

UNIVERSITAS INDONESIA

MOBILE CITIZEN MEDIA

SKRIPSI

AJIB SETYO ARIFIN

0405030087

**FAKULTAS TEKNIK
PROGRAM SARJANA**

DEPOK

JUNI 2009



UNIVERSITAS INDONESIA

MOBILE CITIZEN MEDIA

SKRIPSI

**Diajukan sebagai salah satu syarat untuk memperoleh gelar
Sarjana Teknik**

AJIB SETYO ARIFIN

0405030087

**FAKULTAS TEKNIK
PROGRAM SARJANA**

DEPOK

JUNI 2009

HALAMAN PERNYATAAN ORISINALITAS

Skripsi ini adalah hasil karya saya sendiri, dan semua sumber baik yang dikutip maupun dirujuk telah saya nyatakan dengan benar.

Nama : AJIB SETYO ARIFIN

NPM : 0405030087

Tanda Tangan :

Tanggal : 17 Juni 2009

HALAMAN PENGESAHAN

Skripsi ini diajukan oleh :

Nama : AJIB SETYO ARIFIN

NPM : 0405030087

Program Studi : Teknik Elektro

Judul Seminar : *Mobile Citizen Media*

Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Teknik pada Program Studi Teknik Eektro Fakultas Teknik, Universitas Indonesia

DEWAN PENGUJI

Pembimbing : Prof. Dr. Ir. Dadang Gunawan, M.Eng (.....)

Penguji : Dr. Ir. Muhammad Asvial, M.Eng (.....)

Penguji : Dr-Ing. Ir. Kalamullah Ramli, M.Eng (.....)

Ditetapkan di : Depok

Tanggal :



KATA PENGANTAR

Puji syukur saya panjatkan kepada Allah SWT, Tuhan Yang Maha Esa, karena atas berkat dan rahmat-Nya, saya dapat menyelesaikan skripsi ini. Penulisan skripsi ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gelar Sarjana Teknik Jurusan Teknik Elektro pada Fakultas Teknik Universitas Indonesia. Saya menyadari bahwa, tanpa bantuan dan bimbingan dari berbagai pihak, dari masa perkuliahan sampai pada penyusunan skripsi ini, sangatlah sulit bagi saya untuk menyelesaikan skripsi ini. Oleh karena itu, saya mengucapkan terima kasih kepada:

1. **Prof. Dr. Ir. Dadang Gunawan, M.Eng**, selaku dosen pembimbing yang telah menyediakan waktu, tenaga, dan pikiran untuk mengarahkan saya dalam penyusunan skripsi ini;
2. orang tua dan adik-adik saya, Indra dan Lestari, yang telah memberikan bantuan dukungan material dan moral;
3. Aris Cahyono, Yunarto, dan teman sejati dimanapun engkau berada, terimakasih;
4. rekan-rekan asisten laboratorium telekomunikasi, saya ucapkan terimakasih;
5. rekan-rekan elektro angkatan 2005, tetap semangat.

Akhir kata, saya berharap Allah SWT, Tuhan Yang Maha Esa berkenan membalas segala kebaikan semua pihak yang telah membantu. Semoga skripsi ini membawa manfaat bagi pengembangan ilmu.

Depok, 17 Juni 2009

Penulis

HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI

SKRIPSI UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademik Universitas Indonesia, saya yang bertanda tangan di bawah ini:

Nama : AJIB SETYO ARIFIN

NPM : 0405030087

Program Studi : Teknik Elektro

Departemen : Teknik Elektro

Fakultas : Teknik

Jenis Karya : Skripsi

demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Indonesia **Hak Bebas Royalti Noneksklusif (Non-exclusive Royalty Free Right)** atas skripsi saya yang berjudul :

Mobile Citizen Media

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Universitas Indonesia berhak menyimpan, mengalihmedia/format kan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan mempublikasikan skripsi saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Jakarta

Pada tanggal : 17 Juni 2009

Yang menyatakan

AJIB SETYO ARIFIN

ABSTRAK

Nama : Ajib Setyo Arifin

Program Studi : Teknik Elektro

Judul : *Mobile Citizen Media*

Tugas akhir ini adalah membangun sebuah perangkat lunak *Mobile Citizen Media*, yaitu suatu media yang menjembatani masyarakat untuk saling bertukar informasi dengan menggunakan jaringan telepon seluler GSM dan UMTS.

Mobile Citizen Media adalah sebuah media yang memfasilitasi kegiatan *Citizen Journalism*. Perangkat lunak *Mobile Citizen Media* memiliki tiga komponen penting yaitu pengirim, pengolah, dan penyaji. Pengirim adalah bertindak sebagai *Citizen Journalism*, dibantu menggunakan sebuah perangkat lunak pada telepon seluler yang dibangun menggunakan pemrograman J2ME. Telepon seluler yang digunakan harus mendukung MIDP 2.0 dan CLDC 1.1. Pengolah dan penyaji adalah *website application* yang bertugas mengolah dan menampilkan informasi yang diterimanya dengan membuat suatu *database server* MySQL. Pada proses pengiriman data dari pengirim dan penerima dilakukan pengompresan data teks menggunakan teknik kompresi LZW.

Ujicoba pertama dilakukan dengan pengompresan data teks pada sisi *client*. Data teks akan dikompresi jika ukurannya lebih besar sama dengan 460 byte, dengan titik saturasi efisiensi sebesar 36% dicapai pada saat data berukuran lebih besar sama dengan 3027 byte. Ujicoba kedua dilakukan dengan memanfaatkan jaringan telepon seluler GSM dan UMTS. Pada jaringan GSM, diperlukan waktu pengunggahan selama 0.49 detik per Kbyte, sedangkan pada jaringan UMTS diperlukan waktu pengunggahan selama 0.28 detik per Kbyte.

Kata kunci:

Mobile, GSM, UMTS, Citizen Media, Journalism

ABSTRACT

Name : Ajib Setyo Arifin

Study Program : Electrical Engineering

Title : *Mobile Citizen Media*

This final task is to build a software Mobile Citizen Media, a media that is a bridge for people to exchange information with the mobile phone based on GSM and UMTS network.

Citizen Media Mobile is a system that will help facilitating activities Citizen Journalism. A software Mobile Citizen Media has three important components, called the sender, processing, and displayer. Sender is to act as Citizen Journalism assisted device using a mobile phone and a software is built by J2ME. Mobile phone has to support MIDP 2.0 and CLDC 1.1. Processing and displaying information is done by website combine with a MySQL database server. Before text data sent by sender, it will be compressed using LWZ compression technique.

The first trial conducted with data compression to the text on the client side. Text will be compressed if it's size is greater with 460 byte with saturation point of 36% efficiency achieved at the time of the data size is greater with the 3027 bytes. Testing is done with the second mobile phone network using the GSM and UMTS. In the GSM network, the time required for uploading is 0.49 seconds per Kbyte, while on the UMTS network, the time required for uploading is 0.28 seconds per Kbyte.

Keyword:

Mobile, GSM, UMTS, Citizen Media, Journalism

DAFTAR ISI

	Halaman
Judul.....	i
Halaman Pernyataan Orisinalitas.....	ii
Halaman Pengesahan.....	iii
Kata Pengantar.....	iv
Halaman Pernyataan Persetujuan Publikasi Skripsi Untuk Kepentingan Akademis.....	v
Abstrak.....	vi
Daftar Isi.....	viii
Daftar Gambar.....	x
Daftar Tabel.....	xii
BAB 1.....	1
PENDAHULUAN.....	1
1.1 LATAR BELAKANG.....	1
1.2 TUJUAN.....	3
1.3 BATASAN MASALAH.....	3
1.4 SISTEMATIKA PENULISAN.....	3
BAB 2.....	5
JURNALISTIK, JAVA MICROEDITION, WEBSITE, MYSQL, DAN KOMPRESI-DEKOMPRESI.....	5
2.1 JURNALISTIK.....	5
2.1.1 <i>Mobile Journalism</i>	5
2.1.2 <i>Citizen Journalism</i>	6
2.1.3 <i>Mobile Citizen Media</i>	8
2.2 JAVA MICROEDITION.....	9
2.2.1 <i>Configuration</i>	9
2.2.2 <i>Mobile Information Device Profile(MIDP)</i>	10
2.2.3 <i>API</i>	12
2.3 WEBSITE.....	13
2.4 HTML.....	14

2.5 PHP	15
2.6 MYSQL	16
2.7 KOMPRESI DAN DEKOMPRESI	16
BAB 3	19
PERANCANGAN MOBILE CITIZEN MEDIA	19
3.1 ARSITEKTUR MOBILE CITIZEN MEDIA	19
3.1.1 <i>HANDSET</i>	20
3.1.2 <i>JARINGAN</i>	22
3.1.3 <i>SERVER</i>	25
3.2 DIAGRAM ALIR MOBILE CITIZEN MEDIA	27
3.3 DIAGRAM ALIR PERANGKAT LUNAK PADA <i>HANDSET</i>	29
3.3.1 <i>Arsitektur Perangkat Lunak</i>	29
3.3.2 <i>Diagram Alir Perangkat Lunak</i>	30
3.4 DIAGRAM ALIR <i>WEBSITE</i>	33
3.4.1 <i>Arsitektur Website</i>	33
3.4.2 <i>Diagram Alir Website</i>	34
BAB 4	36
ANALISIS DAN UJICOBA	36
4.1 SKENARIO UJICOBA	36
4.1.1 <i>Instalasi Perangkat Lunak MCM pada Handset</i>	37
4.1.2 <i>Instalasi Website pada Server</i>	38
4.1.3 <i>Langkah-langkah menjalankan MCM pada Handset</i>	39
4.2 UJICOBA EFISIENSI KOMPRESI	42
4.3 UJICOBA PADA JARINGAN GSM	45
4.4 UJICOBA PADA JARINGAN UMTS	47
4.5 PERBANDINGAN WAKTU UPLOAD PADA JARINGAN GSM DAN UMTS	49
4.6 UJICOBA FITUR TAMBAHAN	51
4.6.1 <i>Mini Browser</i>	51
4.6.2 <i>BIAYA</i>	52
BAB 5	54
KESIMPULAN	54
DAFTAR REFERENSI	55
LAMPIRAN	58

DAFTAR GAMBAR

	Halaman
<i>Gambar 2.1</i> Arsitektur J2ME.....	5
<i>Gambar 2.2</i> Daur hidup MIDLet	8
<i>Gambar 2.3</i> Hubungan antara Data Source, Player, Control dan Manager.....	9
<i>Gambar 2.4</i> Hubungan antara JSR 75 dengan MIDP dan CLDC	10
<i>Gambar 3.1</i> Arsitektur Mobile Citizen Media	16
<i>Gambar 3.2</i> Jaringan GSM untuk GPRS.....	21
<i>Gambar 3.3</i> Jaringan UMTS	21
<i>Gambar 3.4</i> Jaringan Internet	22
<i>Gambar 3.5</i> Perangkat keras Server.....	23
<i>Gambar 3.6</i> User Interface XAMPP server	24
<i>Gambar 3.7</i> Squence Diagram Mobile Citizen Media.....	25
<i>Gambar 3.8</i> Arsitektur Software pengiriman informasi pada Mobile Citizen Media di telepon seluler.....	27
<i>Gambar 3.9</i> Arsitektur software untuk Internet Browser pada Mobile Citizen Media di telepon seluler.....	27
<i>Gambar 3.10</i> Diagram alir software handset Mobile Citizen Media	26
<i>Gambar 3.11</i> Arsitektur website pada Mobile Citizen Media.....	30
<i>Gambar 3.12</i> Diagram alir website pada Mobile Citizen Media	32
<i>Gambar 4.1</i> Proses instalasi software pada ponsel.....	34
<i>Gambar 4.2</i> Proses instalasi software selesai	35
<i>Gambar 4.3</i> Tampilan XAMPP pada Internet Browser.....	35
<i>Gambar 4.4</i> Tampilan Form Registrasi.....	36
<i>Gambar 4.5</i> Tampilan Menu Utama	37
<i>Gambar 4.6</i> Tampilan pilihan perintah	37
<i>Gambar 4.7</i> Tampilan saat berita siap diupload.....	38

<i>Gambar 4.8 Tampilan saat berita telah terkirim.....</i>	<i>38</i>
<i>Gambar 4.9 Grafik hasil efisiensi kompresi terhadap ukuran data asli</i>	<i>41</i>
<i>Gambar 4.10 Grafik waktu pengunggahan terhadap variasi ukuran data dari berbagai operator seluler GSM di Indonesia</i>	<i>43</i>
<i>Gambar 4.11 Grafik waktu pengunggahan terhadap variasi ukuran data dari berbagai operator seluler UMTS di Indonesia</i>	<i>45</i>
<i>Gambar 4.12 Grafik waktu pengunggahan terhadap variasi ukuran data dari berbagai operator seluler UMTS di Indonesia</i>	<i>47</i>
<i>Gambar 4.13 Tampilan saat mengakses mini Browser</i>	<i>48</i>
<i>Gambar 4.14 Tampilan berita yang diunduh.....</i>	<i>49</i>

DAFTAR TABEL

	Halaman
<i>Tabel 3.1 Gambar dan Spesifikasi Ponsel</i>	18
<i>Tabel 4.1 Spesifikasi perangkat keras komputer server</i>	33
<i>Tabel 4.2 Data hasil efisiensi kompresi</i>	36
<i>Tabel 4.2 Data pengujian pada jaringan GSM</i>	38
<i>Tabel 4.3 Data pengujian pada jaringan UMTS</i>	40
<i>Tabel 4.4 Data perbandingan waktu upload pada jaringan UMTS dan GSM</i>	41
<i>Tabel 4.5 Data perbandingan biaya dari tarif paling mahal dan paling murah</i>	44

BAB 1

PENDAHULUAN

1.1 LATAR BELAKANG

Dalam dunia jurnalistik, ada sebuah istilah yang dikenal dengan *Citizen Journalism* dan *Citizen Media*. *Citizen Journalism* adalah kegiatan mengumpulkan, melaporkan, menganalisis, dan penyebaran berita atau informasi yang dilakukan bukan oleh seorang wartawan [1]. Sedangkan *Citizen Media* adalah media yang memfasilitasi proses pengumpulan dan penyebaran informasi yang dilakukan masyarakat setiap hari. Pada umumnya, informasi dipublikasikan dan dapat diakses oleh masyarakat umum melalui internet, televisi, radio, media cetak, DVD dan berbagai bentuk lain. Kadang *Citizen Media* melaporkan berita yang tidak terakomodir oleh media-media konvensional pada umumnya. Sebuah contoh dari citizen media melalui siaran radio adalah Pacifica Radio Network yang beroperasi secara independen, didukung oleh pendengarnya, dan merupakan organisasi penyiaran non komersial [2]. Pacifica Radio Network mendukung lima stasiun swasta yang berbeda di Amerika. Di Kanada, ada sebuah media televisi publik yang mengikutsertakan masyarakat dalam materi penyiarannya. Contoh di internet mengenai citizen media adalah *weblogs*, *podcast*(*audio broadcast online*), kolaborasi wiki, dan *vblogs*(*video blogs*), dan *email list*. Adalah sebuah situs internet kolaborasi Gooh, dimana warga Argentina berbagi berita yang bermanfaat, tetapi tidak tertampung dalam media konvensional [3]. Contoh lain di Korea Selatan, Ohmynews, media yang mengikutsertakan masyarakat secara penuh dalam memperoleh berita [4].

