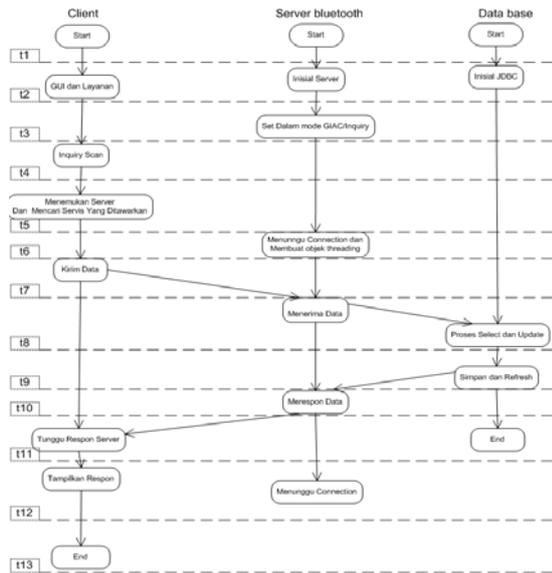
Menunggu dan membuka hubungan setiap ada *client* baru yang melakukan koneksi. Setiap ada *client* baru maka akan dibuatkan objek *threading*.

Menerima data yang dikirim *client* kemudian diolah datanya dan disimpan ke *database*, kemudian memberi *respon* ke *client*.

Dengan mode *threading* yang membagi dua tugas yang berbeda dan diproses bersamaan membuat koneksi dengan *client* dapat berjalan lancar. Dengan *threading* maka *server* dapat memberikan *respon* meski diakses bersama oleh beberapa *client* karena setiap ada koneksi akan

dibuat objek *threading* untuk memproses datanya. Hal ini akan lebih efisien dari pada menggunakan mode urut.

Secara garis besar proses sistem dapat diterangkan dari gambar UML proses dibawah ini.



Gambar 5. UML proses sistem

2.1 Perancangan Server

Tahap-tahap dalam perancangan *server Bluetooth*:

- Menentukan perangkat lunak yang dibutuhkan sistem:
 - Sistem Operasi Windows XP dengan SP2.
 - Netbeans 4.0 dan J2SDK 5.0, JCreator.
 - BlueCove.
- Menentukan Perangkat keras yang dibutuhkan system:
 - Processor Pentium 4.
 - Memory 256 MB DDR.
 - Port USB.
 - Bluetooth adapter Billionton 10m/100m

Hubungan antara perangkat keras dengan perangkat lunak seperti tabel dibawah ini.

Tabel 1. Hubungan perangkat lunak dan keras pada sistem

No	PERANGKAT LUNAK	PERANGKAT KERAS
1	Windows XP-SP2	Processor Pentium 4 256 MB DDR
2	Bluecove	Adapter Bluetooth Billionton 100m Port USB
3	NETBEANS 4.0, J2SDK5 dan Jcreator	Processor Pentium 4 256 MB DDR

- Menentukan aplikasi di *server* dan flowchart *server*

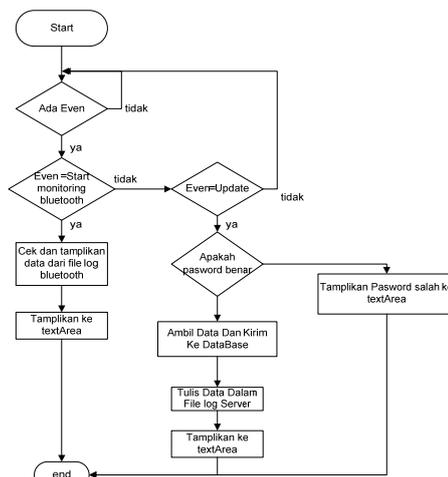
Aplikasi pada *server* terdiri dari dua macam yaitu aplikasi konsol dan aplikasi GUI. dengan membagi ke dalam dua bagian dapat mengurangi kerumitan alur program, meskipun demikian kedua aplikasi ini berhubungan dalam pertukaran data.

Aplikasi GUI berfungsi sebagai tampilan, agar pemakai dapat dengan mudah memakai aplikasi ini. Pada aplikasi GUI menampilkan dan melayani menu *update* jadwal. Menu *update* jadwal yang akan dirancang terdiri dari:

Monitoring Bluetooth dibuat dari komponen *text area* berfungsi untuk menampilkan keterangan terhadap *client* yang memakai *update* jadwal melalui *Bluetooth*. Keterangan pada layar berupa jadwal yang *ter-update* dan keterangan waktu terjadi proses *update* jadwal.