Pemanfaatan telepon seluler dalam *Citizen Media* akan melahirkan sebuah bentuk baru dari *Citizen Journalism*. Didukung perkembangan teknologi telepon seluler yang dapat tersambung dengan internet disamping fitur dasar seperti pengiriman SMS(*short messaging services*), MMS(*multi media messaging services*) memungkinkan *Citizen Journalism* untuk bertukar informasi secara

cepat, mudah, dan murah dari internet dan ke perangkat bergerak lain. Penggunaan telepon seluler yang meningkat dari tahun ke tahun, sebagai contoh, pelanggan telepon seluler di India tumbuh 91% dan 54% di Afrika dalam rentang tahun 2000-2005 [5]. Sebuah alasan dari pertumbuhan seluler yang besar di negara berkembang, kemungkinan adalah pembangunan stasiun seluler lebih murah dari jaringan telepon berbasis kabel. Eric Sundelof, seorang Reuters Digital Vision Fellow dan pencipta situs media mobile akar rumput di *InTheFieldOnline.net*, mengemukakan bahwa, “Lebih murah membangun infrastruktur untuk telepon seluler, contoh di Afrika adalah tempat terbaik untuk meluncurkan berbagai layanan telepon seluler karena tanahnya yang relatif datar sehingga pembangunannya tidak akan menjumpai masalah dengan stasiun pemancar, dan tidak dibutuhkan banyak stasiun pemancar” [6]. Bukti lain adalah sejak 2002, jumlah telepon seluler di dunia telah melebihi jumlah total nomor telepon rumah, dan pada tahun 2006, jumlahnya meningkat mencapai 2,5 miliar nomor sedangkan *fixed* telepon hanya berjumlah 1,4 miliar nomor [7]. Di Indonesia sendiri, pada tahun 2008 pemakai telepon seluler telah mencapai 80 juta dan diperkirakan pada tahun 2010 pemakaian telepon seluler mencapai 120 jt dan pengguna internet 80, 2 juta serta *broadband* 2,5 juta [8]. Dukungan lain adalah fitur-fitur multimedia, seperti pemutar MP3, kamera digital, dan perekam video yang terintegrasi dalam telepon seluler. Ditambah dukungan dari produsen *handset* seperti Motorola, Samsung dan lainnya yang memproduksi *handset* seharga kurang dari US\$30 dan ditargetkan pada masyarakat yang tinggal di negara-negara berkembang [9]. Hal ini akan makin membuka partisipasi dari seluruh lapisan masyarakat.

Hal yang menarik lainnya adalah *website* sebagai wadah yang dapat menampung berbagai bentuk informasi baik berupa pesan teks, gambar, suara, dan video. Sehingga tidak berlebihan apabila *website* disebut dengan media yang *powerfull*. Seperti YouTube, *website* yang menyediakan layanan video. Contoh lain adalah Utterz yang mengizinkan seseorang orang menelepon nomor kode negara tertentu dan merekamnya menjadi sebuah podcast, yang dapat secara otomatis dipublikasikan di internet.

Pada skripsi ini dijelaskan suatu rancangan sistem yang berjudul *Mobile Citizen Media*, yang diharapkan dapat memfasilitasi kegiatan *Citizen Journalism*. *Mobile Citizen Media* adalah sistem terpadu yang terdiri dari tiga komponen yaitu pengirim, pengolah, dan penyaji. Pengirim adalah masyarakat yang telah terdaftar. Pengirim dibantu dengan sebuah telepon seluler dan sebuah perangkat yang dibangun menggunakan program J2ME. Selanjutnya, pengolah dan penyaji adalah website yang bertugas mengolah dan menampilkan berita yang diterima. Ketiga komponen ini terhubung melalui jaringan internet.

1.2 TUJUAN

Tujuan penulisan skripsi ini adalah membuat sebuah rancangan perangkat lunak sistem informasi *Mobile Citizen Media* terintegrasi meliputi pengirim, pengelola dan penerima.

1.3 BATASAN MASALAH

Pada penulisan skripsi ini, penulis membatasi masalah pada rancang bangun perangkat lunak *Mobile Citizen Media* dan analisis penggugahan data melalui jaringan telepon seluler berbasis GSM dan UMTS. Rancang bangun perangkat lunak pada *handset* menggunakan bahasa pemrograman Java, sedangkan pada website menggunakan PHP dan *database* MySQL. Perangkat lunak ini akan disertakan fitur kompresi pada *handset* dan dekompresi pada *website*. *Handset* yang digunakan adalah *handset* yang mempunyai fitur MIDP 2.0 dan CLCD 1.1.

1.4 SISTEMATIKA PENULISAN

Bab 1 Pendahuluan. Pada bagian ini, dibahas latarbelakang, tujuan, batasan masalah dan sistematika penulisan dari penulisan skripsi ini.

Bab 2 Dasar Teori. Pada bagian ini dibahas prinsip dasar pengiriman informasi dilanjutkan dengan pembahasan alat-alat yang digunakan untuk mendukung pembuatan sistem ini meliputi bahasa pemrograman Java Microedition, Web, PHP, *database* MySQL, dan penjelasan tentang kompresi data.

Bab 3 Metodologi Penelitian. Pada bagian ini dijelaskan tahapan-tahapan pembuatan sistem aplikasi *Mobile Citizen Media*, meliputi penjelasan arsitektur *Mobile Citizen Media*, diagram alir perangkat lunak pada *handset*, diagram alir *website*.

Bab 4 Analisis dan Ujicoba. Pada bagian ini dijelaskan hasil pengujian dan evaluasi ujuk kerja *Mobile Citizen Media* yang dijalankan pada jaringan GSM dan UMTS.

Bab 5 Kesimpulan.



BAB 2

JURNALISTIK, JAVA MICROEDITION, WEBSITE, MYSQL, dan KOMPRESI- DEKOMPRESI

2.1 JURNALISTIK

Dalam kaitannya dengan penyampaian informasi, dikenal dengan istilah jurnalistik atau jurnalisme. Jurnalistik mempunyai beberapa definisi, menurut F. Fraser Bond dalam bukunya *An Introduction to Journalism* menyatakan: “*Journalism embraces all the forms in which and through which the news and moment on the news reach the public*”. Jurnalistik adalah segala bentuk yang membuat berita dan ulasan mengenai berita sampai pada kelompok pemerhati. Menurut Adinegoro, jurnalistik adalah semacam kepandaian karang-mengarang yang pokoknya memberi perkarabaran pada masyarakat dengan selekas-lekasnya agar tersiar seluas-luasnya. Sedang menurut Summanang, mengutarakan lebih singkat lagi, jurnalistik adalah segala sesuatu yang menyangkut kewartawanan. Sedikit berbeda jika menurut kamus, jurnalistik diartikan sebagai kegiatan untuk menyiapkan, mengedit, dan menulis surat kabar, majalah, atau berkala lainnya[18][25].

2.1.1 *Mobile Journalism*

Dalam bidang jurnalistik, hal menarik yang pantas untuk dicermati adalah tentang *Mobile Journalism*. *Mobile Journalism* adalah kegiatan peliputan berita yang menggunakan peralatan seperti kamera digital, perekam video dan komputer jinjing yang dilengkapi dengan jaringan wireless broadband. Kegiatan peliputan berita seperti merekam gambar (*images* atau video), mengetik, mengedit, menyunting dan mengirimkan berita dapat dilakukan langsung dari lapangan, bahkan pada saat jurnalis sedang bergerak. Ada dua keunggulan penting yang dimiliki oleh *Mobile Journalism*, pertama adalah sederhana. Arti sederhana disini lebih menitik beratkan pada faktor ukuran. Dengan hanya bermodalkan sebuah *handset*, jurnalis bisa melakukan aktivitas jurnalistiknya. Bahkan karena

begitu ringkas, jurnalis hanya cukup meletakkan *handset*nya dalam saku, tidak membutuhkan ruang yang besar. Keuntungan yang kedua adalah mandiri. Perbedaan antara jurnalis konvensional dan jurnalis bergerak tidak berhenti pada alat kerja yang serba ringkas, tetapi juga pada cara kerja yang serba mandiri. Dengan alat seringkas itu, wartawan tidak hanya berhenti pada menulis berita, tetapi juga bisa mengedit, menyunting, membuat tata letak, menyisipkan suara dan gambar (foto atau video) sebelum berita itu ia kirim[18].

Dengan cara kerja demikian, selain dituntut lebih mandiri, jurnalis bergerak ini juga telah memotong rangkaian kerja yang panjang sebuah manajemen berita. Sederhana saja, pekerjaan editor, penyunting, penyelarasa bahasa, dan penata letak sudah tidak diperlukan lagi. Bukan sekadar menyingkat waktu, tetapi juga tenaga kerja. Mandiri dan profesional adalah tuntutan bagi jurnalis yang bekerja untuk jurnalisme bergerak[18].

2.1.2 Citizen Journalism

Fenomena lain yang menarik adalah kegiatan jurnalistik yang tidak dilakukan oleh jurnalis/wartawan. Kegiatan ini dikenal dengan *citizen journalism*. *Citizen journalism* adalah kegiatan mengumpulkan, melaporkan, menganalisis, dan penyebaran berita atau informasi yang dilakukan bukan oleh seorang wartawan[18][1].

Menurut Steve Outing, *citizen journalism* terbagi dalam sebelas kategori:

- Membuka ruang untuk komentar publik. Pembaca dapat menyampaikan reaksinya, pujian, kritikan dalam artikel penulis. Biasanya blog mecantumkan kotak bagi para pembaca untuk memberi komentar di dalamnya. Pada media cetak konvensional, jenis ini biasa disebut ruang surat pembaca
- Menambah pendapat masyarakat dalam cerita yang ditulis jurnalis profesional. Warga diminta untuk ikut menuliskan pengalamannya pada sebuah topik utama liputan yang dilaporkan jurnalis.
- *Open source* atau *participatory journalism*. Kolaborasi antara jurnalis profesional dengan masyarakat non-jurnalis yang memiliki kemampuan dalam bidang yang dibahas dalam artikel, sebagai bantuan keakuratan

artikel. Terkadang masyarakat non jurnalis ini juga menjadi kontributor tunggal yang menghasilkan artikel. Contoh : The Spokesman-Review/APME reader panel.

- *Citizen bloghouse*. Website yang bisa dibuat sendiri oleh warga, berisi cerita maupun pemikiran pemilik blog. Contoh: blogspot.com
- *Newsroom citizen 'transparency' blogs*, merupakan blog yang disediakan untuk upaya transparansi organisasi sebuah media, dimana pembaca bisa memasukkan keluhan, kritikan, atau pujian atas pekerjaan media tersebut.
- *Stand-alone citizen journalism site* yang melalui proses editing. Website berita yang hampir seluruhnya dikontribusi oleh warga. Di dalamnya ada editor situs yang menjaga kualitas laporan, dan mendidik warga tentang topik-topik yang menarik dan layak untuk dilaporkan. *Stand-alone citizen journalism site* : tanpa melalui proses editing. Hampir serupa dengan nomor 6, hanya saja informasi yang muncul tidak melewati proses editing dari editor situs. Namun, sebaiknya situs mempunyai sarana keamanan, sehingga tulisan-tulisan yang tidak tepat bisa segera dibuang.
- *Stand-alone citizen journalism website* dengan tambahan edisi cetak. Konsepnya bisa diambil dari tipe nomor 6 atau 7, dengan dibuat edisi cetaknya. Biasanya edisi cetak seperti ini adalah insert dari edisi harian atau mingguan.
- *Hybrid: Pro+citizen journalism*. Suatu kerja organisasi media yang menggabungkan pekerjaan jurnalis profesional dengan jurnalis warga. Seperti pada Oh My News, yang mayoritas content-nya (80%) diisi oleh warga, dan sisanya diisi oleh jurnalis profesional. Namun semua tulisan akan disortir oleh staf editor. Editor berperan dalam menilai dan memilih berita yang akan diangkat ke halaman utama.
- Penggabungan antara jurnalis profesional dan jurnalis warga dalam satu atap. Bedanya dengan nomor 9 adalah dalam satu halaman, bisa saja terdapat tulisan dari jurnalis profesional yang dibayar dan kontribusi warga (gratis) tanpa dinding pembatas[18].

- Model Wiki, dimana pembaca sekaligus bertindak sebagai editor. Setiap orang bisa menulis artikel, dan setiap orang bisa memberi tambahan atau komentar pada artikel yang muncul. Contoh: Wikimu.com.

Di Indonesia, *citizen journalism* berkembang tahun 2005 diantaranya dengan munculnya situs *halamansatu.net*, *wikimu.com* dan *panyingkul.com* yang hadir dengan motto jurnalisme orang biasa[18].

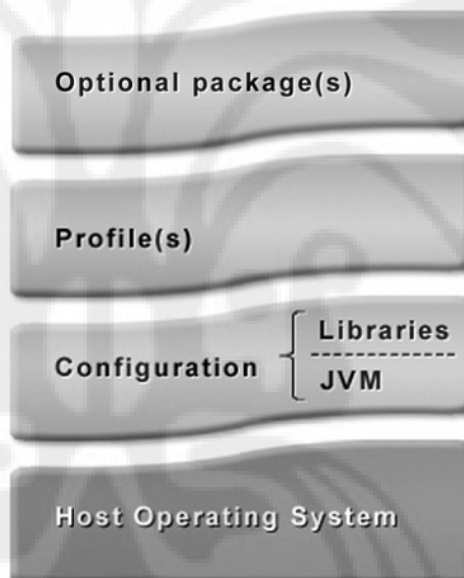
Pada umumnya situs yang menerapkan *citizen journalism* biasanya menampilkan berita-berita yang tidak terakomodasi oleh media massa konvensional. Berita-berita yang tersaji sangat beragam dan dipengaruhi oleh pengirim berita dan komunitas[18].

2.1.3 Mobile Citizen Media

Dengan mengacu pada pengertian *Mobile Journalism*, *Citizen Journalism*, dan *Citizen Media* pada sub bab Latarbelakang, 2.1.1 dan sub bab 2.1.2 maka *Mobile Citizen Media* adalah sebuah media yang memfasilitasi masyarakat untuk melakukan kegiatan jurnalistik, mengumpulkan, menganalisis, melaporkan dan menyebarkan berita, yang dibantu dengan perangkat bergerak seperti ponsel, pager, dan PDA.

2.2 JAVA MICROEDITION

Java Microedition(J2ME) adalah salah satu *platform* java yang digunakan untuk membuat perangkat lunak pada peralatan-peralatan yang bersifat pervasif yang memiliki keterbatasan memori dan sumber daya lainnya, seperti pager, ponsel dan PDA. Gambar 2.1 menunjukkan arsitektur J2ME. J2ME ditargetkan untuk peralatan perfasif yang memiliki sumber daya yang beranekaragam, sehingga J2ME didesain terdiri dari beberapa spesifikasi yang membentuk sebuah *platform*. Spesifikasi-spesifikasi yang digunakan untuk membentuk *platform* pada suatu peralatan pervasif tergantung dari sumber daya yang dimiliki peralatan tersebut. J2ME terdiri dari *configuration* dan *profile*, serta beberapa API seperti yang ditunjukkan pada Gambar 2.1[12].



Gambar 2.1 Arsitektur J2ME[10]

2.2.1 Configuration

Configuration pada J2ME berfungsi untuk menspesifikasikan fitur dari bahasa pemrograman java dan *java virtual machine* yang akan digunakan, serta menentukan *library* dan API(*Application Programming Interface*) yang akan digunakan. Pada J2ME terdapat dua buah konfigurasi, yaitu CDC(*Connected Device Configuration*) dan CLDC (*Connected, Limited Device Configuration*)[12].

2.2.1.1 CDC(Connected Device Configuration)

CDC adalah *configuration* yang digunakan pada peralatan *high-end* yang memiliki memori lebih dari 2 Megabyte baik RAM maupun ROM seperti TV dan *smart phone*. Selain itu peralatan yang mendukung CDC memiliki *processor* yang lebih baik daripada peralatan *low-end* yang menggunakan CLDC. Pada CDC digunakan CVM(*C-Virtual Machine*) sebagai *intepreter*-nya. CVM adalah *intepreter* yang menyerupai JVM(*Java Virtual Machine*) pada J2SE, perbedaannya hanyalah pada letak *garbage collection* pada *virtual machine*, sehingga CVM dapat membuat algoritma *garbage collection* yang lebih singkat dan dapat digunakan secara berulang-ulang[12].

2.2.1.2 CLDC(Connected, Limited Device Configuration)

CLCD adalah *configuration* yang digunakan pada peralatan *low-end* yang memiliki memori 160-512 *Kilobyte*, *processor* 16 atau 32 bit, serta konsumsi daya yang kecil. *Configuration* yang dimiliki CLDC dibuat berdasarkan J2SE yang dikurangi beberapa fungsinya. Beberapa fitur J2SE yang dikurangi pada CLDC antara lain :

- -*Java Native Interface*(JNI)
- -Operasi *floating-point*
- -Finalisasi Objek
- -Paket *java.lang.error*
- -*Daemon thread*

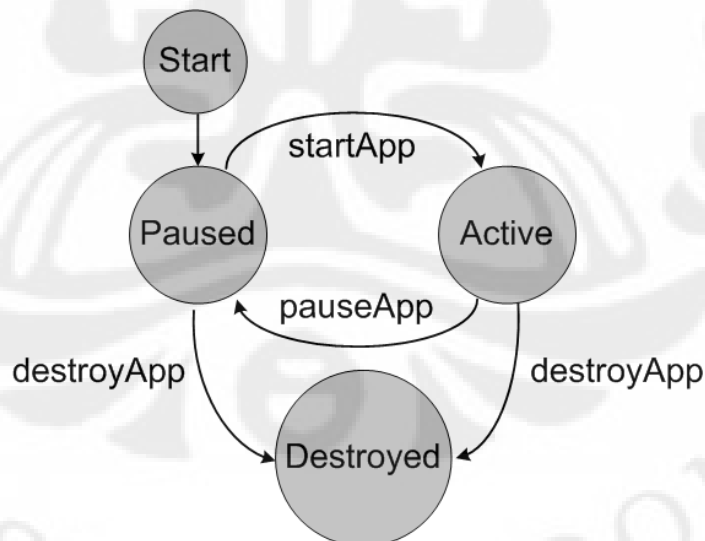
Pada CLDC *virtual achine* yang digunakan adalah KVM(*Kilobyte Virtual Machine*). KVM adalah paket JVM yang didesain untuk perangkat dengan sumber daya yang terbatas. Pada saat ini terdapat CLDC 1.0 dan CLDC 1.1[12].

2.2.2 Mobile Information Device Profile(MIDP)

Profile adalah bagian dari J2ME yang menyediakan API untuk satu jenis peralatan tertentu, misalnya PDAP(*PDA Profile*), yang menyediakan API khusus pada PDA. Salah satu *profile* yang banyak digunakan adalah MIDP yang digunakan pada ponsel, pager, dan Palm OS. Pada saat ini terdapat MIDP 1.0, MIDP 2.0, dan MIDP 2.1, pada MIDP 2.0 terdapat fitur tambahan berupa Mobile Media API(MMAPI) yang berfungsi untuk memainkan *file* audio dan video,

sedangkan pada MIDP 2.1 terdapat fitur tambahan berupa SVG(*Scalable Visual Graphic*)[12].

Aplikasi yang berjalan pada MIDP disebut *Midlet*. *Midlet* terdiri dari minimal sebuah kelas turunan dari kelas abstrak MIDP (*javax.microedition.midlet.MIDLET*). *Midlet* terdiri dari beberapa metode, yaitu *constructor()*, *protected void startApp() throws MidletStateChangeExeption*, *protected void pauseApp()*, *protected void destroyApp(boolean unconditional) throws MidletStateChange Exeption*. Semua metode ini menentukan alur hidup dari *midlet* yang dibuat. Alur hidup *midlet* ini ditunjukkan pada Gambar 2.2. Dimulai ketika *midlet* dijalankan. Awal alur hidup *midlet* dimulai dengan proses inisiasi *midlet* ke dalam kondisi *pause* dengan dijalankannya metode *pauseApp()*, kemudian *midlet* dijalankan dengan metode *startApp()*, metode ini diimplementasikan sebagai *protected* agar *midlet* tidak dapat memanggil metode ini. Untuk keluar dari *midlet* digunakan metode *destryApp()*, metode ini akan memanggil *noyifyDestroyed ()* yang akan memberitahu *platform* utama utama untuk memberhentikan *midlet* yang sedang dijalankan dan melepaskan semua sumber daya yang digunakan *midlet* ini agar dapat digunakan oleh *midlet* lain[12].



Gambar 2.2 Daur hidup MIDLet[12]

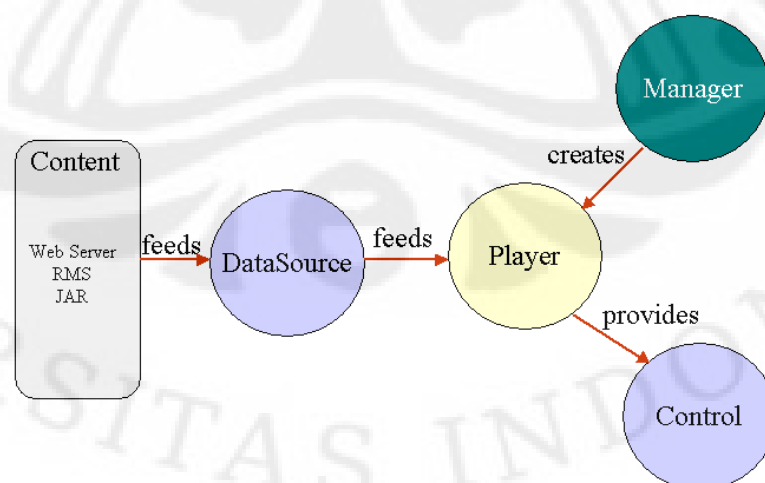
2.2.3 API

Selain *configuration* dan *profile*, J2ME juga dilengkapi beberapa API yang mendukung *platform* ini dengan fungsi-fungsi khusus. Beberapa API yang digunakan dalam skripsi ini adalah MMAPI, *File Connection*, *Web Services* 1.0[12].

2.2.3.1 MMAPI(Mobile Media API)

MMAPI adalah API yang dapat digunakan untuk menambah fungsi-fungsi multimedia pada peralatan pervasif yang menggunakan J2ME. API ini diperkenalkan melalui JSR 135 oleh *Sun Microsystem*. MMAPI bersifat *agnostic* terhadap protokol dan format, sehingga API ini dapat digunakan pada sebuah devais tanpa bergantung pada protokol dan format tertentu yang dapat didukung devais tersebut[12].

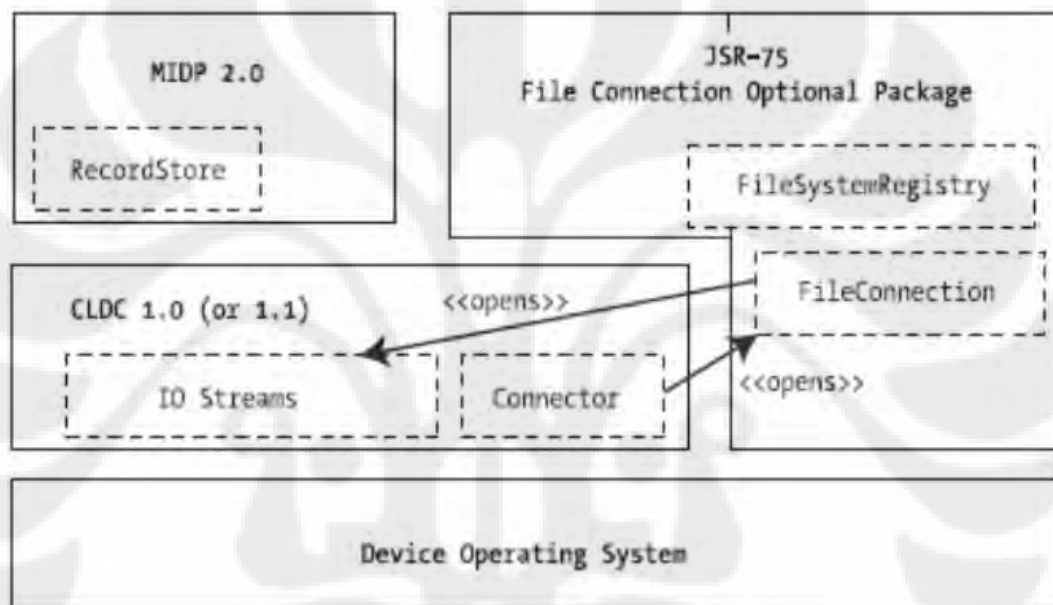
Arsitektur dari MMAPI terdiri dari tiga bagian utama, yaitu *Player*, *DataSource* dan *Manager*. *Player* adalah *interface* yang berkaitan dengan fungsi memainkan dan mengolah data-data *multimedia*. Sedangkan *DataSource* adalah kelas abstrak yang mengenkapsulasi fungsi pengaksesan data multimedia dan pengaturan protokol yang digunakan. Untuk menjembatani fungsi-fungsi dari *Player* dan *DataSource* diperlukan sebuah *Manager*. Ketiga bagian ini didefinisikan masing-masing pada paket *javax.microedition.media*, dan *javax.microedition.media.protocol*. Hubungan antara *Player*, *DataSource* dan *Manager* dapat diilustrasikan pada Gambar 2.3[12].



Gambar 2.3 Hubungan antara Data Source, Player, Control dan Manager[13]

2.2.3.2 File Connection API

Pada saat ini peralatan pervasif khususnya ponsel sudah didukung dengan *file* sistem dengan memori yang besar, dengan menambahkan *SD card* dan *memory stick*. *File Connection API* adalah bagian dari *PDA Optional Package* yang diperkenalkan melalui *JSR 75* yang berfungsi untuk mengakses *file* sistem dari suatu devais pervasif. API yang terdapat pada paket *javax.microedition.io.file* ini dapat digunakan untuk membaca dan menulis *file*, dan membuat *file* atau direktori. Hubungan antara API ini dengan *Configuration* adan *Profile* dideskripsikan pada Gambar 2.4[12].



Gambar 2.4 Hubungan antara JSR 75 dengan MIDP dan CLDC[14]

2.3 WEBSITE

Website diartikan sebagai kumpulan halaman yang digunakan untuk menampilkan informasi berupa teks, gambar, suara, video, dan gabungan semuanya yang biasa dikenal dengan multimedia baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian yang saling terkait antara satu halaman dengan halaman lainnya[18].

Sedangkan menurut terminologi, *website* adalah kumpulan dari halaman-halaman situs, yang biasanya terangkum dalam sebuah domain atau

subdomain, yang tempatnya berada didalam *World Wide Web* (WWW) di Internet. Sebuah *web page* adalah dokumen yang ditulis dalam format HTML (*Hyper Text Markup Language*), yang hampir selalu bisa diakses melalui HTTP, yaitu protokol yang menyampaikan informasi dari *server website* untuk ditampilkan kepada para pemakai melalui *web browser*. Semua publikasi dari *website-website* tersebut dapat membentuk sebuah jaringan informasi yang sangat besar[18].

Halaman-halaman dari *website* akan bisa diakses melalui sebuah URL yang biasa disebut *Homepage*. URL ini mengatur halaman-halaman situs untuk menjadi sebuah hirarki, meskipun, *hyperlink-hyperlink* yang ada di halaman tersebut mengatur para pembaca dan memberitahu mereka susunan keseluruhan dan bagaimana arus informasi ini berjalan[18].

Ada dua jenis website berdasarkan sifatnya yaitu website statis dan dinamis. Website statis adalah salah satu bentuk website yang isi didalam website tersebut tidak dimaksudkan untuk di *update* secara berkala, dan biasanya dikelola secara manual oleh beberapa orang yang menggunakan *software editor* oleh pembuatnya. Sedangkan *website* dinamis adalah *website* yang dapat diperbaharui isi informasinya baik secara berkala maupun tidak. *Website* ini memungkinkan pengguna dapat saling berinteraksi satu sama lain. Dengan metode variabel database, variabel sesi, dll membuat website ini sangat inteaktif[18].

Untuk membangun sebuah website dinamis, ada beberapa bahasa pemrograman yang bisa digunakan antara lain *ColdFusion (CFM)*, *Active Server Pages (ASP)*, *Java Server Pages (JSP)* dan PHP. Agar tampilan semakin menarik maka ditambahkan dengan bahasa Java[18].

2.4 HTML

HTML (*HyperText Mark up Language*) merupakan suatu metode untuk mengimplementasikan konsep *hypertext* dalam suatu naskah atau dokumen. HTML sendiri bukan tergolong pada suatu bahasa pemrograman karena sifatnya yang hanya memberikan tanda (*marking up*) pada suatu naskah teks dan bukan

sebagai program. Berdasarkan kata-kata penyusunnya HTML dapat diartikan lebih dalam lagi menjadi :

- **Hypertext**

Link hypertext adalah kata atau frase yang dapat menunjukkan hubungan suatu naskah dokumen dengan naskah-naskah lainnya. Jika kita klik pada kata atau frase untuk mengikuti link ini maka *web browser* akan memindahkan tampilan pada bagian lain dari naskah atau dokumen yang kita tuju.

- **Markup**

Pada pengertiannya di sini markup menunjukkan bahwa pada *file* HTML berisi suatu intruksi tertentu yang dapat memberikan suatu format pada dokumen yang akan ditampilkan pada *World Wide Web*.

- **Language**

Meski HTML sendiri bukan merupakan bahasa pemrograman, HTML merupakan kumpulan dari beberapa instruksi yang dapat digunakan untuk mengubah-ubah format suatu naskah atau dokumen.

Pada awalnya HTML dikembangkan sebagai subset SGML (*Standard Generalized Mark-up Language*). Karena HTML didedikasikan untuk ditransmisikan melalui media Internet, maka HTML relatif lebih sederhana daripada SGML yang lebih ditekankan pada format dokumen yang berorientasi pada aplikasi[18].

2.5 PHP

PHP (*Hypertext Preprocessor*) adalah jenis *middleware open source*, sehingga dapat digunakan oleh siapa saja dengan cuma-cuma. Selain gratis, PHP memiliki berbagai kelebihan, diantaranya untuk membuat *website* dinamis. PHP akan dijalankan pada sisi *server(server side)* sehingga klien tidak bisa melihat kode aslinya[18].

Dengan PHP, halaman *web* dapat dibuat lebih interaktif. Misalnya dengan memasukkan fungsi-fungsi tertentu. Kita juga akan dipermudah ketika melakukan *updating website*, karena PHP dapat berhubungan langsung dengan *database*[18].

Aturan penulisan PHP yang terintegrasi dalam HTML adalah sebagai berikut,

Kode PHP ditulis dengan diawali tanda `<?php`, dan diakhiri tanda `?>`.

Kode PHP yang disertakan atau disisipkan dalam kode HTML, *file* disimpan dalam *extensi* php.

Untuk memisahkan tiap baris kode PHP digunakan tanda *semicolon*, `;`.

Untuk menampilkan tulisan yang ditulis dalam kode PHP maka digunakan *sintaks* *echo*. Berikut contoh penulisan,

```
<?
echo "Ini tampil di halaman web site";
?>
```

Untuk tujuan manipulasi dan pengolahan data, dalam kode PHP terdapat penggunaan variabel. Penulisan variabel selalu diawali dengan tanda dolar, `[/code]`[18].

Penulisan kode program yang baik akan selalu menyertakan keterangan yang membantu programmer lain dalam memahami isi *source code* tersebut[18].

2.6 MYSQL

Database digunakan untuk penyimpanan data. Demikian pula dengan MySQL. Sistem database ini akan dikombinasikan dengan PHP sebagai sistem mengolah data informasi yang akan dikirim ke komputer klien untuk ditampilkan pada browser. Data pada MySQL dapat dipanggil, dihapus, atau ditambah melalui fungsi *query*.

Database pada MySQL terdiri dari tabel-tabel. Setiap tabel mempunyai kolom, baris, dan *record* untuk menyimpan data[18].

2.7 KOMPRESI dan DEKOMPRESI

Dalam komunikasi menggunakan perangkat seluler, ukuran data masih menjadi hal yang penting. Ukuran data akan sangat erat kaitannya dengan

biaya. Makin kecil ukuran data yang dikirim atau diterima akan semakin kecil pula biaya yang dibutuhkan. Oleh karena itu, dibutuhkan sebuah solusi untuk menguranginya, baik ukuran dan biaya. Kompresi adalah jawabannya.

Pada *Mobile Citizen Media* ini, penulis menggunakan kompresi LZW. Algoritma LZW dikembangkan oleh Terry A. Welch dari metode kompresi sebelumnya yang ditemukan oleh Abraham Lempel dan Jacob Ziv pada tahun 1977[19]. Pada proses kompresinya, menggunakan teknik *dictionary*(kamus). Setiap huruf atau karakter akan dimasukkan ke dalam tabel, dimana tabel ini berfungsi sebagai kamus. Dalam fungsinya sebagai kamus, akan menjalankan tugasnya sebagai referensi pada setiap karakter yang akan dikodekan. Ukuran tabel pada algoritma LZW asli adalah 4096 sampel atau 12 bit, dimana 256 sampel pertama digunakan untuk table karakter single (Extended ASCII), dan sisanya digunakan untuk pasangan karakter atau string dalam data input. Kompresi ini cocok diterapkan pada *Mobile Citizen Media* karena dalam salah satu input beritanya berupa teks/tulisan. Dimana teks tersebut adalah kumpulan dari karakter ASCII.