Menu *input update* jadwal seperti nim, password kuliah, kelas, ruang, waktu, hari.

Aplikasi GUI juga membuat *file log* yang disimpan pada *C:\Documents and Settings\adji\My Documents\percobaan\log TA* untuk mencatat semua proses *update* yang terjadi pada *server*, sehingga dapat ditelusuri jika terjadi kesalahan pada proses *update* jadwal. Aplikasi GUI dibuat dengan bahasa *java* dan menggunakan kompilator Netbeans 4.0. Pemakaian Netbeans lebih mudah dalam menggunakan OOP untuk pembuatan GUI. Pada Netbeans penambahan dan pemakaian komponen yang diinginkan lebih mudah karena sudah tersedia dalam bentuk *click and drag* (pallette). Proses sistem aplikasi GUI dapat dijelaskan dengan gambar flowchart proses dibawah ini

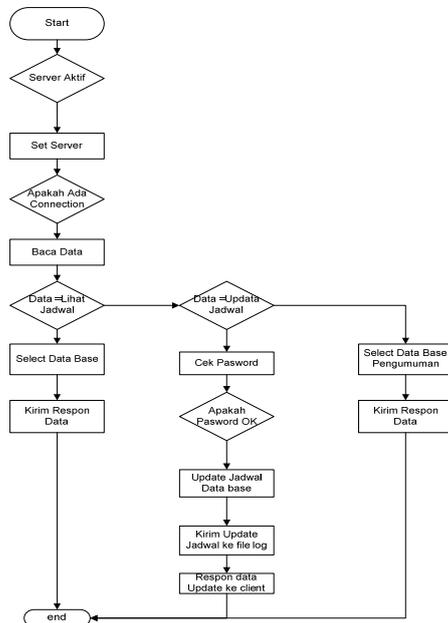


Gambar 6. Flowchart GUI server

Aplikasi *konsol* berfungsi untuk *inisialisasi stack*, *register service*, menunggu koneksi *client* dan melakukan *service* menjawab permintaan *client*. Aplikasi *konsol* juga melakukan *monitoring* pemakaian *Bluetooth* dan menampilkan setiap proses pengiriman dan menerima data pada layar konsol, dengan adanya layar *konsol* pembuat

program dapat melakukan pengamatan sehingga dapat mengetahui jika ada kesalahan .

Sistem kerja aplikasi *konsol* dapat dijelaskan dengan gambar *flowchart* proses dibawah ini.



Gambar 7.Flowchart konsol server

Aplikasi *konsol* dan GUI terinteraksi dalam satu *database* dengan nama myta. Pada *database* myta terdiri dari tabel *jadwal*, *password* dan PRS ketiga tabel tersebut berfungsi sebagai penyimpanan dan pencarian data yang dibutuhkan. Perincian tiap tabel dapat dijelaskan dibawah ini:

- Tabel prs terdiri dari kolom NIM dan ID.Tabel prs berfungsi mengambil ID dari NIM untuk dikirim dan dicari datanya di tabel mahasiswa.
- Tabel mahasiswa terdiri kolom MK(mata kuliah) dan KELAS.Data dari tabel akan digunakan untuk meminta data yang diperlukan pada tabel *jadwal*.
- Tabel *jadwal* terdiri dari kolom MK (mata kuliah),KELAS, WAKTU, HARI, RUANG.berguna untuk proses pencarian *jadwal* kuliah mahasiswa.
- Tabel *password* terdiri dari kolom NIP dan PASS.kedua kolom ini berfungsi untuk pengamanan agar *database* *jadwal* hanya bisa diubah oleh orang tertentu(admin/dosen).

2.2 Perancangan Client

Handphone merupakan Perangkat bergerak yang diperlukan dalam perancangan *client*. Pemakaian *handphone* disebabkan banyak mahasiswa yang sudah mempunyai *handphone*, sehingga cocok untuk penerapan aplikasi ini, syarat-syarat *handphone* yang bisa untuk aplikasi ini dapat dilihat tabel dibawah ini:

Tabel 2. Spesifikasi *handphone* yang digunakan pada *client*

No	SPEKIFIKASI HANDPHONE	KETERANGAN
1	Sistem Operasi	Minimal Simbian OS.V7
2	Teknologi Java	J2ME MIDP2 JSR-82
3	Fitur Perangkat Keras	Pemancar dan Penerima Bluetooth
4	Sisa Alokasi Memori	Minimal 500 KB

Dalam percobaan dan pengembangan program menggunakan beberapa emulator yang membantu dalam perancangan seperti

- J2ME Wireless Toolkit2.2
- Nokia Developer Tool
- Series 60 Developer Tool

Dengan menggunakan *emulator handphone* program yang dibuat dapat dicoba terlebih dahulu di *emulator*, hal ini untuk menghindari kerusakan yang terjadi pada saat mencoba aplikasi ke *handphone* yang sebenarnya.