- **Kompresi**

Berikut adalah algoritma kompresi LZW,

```

w = NIL
while (baca karakter k)
{
If wk ada di tabel dictionary then
w=wk
else
add wk ke dictionary;
output kode =w;
w=k;
}

```

Algoritma LZW melakukan pembacaan satu karakter k pada satu waktu. Karakter k dipasangkan dengan karakter sebelumnya yaitu w , setiap pasangan karakter ini dimasukkan ke dalam tabel. Lalu dibaca lagi karakter k , jika

pasangan w dan k ada di dalam tabel maka w diisi oleh wk dan output kode adalah w , demikian seterusnya hingga aliran data habis.

- **Dekompresi**

Berikut adalah algoritma dekompresi LZW,

```

baca karakter k;
output k;
w = k;
while (baca karakter k)
{
  entry= dictionary entry untuk k
  ouput entry
  add w+entry[0] ke dictionary;
  w=entry;
}

```

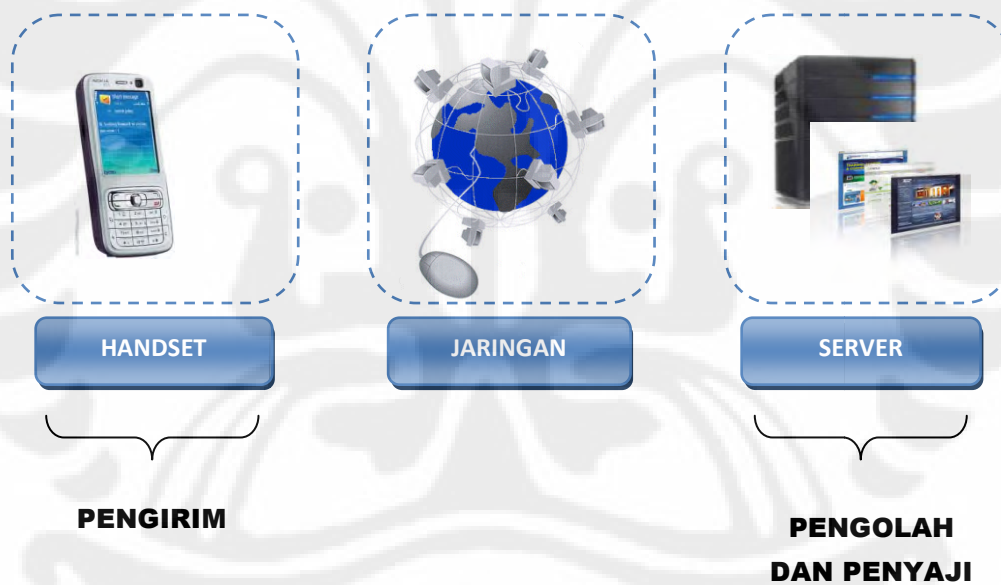
Untuk melakukan dekompresi, tidak diperlukan tabel terdahulu pada proses kompresi data, dan tabel tidak perlu diikutkan dalam data kompresi, dekompresi membuat sendiri tabelnya. Jika ada masukan data karakter k , maka output kode adalah k , selanjutnya nilai w diisi dengan karakter k . Ada satu variable tambahan yaitu masukan yang digunakan untuk mengambil satu karakter dari k untuk dijadikan kamus atau *dictionary*.

BAB 3

PERANCANGAN MOBILE CITIZEN MEDIA

3.1 ARSITEKTUR MOBILE CITIZEN MEDIA

Seperti penjelasan yang terdapat pada sub bab Latarbelakang, bahwa *Mobile Citizen Media* adalah sistem terpadu yang terdiri dari tiga komponen utama yaitu pengirim, pengolah, dan penyaji, hal ini akan menjadi dasar dalam merancang arsitektur sistem secara lebih rinci. Pada Gambar 3.1, digambarkan arsitektur Mobile Citizen Media terdiri dari empat blok yaitu *handset*, *network access*, internet, dan website-database. Masing-masing blok akan dijelaskan lebih rinci pada subbab di bawah ini.







Gambar 3.1 Arsitektur Mobile Citizen Media

3.1.1 HANDSET

Handset atau ponsel pada sistem *Mobile Citizen Media*, digunakan sebagai alat bantu bagi komponen pengirim, *Citizen Journalism*, untuk mengumpulkan dan mengirim informasi dengan berbagai bentuk (teks/tulisan, suara, gambar, dan video). *Handset* atau ponsel yang dapat digunakan harus memenuhi syarat minimum, yaitu pertama, *handset* atau ponsel telah terintegrasi J2ME dengan dukungan MIDP 2.0 dan CLCD 1.1. Yang kedua, *handset* atau ponsel dapat mengakses internet, baik menggunakan jaringan GSM, UMTS, CDMA, WiFi, atau pun jenis-jenis jaringan lain. Pada Tabel 4.1, ditunjukkan daftar spesifikasi ponsel dari beberapa merk yang beredar di masyarakat. Syarat tambahan yang sebaiknya dimiliki oleh handset atau ponsel adalah kemampuan merekam dan memainkan *file* audio dan video serta mengambil gambar dengan format jpeg. Format *file* audio pada umumnya .amr, .mp3, dan .wav. Sedangkan format *file* video pada umumnya .3gp dan .mp4.

Tabel 3.1 Gambar dan Spesifikasi Ponsel

No	Gambar	Keterangan
1	 <p data-bbox="363 1563 997 1675">Sumber : http://www.gsmarena.com/sony_ericsson_w890-2172.php</p>	<ol style="list-style-type: none"> <li data-bbox="1098 1205 1369 1301">1. 2G Network : GSM 850 / 900 / 1800 / 1900 <li data-bbox="1098 1312 1369 1375">2. 3G Network : HSDPA 2100 <li data-bbox="1098 1386 1369 1413">3. Java : MIDP 2.0

2	 <p>Sumber: http://www.gsmarena.com/nokia_n73-1550.php</p>	<ol style="list-style-type: none"> 1. 2G Network : GSM 850 / 900 / 1800 / 1900 2. 3G Network : UMTS 2100 3. Java : MIDP 2.0
3	 <p>Sumber : http://www.gsmarena.com/samsung_t109-2550.php</p>	<ol style="list-style-type: none"> 1. 2G Network : GSM 850 / 1900 2. Java : MIDP 2.0
4	 <p>Sumber: http://tlinkn.manufacturer.globalsources.com/si/6008818893698/pdtl/CDMA-phone/1008448382/CDMA-Phone.htm</p>	<ol style="list-style-type: none"> 1. Network: CDMA 1x, 800/1,900MHz 2. Java : MIDP 2.0

5	 <p>Sumber : http://www.gsmarena.com/nokia_e65-1860.php</p>	<ol style="list-style-type: none"> 1. 2G Network : GSM 850 / 900 / 1800 / 1900 2. 3G Network : UMTS 2100 3. WLAN: Wi-Fi 802.11b/g 4. Java : MIDP 2.0
---	---	--

Semakin lengkap fitur yang dimiliki oleh *handset* atau ponsel, akan semakin banyak bentuk informasi yang dapat dikirimkan.

Selanjutnya agar bisa menjadi bagian dari sistem *Mobile Citizen Media*, *handset* atau ponsel harus dilengkapi dengan sebuah perangkat lunak yang rancangannya akan dibahas pada sub bab 4.2.

3.1.2 JARINGAN

Jaringan yang dimaksud pada sistem ini adalah media transmisi yang digunakan untuk melewatkan informasi yang berasal dari sebuah *handset* atau ponsel. Informasi, dalam proses perjalanannya menuju server, pada umumnya akan melalui dua jaringan yang berbeda. Dua jaringan itu adalah jaringan telepon seluler dan jaringan internet.

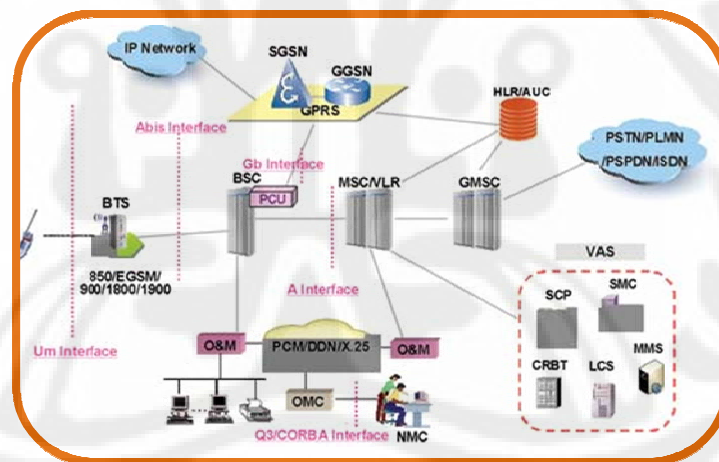
3.1.2.1 Jaringan Telepon

Jaringan telepon yang dimaksud dalam sistem ini adalah jaringan telepon seluler. Di Indonesia, pada saat tugas akhir ini ditulis, terdapat tiga jenis jaringan seluler, yaitu GSM(*Global System for Mobile communications*), UMTS(*Universal Mobile for Telecommunications System*), dan CDMA(*Code Division Multiple Access*). Informasi yang berasal dari *handset* akan melalui salah satu jenis jaringan ini. Pemilihan jaringan telepon yang akan dilewati informasi, tergantung pada jenis layanan yang digunakan oleh *handset* atau ponsel.

Arsitektur jaringan GSM, seperti pada Gambar 3.2, terdiri dari tiga komponen utama yaitu : *Mobile Station*, *Base Station Subsystem(BSS)*, dan *Network Subsystem(NSS)*.

Mobile Station adalah sebuah *handset* yang dilengkapi dengan *SIM card*. Setiap *handset* memiliki nomor *International Mobile Equipment Identity(IMEI)* yang unik sebagai identifikasinya. *SIM card* berisi nomor *IMSI(International Mobile Subscriber Identity)* yang digunakan sebagai identitas pelanggan ke sistem.

Base Station Subsystem(BSS) terdiri dari *BTS(Base Transceiver Station)* dan *BSC(Base Station Controller)*. *BTS* berfungsi sebagai alat *transceiver* radio yang menghubungkan dengan *Mobile Station* menggunakan protokol *radio-link* melalui *Um interface*. *BSC* berfungsi sebagai pemelihara konektivitas antara *BTS* dan *Mobile Station* saat terjadi panggilan dan meneruskannya ke *Network Subsystem(NSS)*.



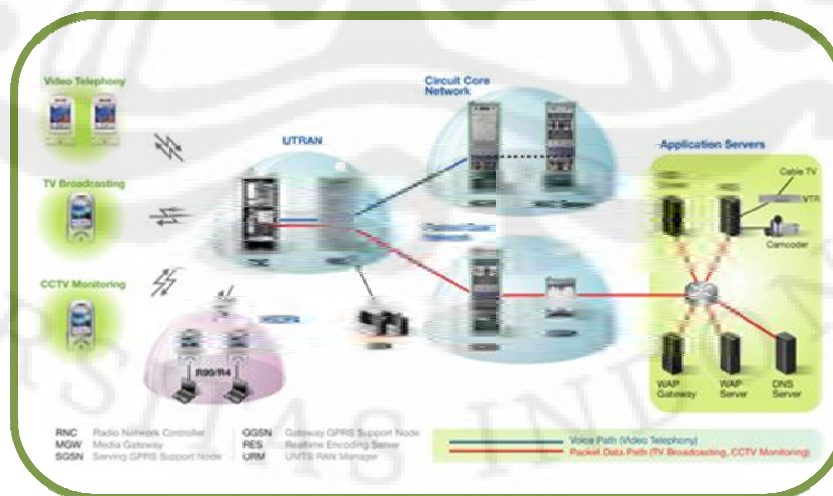
Gambar 3.2 Jaringan GSM untuk GPRS[16]

Network Subsystem terdiri dari *MSC(Mobile Switching Center)*, *HLR(Home Location Register)*, *VLR(Visitor Location Register)*, *AuC(Authentication Center)*, dan *EIR(Equipment Identity Register)*. *MSC* bertanggung jawab melakukan penyambungan dan menentukan proses *routing* kepada setiap pelanggan yang akan melakukan panggilan. *HLR* dan *VLR* adalah tempat menyimpan nomor-nomor pelanggan. *AuC* bertanggung jawab terhadap

proses autentikasi pelanggan pada sistem. EIR berfungsi menyimpan nomor IMEI dari setiap *handset* pelanggan yang diperlukan pada saat autentikasi.

Pada jaringan GSM agar terhubung dengan jaringan internet diperlukan sebuah jaringan GPRS sebagai gerbang komunikasi dengan jaringan yang berbasis IP. GPRS mempunyai dua komponen utama SGSN(*Serving GPRS Support Node*) dan GGSN(*Gateway GPRS Support Node*). SGSN berfungsi mengatur semua aktivitas pelanggan diantaranya proses penarifan dan enkripsi data. Sedangkan GGSN berfungsi sebagai *gateway* antara jaringan internal(GSM) dan jaringan eksternal(internet).

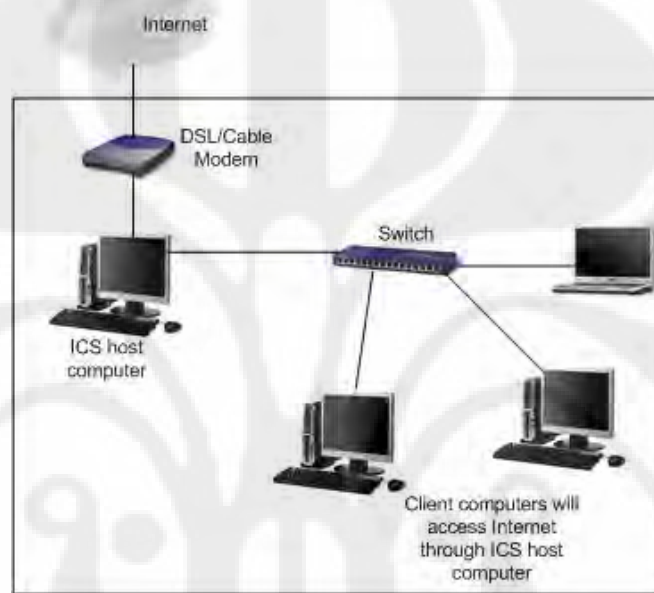
Arsitektur jaringan UMTS seperti pada Gambar 3.3, terdiri atas *Mobile Equipment*, UTRAN(*UMTS Terrestrial Radio Access Network*), *Circuit Network*, dan *Packet Network*. Jaringan UMTS adalah jaringan seluler generasi ketiga(3G). BTS dalam jaringan UMTS dikenal dengan Node B sedangkan BSC dikenal dengan RNC(*Radio Network Controller*). UMTS terdiri atas dua jaringan yaitu *Circuit Network* dan *Packet Network*. *Circuit Network* bertanggung jawab menangani komunikasi suara layaknya pada jaringan GSM. Sedangkan *Packet Network* bertanggung jawab melakukan komunikasi data layaknya GPRS pada GSM. Namun, yang berbeda dari GSM adalah kedua jaringan ini terintegrasi menjadi satu kesatuan. Perbedaan lain, pada jaringan UMTS dapat memberikan layanan data mencapai 2Mbps.



Gambar 3.3 Jaringan UMTS[20]

3.1.2.2 Jaringan Internet

Setelah melewati jaringan telepon seluler, informasi akan masuk jaringan internet. Namun ada beberapa jenis handset dan ponsel tertentu yang dilengkapi dengan akses WiFi atau WiMAX maka informasi akan langsung masuk melalui jaringan internet. Namun, *handset* dan ponsel jenis ini belum banyak beredar dalam masyarakat.



Gambar 3.4 Jaringan Internet[17]

Seperti pada Gambar 3.4, informasi akan langsung masuk ke *server*.

3.1.3 SERVER

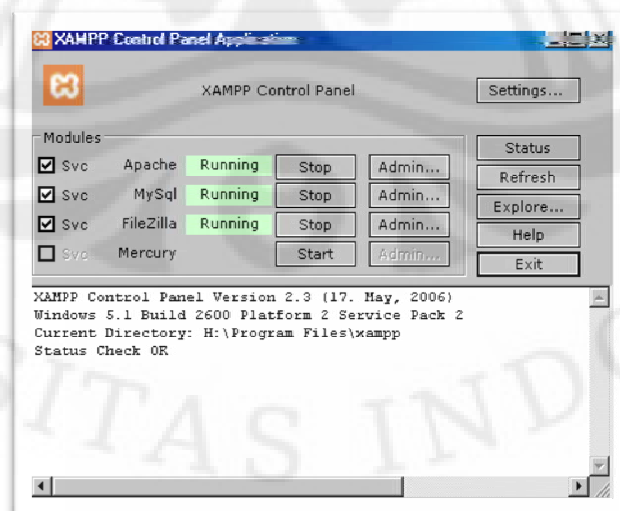
Server pada sistem ini mewakili komponen sebagai pengolah dan penyaji. Dengan bantuan *website* dan *database*, informasi yang diterima, akan diolah, dipisahkan berdasarkan jenisnya, text, gambar, suara, dan atau video. Dalam proses penerimaan disertai pula proses autentikasi pengguna sebagai protokol keamanan. *Database* digunakan sebagai media penyimpanan informasi yang terstruktur dan terorganisir sehingga dalam proses penyajian dalam bentuk halaman-halaman website menjadi lebih mudah.

Komponen server yang digunakan ada dua berupa *hardware* dan *software*. *Hardware*, berupa seperangkat komputer yang terdiri dari CPU, RAM, *Hardisk*. Sedangkan *software*, berupa perangkat lunak server yang disebut XAMPP. Pada Gambar 3.5, contoh sebuah perangkat keras *server*.



Gambar 3.5 Perangkat keras Server[24]

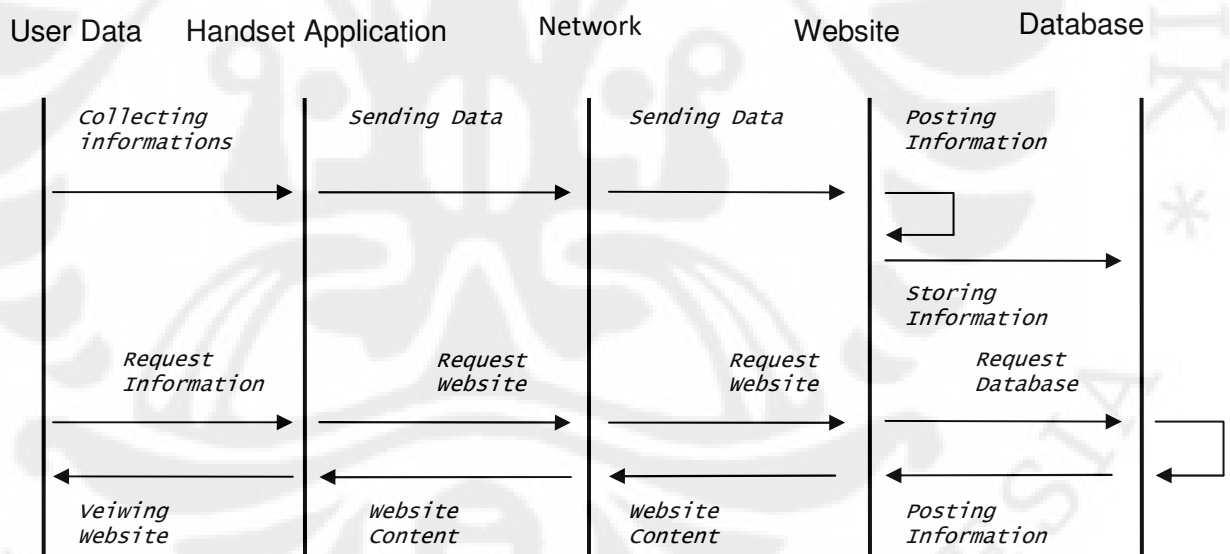
Pada Gambar 3.6 adalah tampilan *User Interface* dari perangkat lunak XAMPP. XAMPP adalah *software server*, maksudnya adalah *software* yang menjalankan eksekusi terhadap setiap *request* yang diminta oleh *client*. XAMPP akan mengeksekusi *website* yang ditulis dengan bahasa pemrograman PHP. Fitur lain dari XAMPP adalah *database* MySQL yang telah terintegrasi. Proses kontrol dapat dilakukan dengan mudah, dengan sebuah panel *Start* dan *Stop*, yang terlihat dalam *interface*-nya pada Gambar 3.6.



Gambar 3.6 User Interface XAMPP server

3.2 DIAGRAM ALIR MOBILE CITIZEN MEDIA

Secara garis besar diagram alir *Mobile Citizen Media* terlihat pada Gambar 3.7. Ada lima elemen penting yang terlibat dalam *Mobile Citizen Media*, yaitu *User Data*, *Handset Application*, *Network*, *Website* dan *Database*. *User data* adalah segala bentuk informasi yang berasal dari pengguna. Informasi ini bisa berupa identitas diri ataupun konten berita yang akan dikirimkan. *Handset Application* adalah sebuah *software* yang dipasang pada *handset* sebagai *tools* untuk mengumpulkan informasi baik berupa tulisan, gambar, suara, dan atau video. *Network* adalah jaringan yang digunakan untuk melewatkan informasi. *Network* terdiri dari dua jenis yaitu *Cellular Network*(jaringan seluler, GSM dan UMTS) dan *Internet Network*(jaringan internet). Elemen selanjutnya adalah *website*, *website* sebagai media pengolah dan penyaji informasi yang dikirim oleh *user*. Elemen terakhir adalah *Database* sebagai media penyimpanan informasi.



Gambar 3.7 Diagram Alir Mobile Citizen Media

Selanjutnya adalah proses kerja *Mobile Citizen Media* :

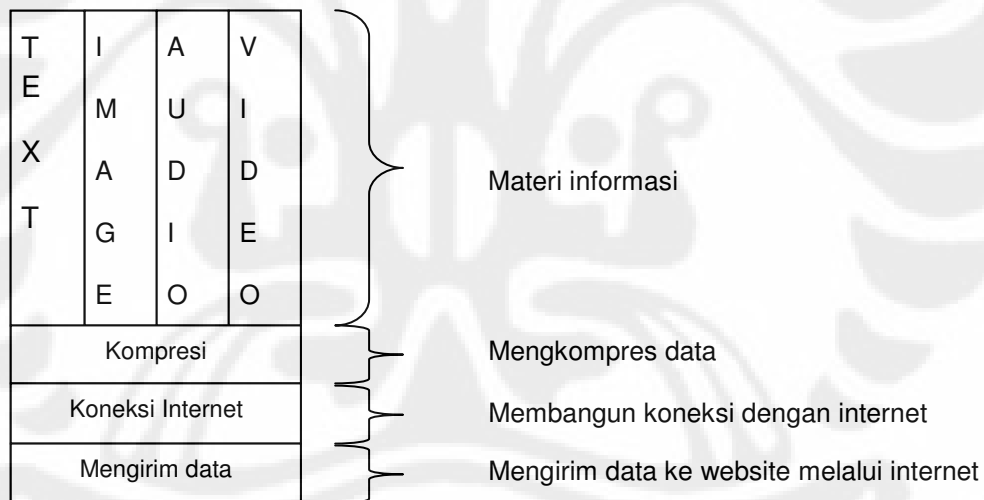
1. *User* mengumpulkan dan memuatkan berbagai jenis informasi baik berupa teks, gambar, suara, dan atau video.

2. Berbagai jenis informasi tersebut di kompres.
3. Selanjutnya data terkompresi tersebut dikirim ke *website* melalui jaringan seluler sebelum melalui jaringan internet.
4. Data yang diterima di *server* akan didekompresi untuk disimpan di dalam *database*.
5. Selanjutnya data tersebut ditampilkan dalam bentuk halaman *website*.
6. Saat *user* ingin melihat berita atau mengecek hasil *posting-an*, *user* dapat menggunakan *internet browser* yang terdapat pada *handset application*.
7. Jika proses *request* berjalan dengan baik, *user* dapat mengakses isi *website* melalui *handset*.

3.3 DIAGRAM ALIR PERANGKAT LUNAK PADA HANDSET

3.3.1 Arsitektur Perangkat Lunak

Pada Gambar 3.8, berbagai bentuk informasi baik berupa teks, gambar, suara, dan atau video, dapat dimuatkan ke dalam *software* ini. Pemuatan bentuk-bentuk informasi di atas sangat tergantung dari jenis dan fitur-fitur telepon seluler yang digunakan. Informasi teks yang dapat dimuatkan dalam *software* ini berjumlah 5120 karakter. Gambar yang dapat dimuatkan berformat JPEG, sedangkan data suara beformat *wav*. Data gambar dapat diambil dari akses kamera pada ponsel secara langsung atau dari file yang disimpan dalam ponsel. *Software* ini juga dapat merekam suara dengan format *wav* selama 20 detik. Kemampuan lainnya adalah merekam video berformat *3gp* dengan durasi tanpa batas (tergantung ketersediaan memori ponsel).

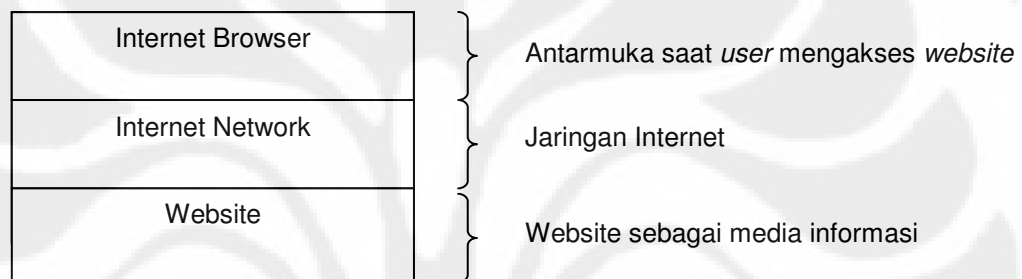


Gambar 3.8 Arsitektur Software pengiriman informasi pada Mobile Citizen Media di telepon seluler

Komponen selanjutnya adalah kompresi. Kompresi menggunakan algoritma LZW, yang telah dijelaskan pada sub bab 2.7. Data yang dikompresi adalah data teks. Adapun data lainnya seperti gambar, suara, dan video tidak dikompresi. Hal ini karena data seperti berformat JPEG dan *3gp* telah mengalami kompresi.

Koneksi internet yang digunakan dalam *software* ini menggunakan protokol *HTTP Connection*. *Method* yang digunakan adalah *GET* dan *POST*. *Method GET* digunakan untuk mengirim data yang berupa teks saja sedangkan data berupa stream seperti data gambar, suara dan video menggunakan *method POST*.

Selanjutnya, untuk mengakses informasi yang berada di *website Mobile Citizen Media*, pada *software Mobile Citizen Media* dilengkapi sebuah *internet browser* sederhana. Arsitektur *software internet browser* adalah seperti pada Gambar 3.9 .



Gambar 3.9 Arsitektur *software* untuk *Internet Browser* pada *Mobile Citizen Media* di telepon seluler

Layaknya *internet browser* pada komputer, *internet browser* ini akan berfungsi menampilkan isi *website*, sehingga masyarakat dapat mengakses informasi dan mengecek apakah informasi yang dikirimnya telah ter-*posting* di dalam *website*.

3.3.2 Diagram Alir Perangkat Lunak

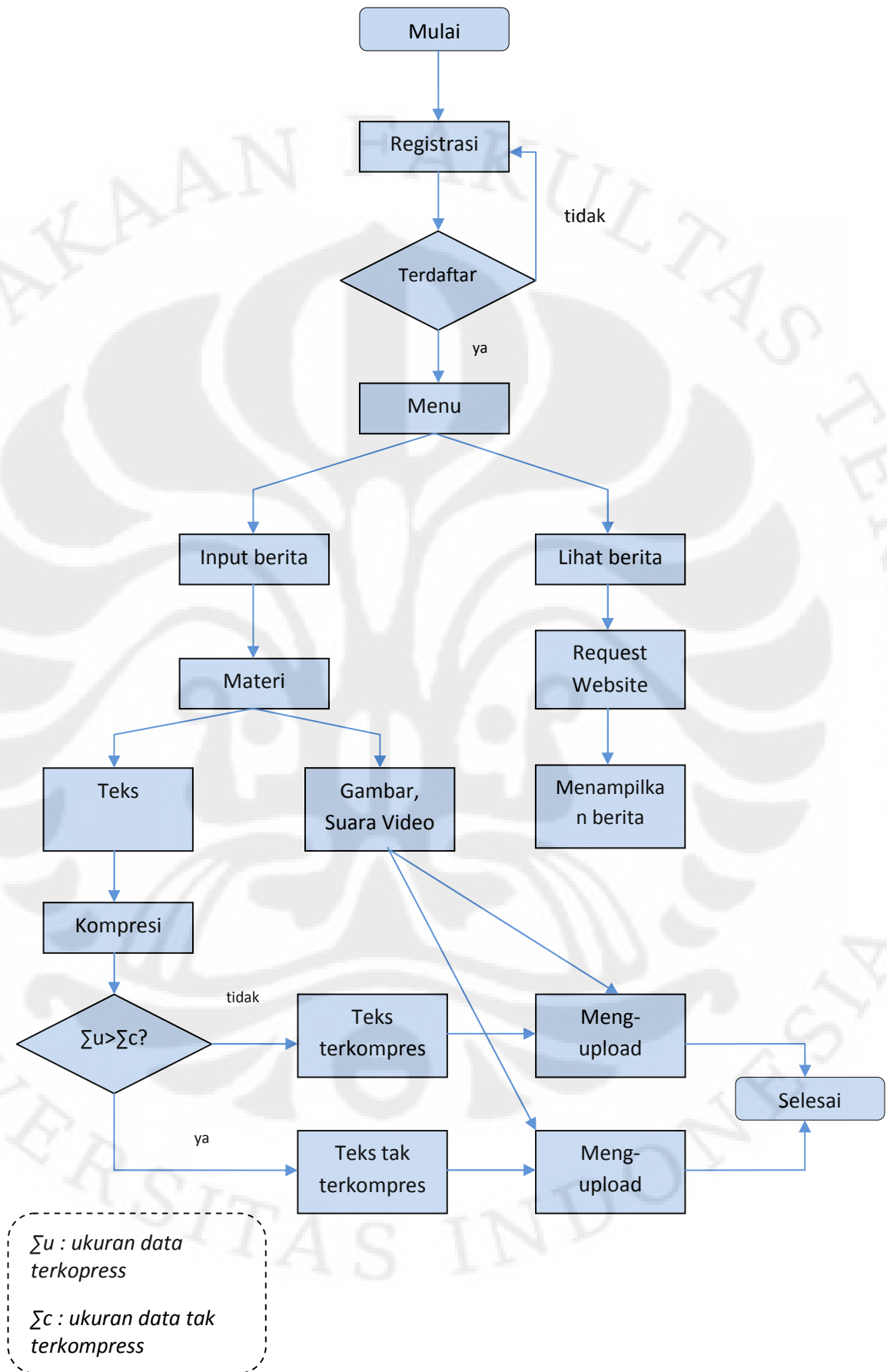
Gambar 3.10 menunjukkan diagram alir perangkat lunak *Mobile Citizen Media* pada ponsel.

Alur kerja perangkat lunak ini adalah sebagai berikut:

1. Pada saat pertama kali menjalankan aplikasi ini, *user* akan diminta untuk registrasi terlebih dahulu. Identitas yang perlu dimasukkan dalam proses registrasi terdiri dari : *username*, *password*, nama lengkap, dan nomor ponsel. Data identitas ini akan dikirim ke *website* dan disimpan dalam memori

ponsel. Sehingga pada prpses pengecekan registrasi selanjutnya hanya merujuk pada informasi yang terdapat pada memori ponsel.

2. Selanjutnya *user* dapat memasukkan materi berita baik berupa teks, gambar, suara, dan atau video dari *handset* mereka. Pada saat yang bersamaan pula, *user* dapat mengakses berita terbaru dari *website Mobile Citizen Media*.
3. Data teks akan dikompresi menggunakan teknik kompresi LZW, jika besar data yang dikompresi lebih kecil maka data tersebut dikirim, sebaliknya jika lebih besar maka data yang tidak terkompresi-lah yang akan dikirim.
4. Data-data tersebut dikirimkan melalui jaringan internet menuju *server*.