Adapun jenis-jenis *handphone* yang sudah memiliki fitur diatas seperti: BenQP30,Sony Ericsson P900,Nokia 6600,Nokia 6620,Nokia 7700. Pada percobaan perancang memakai Nokia 6600 karena Nokia 6600 memiliki berbagai kelebihan dengan harga yang tidak terlalu mahal. Aplikasi pada *handphone* dibuat semudah mungkin agar dapat digunakan dan dimanfaatkan orang banyak. Aplikasi ini terdiri dari tiga menu:

- Menu lihat *jadwal*: Pada menu ini mahasiswa dapat melihat *jadwal* kuliah terbaru yang akan berlangsung selama 1 minggu. Mahasiswa dapat melihat *jadwal* dengan memasukkan NIMnya.
- Menu *update* *jadwal*: Pada menu ini dosen dapat mengubah *jadwal* kuliah yang ada dengan memasukkan NIP, *password*, kuliah, kelas, ruang, waktu, hari yang sesuai. Jika *password* dan NIM tidak sama dengan di *server* akan dikirim pesan kesalahan tapi jika benar akan dikirim sukses *update* .
- Menu pengumuman: Pada menu ini mahasiswa dapat melihat pengumuman yang disimpan dalam *database*.

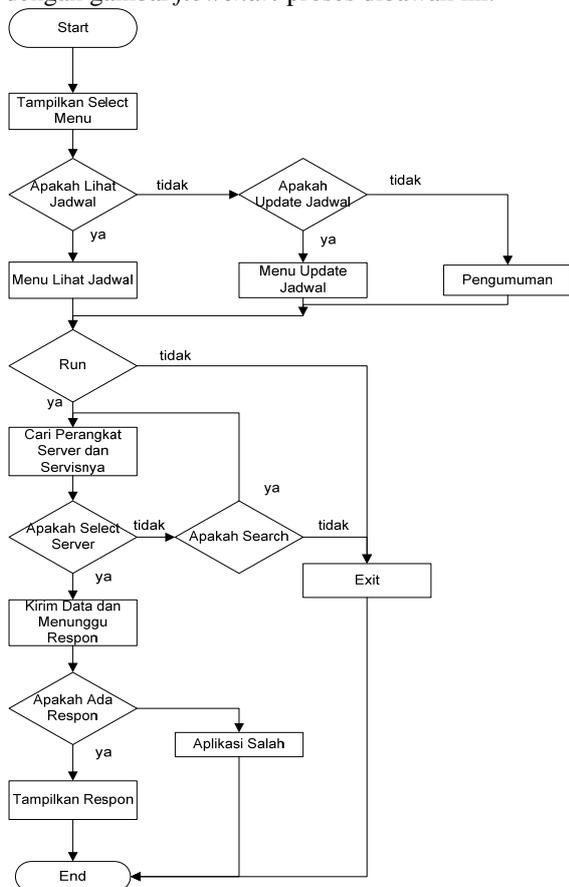
Ketiga menu tersebut melakukan hubungan ke *server* dengan menggunakan media Bluetooth pada fitur *handphone*. Program akan dibuat *threading* saat *handphone* melakukan koneksi ke *server*, pemakaian *threading* untuk menghindari kesalahan jika pengguna menggagalkan hubungan yang dapat membuat memori *handphone* menjadi penuh dan terbakar.

Beberapa langkah yang dilakukan saat koneksi *client-server* pada *handphone* adalah sebagai berikut

- Stack initialization* yaitu melakukan inisialisasi terhadap JSR-82 yang terdapat pada library J2ME MIDP2 .