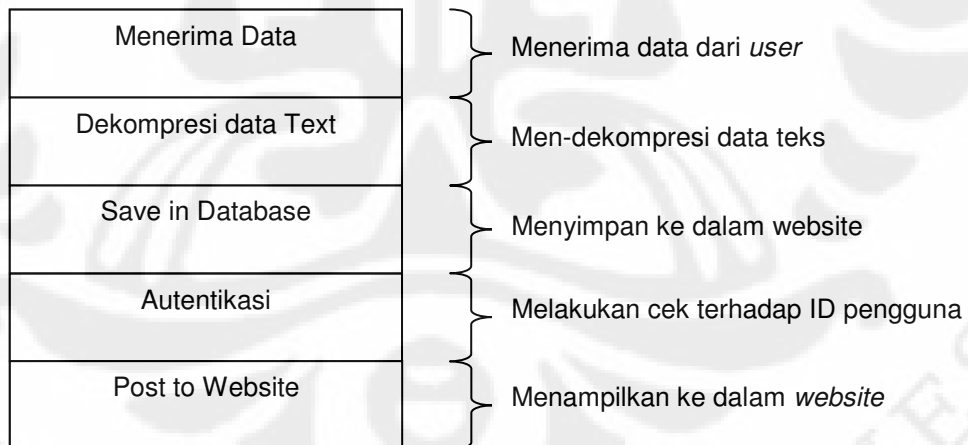


Gambar 3.10 Diagram alir software handset Mobile Citizen Media

3.4 DIAGRAM ALIR WEBSITE

3.4.1 Arsitektur Website

Arsitektur website mempunyai empat proses utama yaitu dekompresi, penyimpanan, autentikasi, dan *posting*. Hal ini diilustrasikan seperti pada Gambar 3.11. Dekompresi dilakukan apabila data yang diterima adalah data terkompresi. Namun, ada kemungkinan data yang diterima tidak terkompresi karena seperti penjelasan pada sub bab 3.5.2, jika ukuran data terkompresi lebih besar dari ukuran aslinya. Sehingga ada kemungkinan data yang diterima tidak melalui proses kompresi. Proses selanjutnya adalah penyimpanan, data disimpan dalam *database* secara terpisah ke dalam masing-masing *record* berdasarkan kategorinya. Proses autentikasi disini berfungsi untuk menjamin bahwa berita yang ditampilkan berasal dari anggota *Mobile Citizen Media* yang jelas. Hal ini berkaitan dengan proses berikutnya yaitu *posting*. *Posting* adalah proses menampilkan berita yang telah terautentikasi.

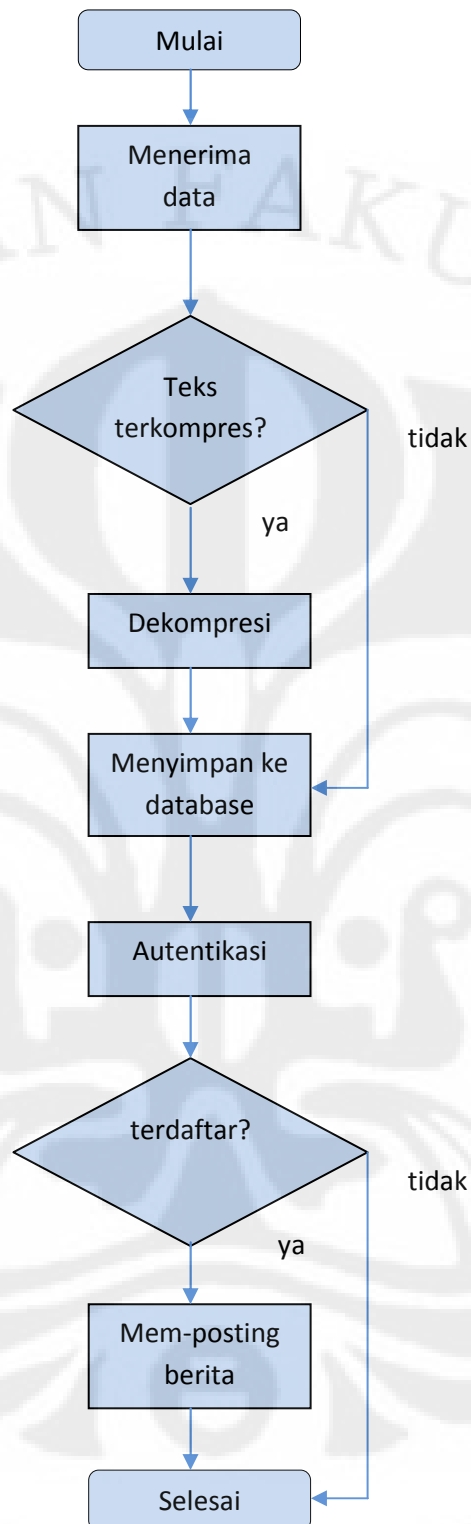


Gambar 3.11 Arsitektur website pada *Mobile Citizen Media*

3.4.2 Diagram Alir Website

Pada Gambar 3.12 ditunjukkan diagram alir *website Mobile Citizen Media*. Berikut ini adalah alur kerja *website* :

1. Pertama, *server* menerima data dari internet yang dikirim oleh para pengguna.
2. Proses selanjutnya adalah mengecek apakah data teks yang diterima dalam bentuk terkompresi, jika ya maka data teks tersebut akan dikompresikan terlebih dahulu. Sedangkan data yang lain seperti data gambar, suara, dan atau video akan langsung disimpan ke dalam database. Untuk membedakan data teks tersebut terkompresi atau tidak, digunakan sebuah penanda dalam data tersebut.
3. Sebaliknya jika data teks tersebut tidak terkompresi maka data tersebut akan langsung disimpan ke dalam *database*.
4. Selanjutnya, sebelum data-data tersebut ditampilkan di *website*, data tersebut akan terlebih dahulu diautentikasi, dicek apakah pengirim telah terdaftar di *database* anggota *Mobile Citizen Media*.
5. Jika ya, maka berita tersebut ditampilkan di *website*.



Gambar 3.12 Diagram alir website pada Mobile Citizen Media

BAB 4

ANALISIS DAN UJICOBA

Ujicoba *Mobile Citizen Media* terdiri tiga jenis, yaitu ujicoba efisiensi kompresi, ujicoba pengunggahan data(*upload*) melalui jaringan GPRS, dan ujicoba pengunggahan data(*upload*) melalui jaringan UMTS/3G. Adapun parameter-paramater yang digunakan adalah besarnya ukuran data dalam satuan *byte* dan waktu pengunggahan (*upload*) dalam satuan detik.

4.1 SKENARIO UJICOBA

Mobile Citizen Media dibangun dari dua sub bagian utama, yaitu pada sisi *server* dan pada sisi *client*(ponsel). Pada sisi *server* menggunakan bantuan berupa perangkat keras(*hardware*) dan perangkat lunak(*software*). Spesifikasi perangkat keras komputer yang digunakan sebagai *server* terlihat dalam Tabel 4.1.

Tabel 4.1 Spesifikasi perangkat keras komputer *server*

NO	Nama	Kapasitas
1	Processor P4	2,8 GHz
2	Memori RAM 512 MHz	512 MHz
3	Harddisk Seagate	80 Gbyte

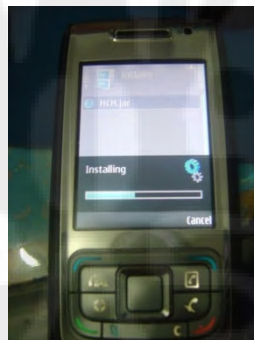
Adapun *software* yang digunakan sebagai *server* menggunakan XAMPP 2.3 dengan sistem *database* MySQL 5.0.21. Selanjutnya pada sisi *client*, digunakan ponsel Nokia E65 yang diatur pada layanan jaringan GSM atau jaringan UMTS saja. Operator seluler yang digunakan dalam ujicoba ini adalah Telkomsel, Indosat, XL, Axis, dan 3. Alasan menggunakan lebih dari satu operator seluler yaitu mengukur tingkat performansi layanan data pada jaringan GSM dan UMTS khususnya proses *upload* dari masing-masing operator.

Pada ujicoba ini, komputer *server* diletakkan di Laboratorium Telekomunikasi, Departemen Teknik Elektro, Universitas Indonesia. Dengan alamat, <http://www.ee.ui.ac.id/telkom/mcm2>.

4.1.1 Instalasi Perangkat Lunak MCM pada Handset

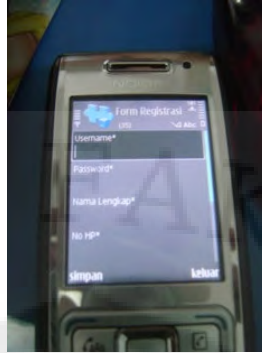
Sebelum melakukan ujicoba, perangkat lunak Mobile Citizen Media pada *handset* harus di-*instal* terlebih dahulu. Langkah-langkah instalasi adalah sebagai berikut:

1. Awalnya *MCM.jar* yang telah dibangun menggunakan Netbeans IDE 6.0 disimpan ke dalam memori ponsel. Ada beberapa tipe merk ponsel yang menyediakan memori eksternal dalam bentuk *microSD*, sebagai media penyimpanan data tambahan. Untuk aplikasi *MCM.jar* ini, penyimpanan bisa ditempatkan pada memori ponsel internal maupun eksternal.
2. Langkah selanjutnya adalah meng-*instal* *MCM.jar* tersebut agar bisa dijalankan pada ponsel. Proses *instalasi software* seperti pada Gambar 4.1.



Gambar 4.1 Proses instalasi software pada ponsel

3. Apabila instalasi berhasil, maka akan muncul tampilan pertama kali seperti pada Gambar 4.2. Tampilan ini digunakan sebagai registrasi pengguna pada *Mobile Citizen Media*. Identitas yang harus dimasukkan meliputi, username, password, nama lengkap, nomor ponsel dan email jika ada. Data identitas ini akan dikirim ke *database server* untuk disimpan, sebagai identitas pengguna *Mobile Citizen Media*. Akhirnya proses instalasi *software* selesai.



Gambar 4.2 Proses instalasi software selesai

4.1.2 Instalasi Website pada Server

Proses selanjutnya adalah *instalasi software* pada *server*. Langkah-langkahnya adalah sebagai berikut:

1. Hal pertama yang harus dilakukan adalah mengunduh software XAMPP 2.3 di www.nat32.com/xampp. Software ini gratis dan dapat didistribusikan secara bebas.
2. Software XAMPP ini dapat di-*instal* pada dua jenis OS (*Operating System*) yang berbeda yaitu berbasis *Linux* dan *Windows*.
3. Apabila proses instalasi selesai maka dapat dicek pada *Internet Browser* dengan memasukkan alamat <http://localhost>, maka akan muncul tampilan seperti Gambar 4.3.



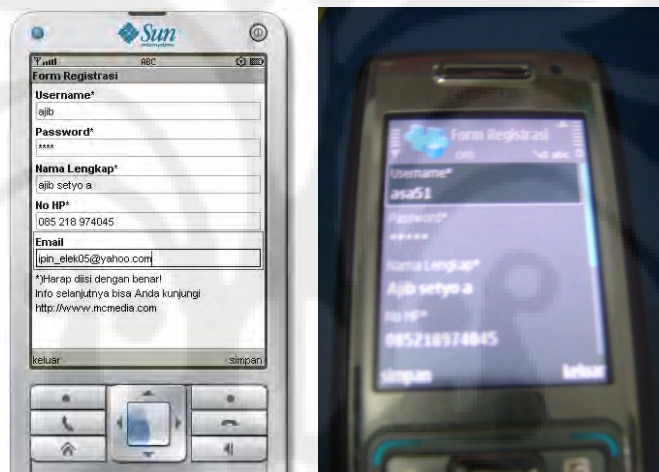
Gambar 4.3 Tampilan XAMPP pada Internet Browser

4. Instalasi XAMPP ini juga sekaligus telah meng-*instal software database MySQL*.

4.1.3 Langkah-langkah menjalankan *MCM* pada *Handset*

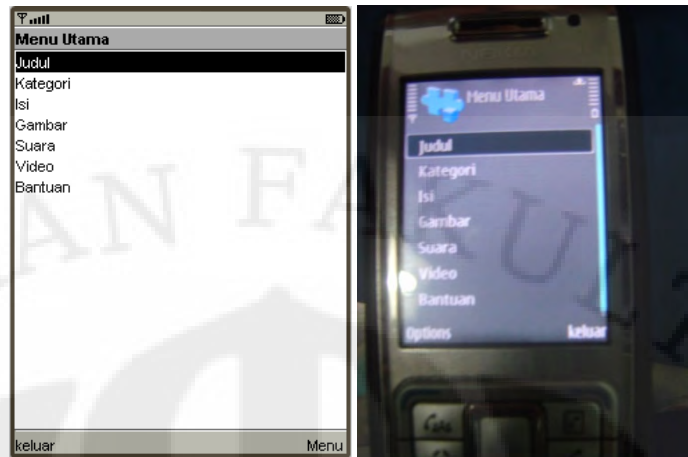
Apabila *software MCM* pada handset telah ter-*instal* maka salah satu proses jurnaslistik, yaitu mengumpulkan berita, dapat dilakukan. Berikut adalah langkah-langkah menjalankan *software MCM* pada handset:

1. Pertama, setelah proses instalasi selesai, *user* diminta meregistrasikan identitas diri untuk disimpan dalam *database server* pengguna. Identitas yang harus deregister adalah *username*, *password*, nama lengkap, nomor ponsel, dan email jika ada. Tampilan *form* registrasi seperti Gambar 4.4,



Gambar 4.4 Tampilan Form Registrasi

2. Apabila registrasi berhasil, maka akan masuk ke menu utama. Pada menu utama seperti pada Gambar 4.5, ada tujuh pilihan yaitu *Judul*, untuk memasukkan judul berita, *Kategori*, untuk memilih kategori berita, *Text*, untuk menuliskan isi berita, *Gambar*, *Suara* dan *Video* masing-masing untuk memasukkan jenis-jenis berita berupa *Gambar*, *Suara* dan atau *Video*. Pada *Gambar*, *user* bisa meng-*insert* gambar, baik dari memori ponsel maupun memotretnya langsung dari kamera ponsel. Sedangkan pada *Suara* dan *Video*, *user* bisa merekamnya melalui ponsel.



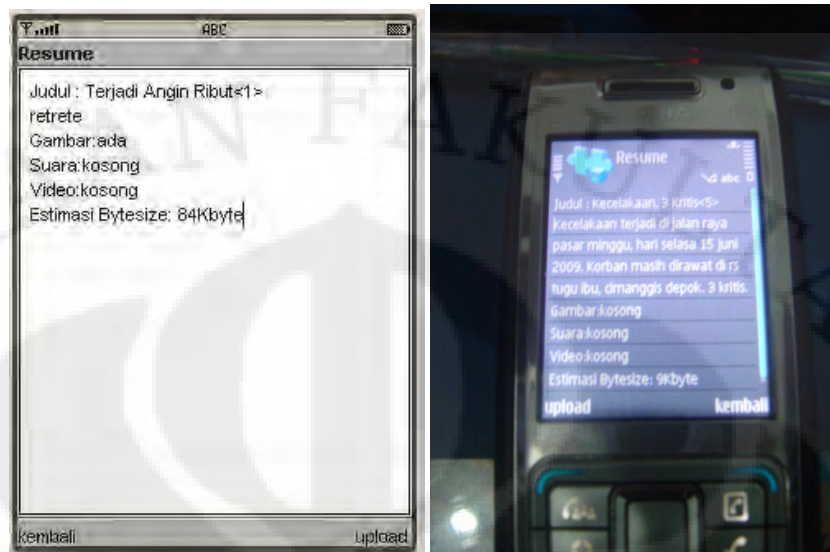
Gambar 4.5 Tampilan Menu Utama

3. Pada perintah *Menu*, seperti pada Gambar 4.6, terdapat beberapa pilihan sub menu yaitu *Open*, untuk membuka menu-menu utama, *Resume*, untuk melihat ringkasan sebelum berita tersebut dikirim, *Pengaturan*, untuk meng-*edit* identitas pengguna, dan *Berita Terkini*, adalah untuk masuk ke mini Browser.



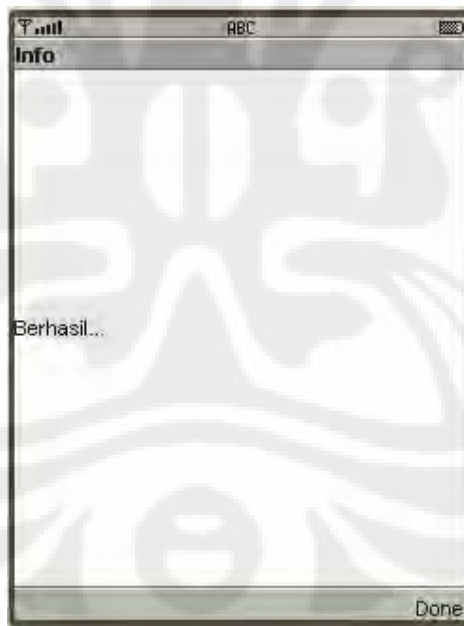
Gambar 4.6 Tampilan pilihan perintah

4. Gambar 4.7 adalah tampilan *Resume*, saat berita siap di upload.



Gambar 4.7 Tampilan saat berita siap diupload

5. Gambar 4.8 adalah tampilan ketika berita berhasil di-upload.



Gambar 4.8 Tampilan saat berita telah terkirim

4.2 UJICоба EFISIENSI KOMPRESI

Pada ujicoba efisiensi kompresi ini dilakukan dengan bervariasi ukuran data yang akan dikompresi. Ukuran data merepresentasikan jumlah karakter yang digunakan, misalnya 118 byte artinya ada 118 karakter/huruf yang dikompresikan. Ukuran data dimulai dari 37 byte hingga 4058 byte. Dengan dilakukan 25 kali pengujian. Karakter yang diambil secara acak, dari berbagai jenis berita.

Dari Tabel 4.2, hasil ujicoba hasil efisiensi menunjukkan nilai efisiensi positif dan negatif. Nilai efisiensi positif artinya ukuran data terkompresi lebih kecil dari pada data aslinya. Sebaliknya nilai efisiensi negatif menandakan ukuran data hasil kompresi lebih besar dari ukuran data aslinya. Hasil ujicoba nilai efisiensi positif terjadi ketika data asli lebih besar sama dengan 460 byte dengan nilai efisiensi sebesar 2.6%. Dan hasil efisiensi data bernilai negatif ketika ukuran data lebih kecil sama dengan 377 byte dengan nilai efisiensi -0,53%.

Berikut ini adalah penjelasan mengapa terjadi efisiensi negatif pada hasil kompresi,

1. Setiap karakter mempunyai panjang 8 bit.

1 huruf/karakter = 8 bit > 1010 0000

2. Masing-masing karakter dikode-kan dengan panjang 12 bit, misal ada dua karakter, maka akan ada dua *codeword* dengan total panjangnya adalah 24 bit.

2 huruf > 1010 0000 0101 1110

Codeword > 0000 1010 0000 0101 1110 0000

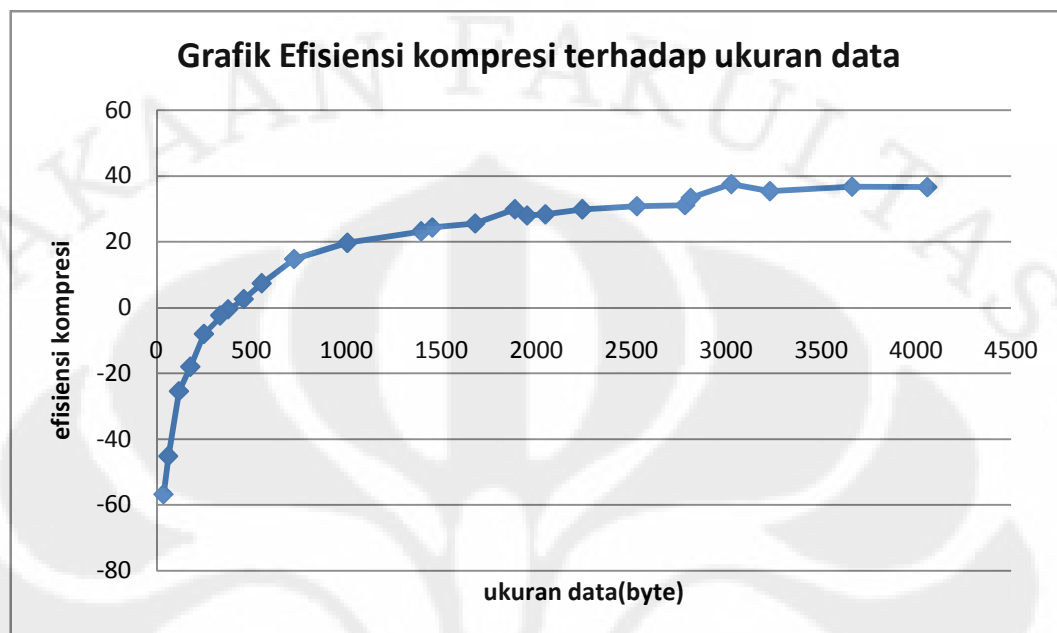
3. Dalam Java Microedition, data yang dikirimkan dibentuk ke dalam satuan byte (1 byte = 8 bit), maka akan didapat 3 karakter baru, yang jumlah totalnya 24 bit (3 byte). Maka dari hal ini terjadi pembengkakan jumlah bit dari 16 bit (2 byte) menjadi 24 bit (3 byte).

Hasil pengujian kompresi

Tabel 4.2 Data hasil efisensi kompresi

No	Ukuran data asli(byte)	Ukuran data terkompresi(byte)	Efisensi(%)
1	37	58	-56.7567568
2	62	90	-45.1612903
3	118	148	-25.4237288
4	178	210	-17.9775281
5	250	270	-8
6	335	343	-2.3880597
7	377	379	-0.53050398
8	460	448	2.608695652
9	554	513	7.400722022
10	725	618	14.75862069
11	1005	807	19.70149254
12	1394	1071	23.17073171
13	1452	1098	24.38016529
14	1678	1249	25.56615018
15	1887	1323	29.88871224
16	1951	1405	27.98564839
17	2048	1467	28.36914063
18	2242	1572	29.88403211
19	2529	1749	30.84223013
20	2783	1917	31.1174991
21	2813	1875	33.34518308
22	3027	1891	37.52890651
23	3230	2086	35.41795666
24	3662	2316	36.75587111
25	4058	2572	36.61902415

Grafik hasil pengujian kompresi nilai efisiensi terhadap ukuran data asli dapat dilihat pada Gambar 4.9.



Gambar 4.9 Grafik hasil efisiensi kompresi terhadap ukuran data asli

Dari Gambar 4.9 terjadi saturasi efisiensi pada nilai lebih kurang 36%, dengan titik saturasi dimulai dengan ukuran data sebesar 3027 byte. Didapatkan nilai efisiensi yang minus, untuk data kurang sama dengan 377 byte. Nilai minus ini berarti, bahwa besar data terkompres lebih besar dari data asli.

Dengan adanya kemungkinan terjadi efisiensi negatif pada proses kompresi, untuk menanggulangi masalah ini maka ditambahkan *Smart Decision Algorithm*. Algoritma ini akan mengecek apakah hasil kompresi lebih besar atau lebih kecil. Jika lebih besar, artinya mempunyai nilai efisiensi negatif, maka data yang dikirimkan adalah data asli. Dengan demikian, data yang dikirimkan adalah data dengan ukuran yang paling minimal.

Algoritma ini dipasang pada sisi *client* dan *server*. Pada sisi *client* berfungsi memutuskan data mana yang akan dikirimkan, sedangkan pada *server* akan memutuskan apakah data yang diterima merupakan data terkompresi atau tidak. Pada *server* digunakan sebuah *flag*/penanda yang unik sehingga data

tersebut dapat dikenali dan dapat diputuskan apakah data teks tersebut perlu didekompresi atau tidak.

4.3 UJICоба PADA JARINGAN GSM

Ujicoba menggunakan jaringan GSM dilakukan menggunakan sebuah ponsel Nokia E65. Pengambilan data ini dilakukan di daerah Jagakarsa, Jakarta Selatan, pada tanggal 13 Juni 2009. Maksud pengujian menggunakan satu buah ponsel adalah agar kecepatan pemrosesan data seragam sehingga dapat fokus pada waktu pengunggahan(*upload*).

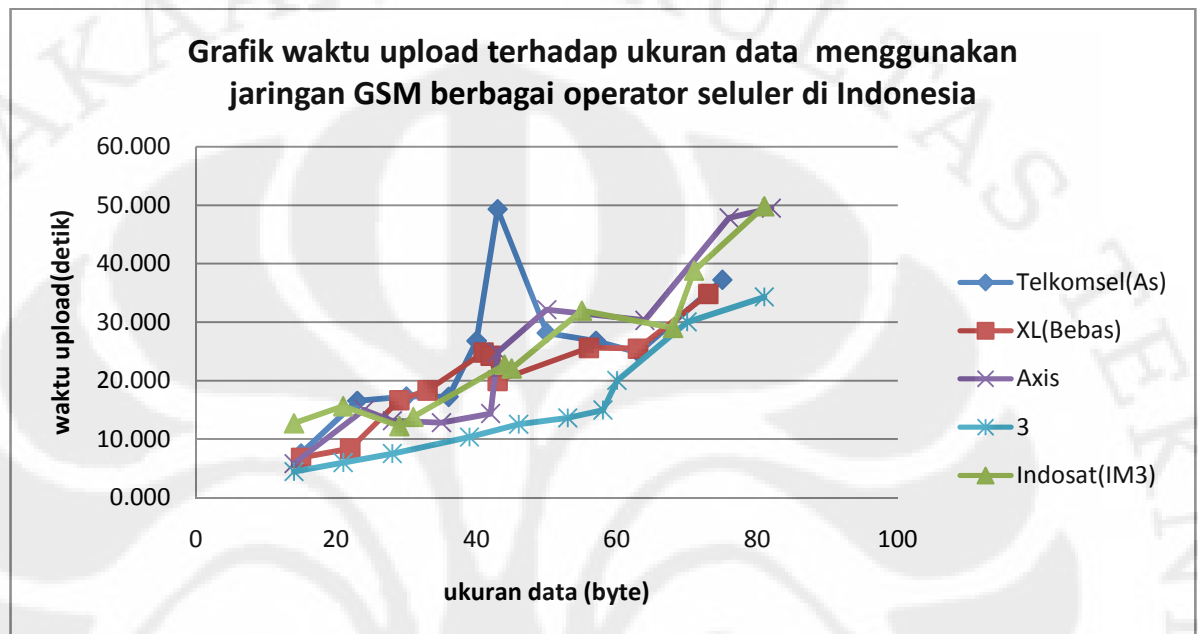
Ukuran data yang diujicobakan divariasikan dari 14 Kbyte hingga 81 Kbyte. Setiap ukuran data dilakukan pengambilan sampel sebanyak 3 kali. Variasi lain adalah data-data ini tidak hanya diambil dari satu operator seluler, tetapi ada 5 operator, yaitu Telkomsel, XL, Axis, 3 dan Indosat.

Hasil pengujian pada jaringan GSM

Tabel 4.2 Data pengujian pada jaringan GSM

No	Telkomsel		xl		axis		3		Indosat(IM3)	
	Ukuran (Kbyte)	Waktu (detik)	Ukuran (Kbyte)	Waktu (detik)	Ukuran (Kbyte)	Waktu (detik)	Ukuran (Kbyte)	Waktu (detik)	Ukuran (Kbyte)	Waktu (detik)
1	15	7.519	15	6.806	14	5.798	14	4.402	14	12.68
2	23	16.501	22	8.353	24	15.001	21	5.958	21	15.62
3	30	17.251	29	16.639	28	13.141	28	7.484	29	12.21
4	36	17.194	33	18.317	35	12.780	39	10.321	31	13.78
5	40	26.776	41	24.784	42	14.360	46	12.503	44	22.72
6	43	49.326	42	24.240	43	24.843	53	13.601	45	22.08
7	50	28.141	43	19.950	50	32.107	58	14.928	55	31.97
8	57	26.777	56	25.593	64	30.290	60	19.984	68	29.02
9	63	24.787	63	25.471	76	47.798	70	30.032	71	38.84
10	75	37.235	73	34.805	82	49.460	81	34.287	81	49.82

Grafik hasil pengujian pada jaringan GSM



Gambar 4.10 Grafik waktu pengunggahan terhadap variasi ukuran data dari berbagai operator seluler GSM di Indonesia

Dari gambar grafik pada Gambar 4.10, terlihat bahwa waktu *upload* yang paling cepat adalah pada operator 3. Hal ini ditandai dengan grafik yang berwarna biru muda terletak paling bawah di antara grafik lain.

Operator 3 adalah pendatang baru di dunia bisnis seluler di Indonesia, operator ini memiliki jumlah pelanggan 3,2 juta, yang jauh lebih sedikit jika dibandingkan dengan Telkomsel yang punya 50 juta pelanggan[21][22]. Oleh karena itu, waktu pendudukan(*holding time*) pada jaringan 3 lebih kecil sehingga mengakibatkan banyak slot yang kosong yang dapat dimaksimalkan untuk komunikasi data. Slot yang kosong akan meningkatkan bandwidth komunikasi, bandwidth yang besar akan memperbanyak bit yang dilewatkan setiap detiknya, sehingga data cepat terkirim. Sebaliknya dengan Telkomsel, yang mempunyai pelanggan banyak, menyebabkan waktu

pendudukan semakin besar sehingga slot yang kosong semakin sedikit, mengakibatkan bandwidth yang sempit sehingga bit yang dilewatkan semakin sedikit, menyebabkan waktu yang diperlukan untuk mengirim data makin lama.

4.4 UJICOBA PADA JARINGAN UMTS

Sama halnya ujicoba pada GSM, ujicoba pada jaringan UMTS juga bervariasi ukuran data yang di-*upload* dari 14Kbyte hingga 80 Kbyte. Ujicoba ini juga menggunakan ponsel Nokia E65 yang diatur agar beroperasi pada jaringan 3G saja.

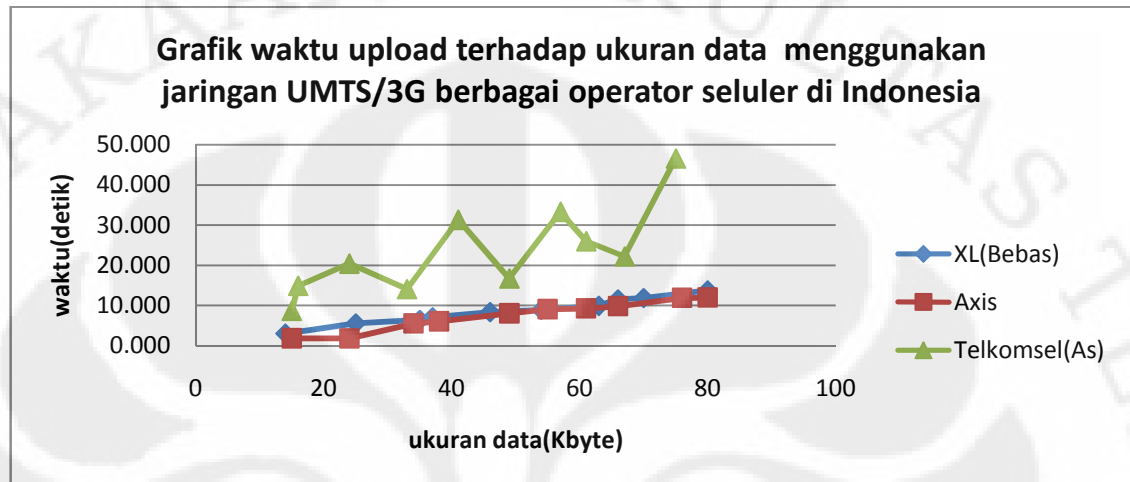
Sedikit perbedaan disini, bahwa hanya 3 oprator seluler yang diujicobakan yaitu XL, Axis, dan Telkomsel. Dua operator lain, yaitu 3 dan Indosat tidak dapat diambil datanya karena, untuk operator 3, tempat dilakukan pengambilan data tidak terjangkau layanan 3G, dan untuk operator Indosat, layanan 3G hanya dikhususkan kepada pengguna kartu IM2, sedangkan ujicoba dilakukan menggunakan kartu IM3.