- b. *Device discovery* yaitu mencari perangkat *Bluetooth* didaerah sekitarnya.
- c. *Service discovery* yaitu mencari service yang diinginkan.
- d. *Service* layanan yaitu melakukan pengiriman data dari *client* ke *server* dan menerima data dari *server*. Setiap menu mengirimkan index tertentu ke *server*, sehingga *server* dapat mengetahui apa yang diinginkan *client*. Untuk menu lihat jadwal akan dikirim *index* "pil1", menu *update* jadwal akan dikirim *index* "pil2", menu pengumuman akan dikirim *index* "pil3".

Proses sistem aplikasi J2ME dapat dijelaskan dengan gambar *flowchart* proses dibawah ini.



Gambar 8. Flowchart *client* pada J2ME

3. PENGUKURAN DAN ANALISA

Pengukuran dilakukan untuk melihat kelayakan perancangan dalam implementasinya di lapangan dengan melihat parameter *delay* dan *bitrate*.

3.1 Pengukuran Delay

Pengukuran meliputi *delay* pencarian *server* dan pengiriman data terhadap jarak, *delay* pengiriman data terhadap banyaknya data yang dikirim. Pengukuran untuk mengetahui pengaruh jarak dan banyaknya data yang dikirim terhadap jalannya aplikasi ini dan juga membuktikan bahwa *program* berjalan sesuai dengan perencanaan

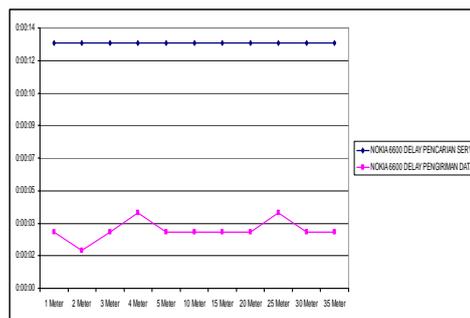
3.2 Delay pencarian *server* dan pengiriman data terhadap jarak

Kedua percobaan dibawah mengirimkan data *upload* 18 bytes dan data *download* 208 bytes. Percobaan ini diuji pada menu Lihat Jadwal.

Pada percobaan pertama, satu *client* melakukan koneksi ke *server*. Dari pengukuran didapat data seperti table 3. dibawah ini

Tabel 3. Satu *client* melakukan koneksi ke *server*

JARAK	DELAY NOKIA 6600	
	PENCARIAN SERVER(Ds)	PENGIRIMAN DATA(Dd)
1 Meter	0:00:13	0:00:03
2 Meter	0:00:13	0:00:02
3 Meter	0:00:13	0:00:03
4 Meter	0:00:13	0:00:04
5 Meter	0:00:13	0:00:03
10 Meter	0:00:13	0:00:03
15 Meter	0:00:13	0:00:03
20 Meter	0:00:13	0:00:03
25 Meter	0:00:13	0:00:04
30 Meter	0:00:13	0:00:03
35 Meter	0:00:13	0:00:03



Gambar 9. Grafik delay *Bluetooth* terhadap jarak berdasar tabel 1

Tabel 4. Dua *client* melakukan koneksi ke *server*

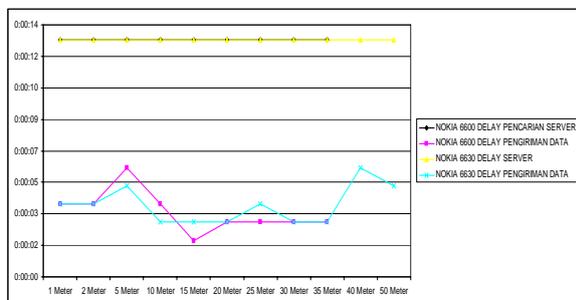
	DELAY NOKIA 6600		DELAY NOKIA 6630	
	(Ds)	(Dd)	(Ds)	(Dd)
1 m	0:00:13	0:00:04	0:00:13	0:00:04
2 m	0:00:13	0:00:04	0:00:13	0:00:04
5 m	0:00:13	0:00:06	0:00:13	0:00:05
10 m	0:00:13	0:00:04	0:00:13	0:00:03
15 m	0:00:13	0:00:02	0:00:13	0:00:03
20 m	0:00:13	0:00:03	0:00:13	0:00:03
25 m	0:00:13	0:00:03	0:00:13	0:00:04
30 m	0:00:13	0:00:03	0:00:13	0:00:03
35 m	0:00:13	0:00:03	0:00:13	0:00:03
40 m			0:00:13	0:00:06
50 m			0:00:13	0:00:05

Didapatkan rata-rata *delay* pencarian *server* = 13 detik dan rata-rata *delay* pengiriman data = 3.09 detik

Dari grafik gambar 9 perbedaan jarak antara *client-server* tidak mempengaruhi *delay* pencarian *server* dan *delay* pengiriman data, dikarenakan jangkauan *Bluetooth* yang kecil (35 meter)

sedangkan sinyal *Bluetooth* memiliki kecepatan 3.10^8 m/s. Jadi *delay* jarak akan kecil sekali jadi bisa diabaikan.

Pada percobaan kedua, dua *client* melakukan koneksi ke *server* secara bersamaan ini untuk menguji jika terjadi akses bersamaan. Dari pengukuran didapat data seperti table 4.



Gambar 10. Grafik *delay Bluetooth* terhadap jarak berdasar Tabel 2

Client 1 :

Rata-rata *delay* pencarian *server* = 13 detik

Rata-rata *delay* pengiriman data = 3.5 detik

Client 2:

Rata-rata *delay* pencarian *server* = 13 detik

Rata-rata *delay* pengiriman data = 3.9 detik

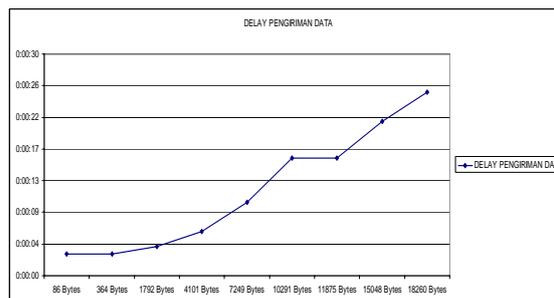
Dari grafik gambar 10 pengaksesan dua *client* tidak mengubah *delay* pencarian *server* tetapi saat pengiriman data terjadi perbedaan *delay* antara *client* 1 (rata-rata=3,5) dan *client* 2 (rata-rata=3,9).

3.3 Delay pengiriman data terhadap banyaknya data yang dikirim

Pada percobaan ini di uji pada menu Pengumuman, karena *database* pengumuman lebih mudah dalam pengaturannya.percobaan dilakukan pada jarak yang tetap.Pengukuran didapat data seperti table dibawah ini.

Tabel 5. *Delay* Pengiriman data vs banyak data

NO	DATA		DELAY PENGIRIMAN DATA(Dd)
	UP LOAD (Bytes)	DOWN LOAD (Bytes)	
1	5	81	0:00:03
2	5	359	0:00:03
3	5	1787	0:00:04
4	5	4096	0:00:06
5	5	7244	0:00:10
6	5	10286	0:00:16
7	5	11873	0:00:16
8	5	15043	0:00:21
9	5	18255	0:00:25



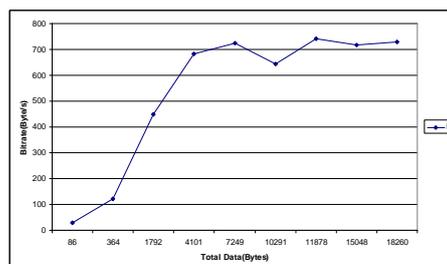
Gambar 11. Grafik *delay Bluetooth* terhadap banyaknya data yang dikirim berdasar Tabel 3

Dari grafik gambar 11 dapat disimpulkan bahwa semakin besar data semakin besar *delay* pengiriman datanya.

Bitrate adalah perbandingan antara jumlah data yang dikirim dengan waktu yang dibutuhkan, dari Tabel 3 hasil pengukuran diatas didapatkan tabel bitrate seperti dibawah ini.

Tabel 6. Bitrate terhadap banyaknya data yang dikirim

No	Total data(Bytes)	Bitrate(Bytes/s)
1	86	28.66
2	364	121.33
3	1792	448
4	4101	683.5
5	7249	724.9
6	10291	642.8
7	11878	742.3
8	15048	716.57
9	18260	730.4



Gambar 12. Grafik Bitrate terhadap banyaknya data berdasar Tabel 4

Dari grafik gambar 12 ada kecenderungan bit rate naik saat data yang dikirimkan bertambah banyak, saat data 86 Bytes memiliki bit rate 28,66 Bytes/s dan saat data 18260 memiliki bit rate 730 Bytes/s.

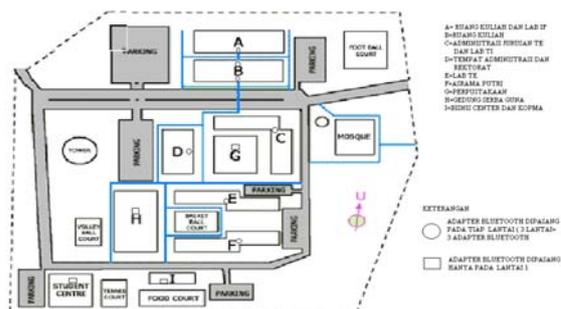
3.4 Penempatan Adapter *Bluetooth*

Layanan ini difokuskan tempat-tempat yang banyak dilakukan aktivitas mahasiswa.Supaya layanan ini dapat dirasakan maksimal maka diperlukan penempatan yang tepat posisi adapter *Bluetooth*, hal ini dilakukan untuk mengurangi daerah yang tidak mendapatkan sinyal(*blankspot*).

Berdasarkan pengukuran diatas didapatkan maksimal jarak yang tercakup *Bluetooth* 35 meter dengan user yang mengakses bersamaan maksimal 7.

Rancangan penempatan adapter *Bluetooth* seperti berikut:

- Pada gedung A setiap lantai diberi 1 *repeater Bluetooth*.
- Pada gedung B setiap lantai diberi 2 *repeater Bluetooth* karena pemakaian gedung ini untuk mahasiswa dirasakan lebih sering dari pada gedung yang lain, untuk *repeater* yang dekat gedung A akan melingkupi sebagian gedung A. Sedangkan *repeater* yang satunya akan mencoverage gedung B dari depan.
- Pada gedung C setiap lantai diberi 1 *repeater Bluetooth* dan ditempatkan pada dinding belok siku, agar sinyal dapat terjangkau dalam kondisi tanpa redaman.
- Pada gedung D setiap lantai diberi 1 *repeater Bluetooth* dan ditempatkan pada daerah yang paling belakang, dilakukan untuk mencoverage mahasiswa yang sedang melintasi jalan diantara gedung D dan gedung G.
- Pada gedung E setiap lantai diberi 1 *repeater Bluetooth* dan ditempatkan pada dinding belok siku, agar sinyal dapat terjangkau dalam kondisi tanpa redaman.
- Pada gedung F setiap lantai diberi 1 *repeater Bluetooth*, dan ditempatkan pada dinding belok siku, agar sinyal dapat terjangkau dalam kondisi tanpa redaman .
- Pada gedung G cukup diberi 1 *repeater* karena kondisi ruangan yang perpustakaan yang lossless.
- Pada gedung H cukup diberi 1 *repeater* karena kondisi ruangan yang perpustakaan yang lossless.
- Pada gedung I cukup diberi 1 *repeater* karena kondisi ruangan yang perpustakaan yang lossless. *Repeater* dipasang diposisi belakang, agar dapat mencoverage pengguna yang berada di kantin .



Gambar 13. Denah penempatan adapter *Bluetooth* di IT Telkom

4. KESIMPULAN

Berdasarkan hasil pembuatan dan analisa perangkat lunak serta perhitungan yang telah dilakukan maka dapat diambil beberapa kesimpulan sebagai berikut:

1. Aplikasi jadwal dan pengumuman kuliah dapat diterapkan dengan teknologi *Bluetooth*.
2. Pemakaian Teknologi *Bluetooth* pada aplikasi menu Lihat Jadwal membutuhkan *delay* rata rata pencarian *server* 13 detik dan *delay* pengiriman data 3,09 detik pada pengiriman data upload 18 bytes dan data download 208 bytes.
3. Ada kecenderungan bit rate naik saat data yang dikirimkan bertambah banyak, saat data 86 Bytes memiliki bit rate 28,66 Bytes/s dan saat data 18260 memiliki bit rate 730 Bytes/s

Pustaka

- Jode, Martin de. (2004). *Programing The Java 2 Micro Edition For Symbian OS*. John Wiley & Sons Ltd, England.
- Klingsheim, André N. (2004). *J2ME Bluetooth Programming*. Department of Informatics University of Bergen, Bergen.
- Ericson, Sony. (2004). *Developing Applications with the Java APIs for Bluetooth™(JSR-82)*. Developer Training Material.
- Rustandi, Andi. (2004). *Voice Transmission over Bluetooth™* Departemen Teknik Informatika Fakultas Teknologi Industri Institut Teknologi Bandung, Bandung.
- Wiharta, Setya Adji. (2004). *Desain Prototype SMS Melalui Teknologi WAP Dengan Bearer GPRS*. Sekolah Tinggi Teknologi Telkom, Bandung.