Hasil pengujian pada jaringan UMTS

Tabel 4.3 Data pengujian pada jaringan UMTS

No	xl		Axis		Telkomsel	
	Ukuran (Kbyte)	Waktu (detik)	Ukuran (Kbyte)	Waktu (detik)	Ukuran (Kbyte)	Waktu (detik)
1	14	3.030	15	1.868	15	8.718
2	25	5.565	24	1.850	16	14.926
3	35	6.421	34	5.547	24	20.450
4	37	7.065	38	6.081	33	14.097
5	46	8.340	49	8.058	41	31.340
6	54	8.897	55	9.109	49	16.739
7	63	9.919	61	9.286	57	33.313
8	66	11.499	66	9.854	61	26.018
9	70	11.860	76	11.903	67	22.204
10	80	13.756	80	11.982	75	46.556

Grafik hasil pengujian pada jaringan UMTS



Gambar 4.11 Grafik waktu pengunggahan terhadap variasi ukuran data dari berbagai operator seluler UMTS di Indonesia

Dari grafik pada Gambar 4.12, terlihat bahwa waktu upload yang relatif sama antara operator Axis dan XL serta lebih cepat dari operator Telkomsel. Terjadi fluktuasi yang sangat besar terjadi pada operator Telkomsel. Hal ini disebabkan oleh dua faktor yang saling terkait yaitu jumlah pelanggan dan jangkauan jaringan 3G yang bersifat dinamis. Pada operator Axis yang memiliki jumlah pelanggan 500.000[23] dan XL sebanyak 21,3 juta[15] karena jumlah pelanggannya relatif lebih sedikit dari Telkomsel, maka layanan 3G cukup melayani pelanggan-pelanggannya sehingga daerah layanannya cukup stabil. Berbeda dengan Telkomsel, dengan jumlah pelanggan yang banyak dengan sebaran yang acak maka jangkauan daerah layanan 3G akan berubah secara dinamis. Hal ini sesuai dengan karakteristik dengan jangkauan 3G, bahwa semakin banyak pelanggan suatu lokasi maka akan menurunkan kualitas layanan dan mempersempit jangkauan/radius. Pelanggan yang terletak pada radius yang jauh akan ditandai dengan fluktuasi naik dan turunya energi sinyal. Pada saat

sinyal naik kecepatan pengiriman data dapat dilakukan relatif lebih singkat dibanding dengan saat energi sinyal kecil.

4.5 PERBANDINGAN WAKTU UPLOAD PADA JARINGAN GSM DAN UMTS

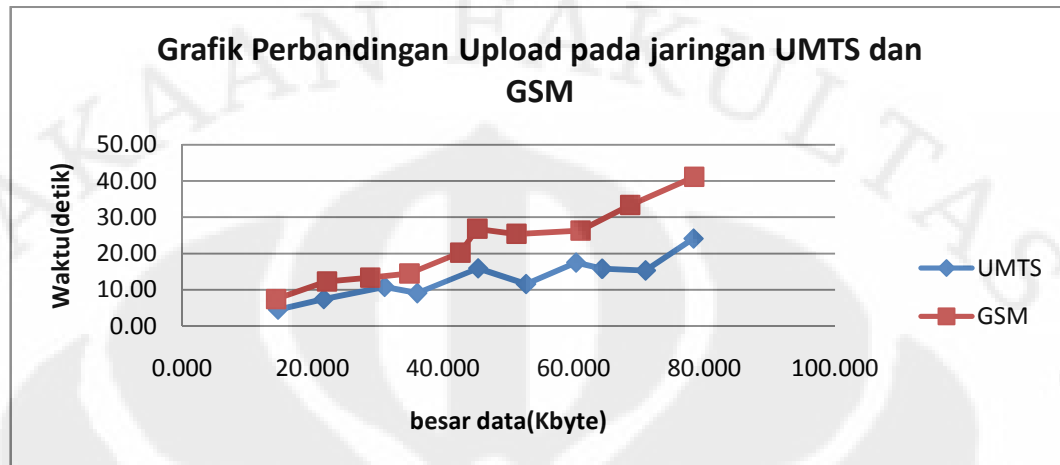
Data-data pada Tabel 4.5 adalah hasil rata-rata dari 5 operator yang diujicobakan pada jaringan GSM dan 3 operator pada jaringan UMTS. Perbandingan waktu upload dapat terlihat pada grafik yang terdapat di Gambar 4.12.

Data

Tabel 4.4 Data perbandingan waktu upload pada jaringan UMTS dan GSM

UMTS		GSM		
Ukuran(Kbyte)	waktu(detik)	Ukuran(Kbyte)	waktu(detik)	
14.66	4.54	14.4	7.44	
21.66	7.45	22.2	12.29	
31.00	10.81	28.8	13.35	
36.00	9.08	34.8	14.48	
45.33	15.91	42.6	20.23	
52.66	11.58	45.2	26.82	
60.33	17.51	51.2	25.42	
64.33	15.79	61	26.33	
71.00	15.32	68.6	33.39	
78.33	24.10	78.4	41.12	
47.53	24.1	44.72	41.12	Rata-rata
	0.277909		0.493895	Rata rata waktu(s/Kbyte)

Grafik



Gambar 4.12 Grafik waktu pengunggahan terhadap variasi ukuran data dari berbagai operator seluler UMTS di Indonesia

Dari Tabel 4.4 didapatkan data hasil rata-rata waktu upload pada jaringan GSM operator-operator seluler di Indonesia dan data hasil rata-rata waktu upload pada jaringan UMTS. Dari Gambar 4.12, didapatkan bahwa rata-rata waktu upload pada jaringan UMTS relative lebih cepat dibandingkan dengan jaringan GSM. Misalkan untuk besar data 14 Kbyte, pada jaringan GSM diperlukan waktu 7,44 detik, sedangkan pada jaringan UMTS diperlukan waktu 4.54 detik. Begitu pula dengan besar data 78,33 Kbyte, pada jaringan UMTS diperlukan 24,10 detik sedangkan pada jaringan GSM diperlukan waktu 41,12 detik. Dari data di atas, terlihat bahwa waktu upload pada jaringan GSM hampir dua kali lipat waktu yang diperlukan pada jaringan UMTS.

Pengiriman data melalui jaringan GSM lebih lama dibandingkan dengan pengiriman melalui jaringan UMTS hal ini disebabkan oleh bandwidth yang berbeda pada kedua jaringan tersebut. Pada UMTS bandwidth yang ditawarkan bisa mencapai 2 Mbps sedangkan pada GSM dengan GPRS-nya sebesar 115 kbps, sehingga akan lebih banyak data yang bisa dilewatkan melalui jaringan UMTS.

4.6 UJICOBA FITUR TAMBAHAN

4.6.1 Mini Browser

Pada *software* MCM yang dipasang pada handset, terdapat fitur tambahan yang berfungsi untuk melihat berita terbaru dari website MCM. Browser ini akan menampilkan lima berita terbaru pada *website* MCM.

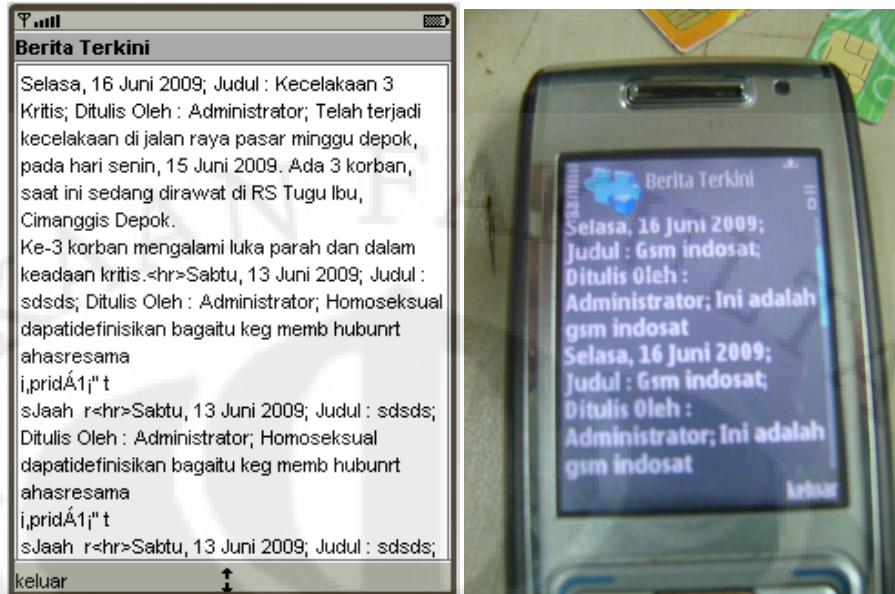
Langkah-langkah menjalankan browser ini sebagai berikut:

1. Pada perintah menu terdapat pilihan *Berita terkini*, seperti pada Gambar 4.13.



Gambar 4.13 Tampilan saat mengakses mini Browser

2. Selanjutnya setelah proses *downloading* selesai, maka akan muncul tampilan berita yang telah diunduh. Terlihat pada Gambar 4.14.



Gambar 4.14 Tampilan berita yang diunduh

Dalam mini browser ini hanya mampu menampilkan berita dalam bentuk teks. Berita yang ditampilkan adalah lima berita terbaru yang sedang ditampilkan pada *website*. Informasi yang ditampilkan meliputi hari dan tanggal, judul berita, penulis, dan Isi berita. Biasanya kelima berita terbaru tidak bisa termuat dalam satu halaman display ponsel, maka untuk bisa melihat seluruhnya maka *user* harus men-*scroll* layar ke arah atas atau bawah .

4.6.2 Biaya

Pada ujicoba ini, tidak hanya dilihat dari aspek teknis, tetapi juga dari aspek ekonomis, dengan maksud memberikan estimasi biaya dalam rangka pertimbangan implementasi MCM dalam masyarakat. Berdasarkan jenis berita yang dikirim akan merepresentasikan ukuran data yang akan dikirim. Berdasarkan tarif operator seluler untuk akses internet yang paling mahal dan yang paling murah dihitung per *Kbyte*-nya. Berikut ini Tabel 4.5 ,menyajikan data tersebut,

Tabel 4.5 Data perbandingan biaya dari tarif paling mahal dan paling murah

No	Jenis berita	Ukuran data(Kbyte)	Paling mahal(Rp 12/Kbyte)	Paling Murah(Rp 0.01/Kbyte)
1	teks	4	48	0.04
2	teks dan gambar	59	708	0.59
3	teks dan suara(10 detik)	163	1956	1.63
4	teks dan video(10 detik)	132	1584	1.32
5	teks, gambar, suara	199	2388	1.99
6	teks, gambar,video	234	2808	2.34
7	teks, gambar, suara dan video	438	5256	4.38

Dari Tabel 4.5, dapat terlihat bahwa untuk mengirimkan berita dalam bentuk teks saja, dengan ukuran data 4 Kbyte, dengan tarif paling murah adalah sebesar Rp 0,04 dan tarif paling mahal adalah Rp 48,00. Sedangkan untuk mengirimkan berita multimedia dengan memasukkan teks, gambar, suara, dan video dengan besar 438 Kbyte dengan tarif paling mahal adalah sebesar Rp 5.256,00 dan dengan tarif paling murah adalah Rp 4.38,00.

BAB 5

KESIMPULAN

1. Rancangan perangkat lunak *Mobile Citizen Media* telah memenuhi kriteria sesuai dengan pengertian *Mobile Citizen Media*.
2. Rancangan perangkat lunak *Mobile Citizen Media* dapat berjalan baik dan dapat diimplementasikan.
3. Data teks akan dikompresi jika besarnya lebih besar sama dengan 460 *byte*.
4. Efisiensi kompresi maksimum dengan algoritma LZW mempunyai nilai 36%.
5. Titik saturasi efisiensi terjadi pada data ukuran lebih besar sama dengan 3027 *byte*.
6. Waktu pengunggahan (upload) rata-rata menggunakan jaringan GSM sebesar 0.49 detik/Kbyte , lebih lama dibandingkan menggunakan jaringan UMTS sebesar 0.28 detik/Kbyte .

Daftar Referensi

- [1] Baase, S, "A Gift of Fire". Prentice Hall, 3rd Edition, 2008.
- [2] "____", "Citizen Media", 2008, diakses dari :
http://en.wikipedia.org/wiki/Citizen_media
- [3] "____", "Mobile Media Content & Commerce Distribution : Citizen Correspondent", 2008 diakses dari :
<http://lrolando.blogspot.com/2008/07/citizen-correspondents.html>
- [4] Ohmynews, "OhmyNews Citizen Journalism School Opens", diakses dari
http://english.ohmynews.com/articleview/article_view.asp?
- [5] Wikinvest, "Mobile Phone Adoption in Developing Countries", 2008, diakses dari :
http://www.wikinvest.com/concept/Mobile_Phone_Adoption_in_Developing_Countries
- [6] Glaser, M., "Stanford Fellow Imagines Every Cell Phone as Citizen Media Outlet", 2006, diakses dari :
http://www.pbs.org/mediashift/2006/07/digging_deeperstanford_fellow.html
- [7] Euromonitor, "The Global Digital Divide", 2007, diakses dari :
http://www.euromonitor.com/The_global_digital_divide
- [8] Apricot 2007, "Telkom Dukung Dan Berperan Aktif Dalam Melaksanakan Cetak Biru ICT Indonesia", 2007, diakses dari : <http://telkom.co.id/pok-media/siaran-pers/apricot-2007>.
- [9] MIT, "Why Africa?", 2008, diakses dari :
<http://eprom.mit.edu/whyafrika.html>

- [10] “_____”, “*Overview of Java 2 Platform, Micro Edition (J2ME™)*”, 2003, diakses dari:
http://www.informit.com/content/images/0321197984/samplechapter/riggsc_h02.pdf
- [11] Ortiz, C. Enrique, “*Managing the MIDlet Life-Cycle with a Finite State Machine*”, 2004, diakses dari:
http://developers.sun.com/mobility/midp/articles/fsm/fsm_fig3.gif
- [12] Mardalain, M.Ginta. “*GYSURV : APLIKASI MOBILE SURVEILANCE SYSTEM BERBASISKAN TEKNOLOGI GSM DAN UMTS*”, Skripsi, 2007
- [13] Mahmoud , Qusay H. “*The J2ME Mobile Media API*”, 2003 diakses dari :
<http://developers.sun.com/mobility/midp/articles/mmapioverview/fig1.gif>
- [14] Vikram Goyal, *Pro Java ME MMAPI Mobile Media API for Java Micro Edition*(New York: Apress, 2006), hal 4
- [15] “_____”, “*XL Pusatkan Perhatian pada Peningkatan Jumlah Pelanggan yang Berkualitas*”, 2009, diakses dari :
<http://www.xl.co.id/tabid/274/newsId/10368/Default.aspx>
- [16] “_____”, “*ZTE’s GSM Solution in action*”, diakses dari :
<http://wwwen.zte.com.cn/main/newspic/20067247303920.jpg>
- [17] “_____”, diakses dari :<http://www.home-network-help.com/images/ics-network.jpg>
- [18] Arifin, Ajib Setyo. “*FREE MOBILE JOURNALISM*”, Seminar, 2008
- [19] Fitriansah, Ahmad. “*ANALISIS PERBANDINGAN KINERJA ALGORITMA KOMPRESI LZW, HUFFMAN DAN DEFLATE PADA BERBAGAI JENIS FILE*”, 2007, diakses dari :

http://www.itelkom.ac.id/library/index.php?option=com_content&view=article&id=214:algoritma-lzw-cont&catid=20:informatika&Itemid=15

- [20] Sing li and Jonathan Knodsen, “*Beginning J2ME: From Novice to Profesional, Third Edition*”(New York:Apress, 2005), hal 118 dan 167
- [21] “_____”, ” *Telkomsel Incar Kenaikan Pelanggan 10 Juta Orang*”, 2008, diakses dari:
<http://www.tempointeraktif.com/hg/ekbis/2008/02/21/brk,20080221-117875.id.html>
- [22] “_____”, ” *Genjot Pelanggan, Operator Seluler Gelar (Lagi) Kiat Bundling*”, 2008, diakses dari : <http://www.tabloid-ponsel.com/news-rss/42-news-topnews/265-genjot-pelanggan-operator-seluler-gelar-lagi-kiat-bundling>
- [23] Candraruna, Muhammad , “*Target Axis 2008, 3700 BTS 2 Juta Pelanggan*”, 2008, diakses dari :
<http://techno.okezone.com/index.php/ReadStory/2008/07/16/54/128225/target-axis-2008-3700-bts-2-juta-pelanggan>
- [24] Puspita, Intan, “*Jaringan*”, 2009,diakses dari:
<http://andrijaringankomputer.blogspot.com/2009/05/jaringan.html>
- [25] “___”, “*Pengantar Ilmu Jurnalsitik*”, diakses dari :
<http://jurnalistikuinsgd.wordpress.com/2007/04/26/pengantar-ilmu-jurnalistik/>

Lampiran




```

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package midletMCM;

import java.io.ByteArrayInputStream;
import java.io.ByteArrayOutputStream;
//import java.io.DataInputStream;
//import java.io.DataOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import javax.microedition.io.Connector;
import javax.microedition.io.file.FileConnection;
import javax.microedition.media.Player;
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;
import javax.microedition.media.Manager;
import javax.microedition.media.MediaException;
import javax.microedition.media.control.RecordControl;
import javax.microedition.media.control.VideoControl;
import org.netbeans.microedition.lcdui.pda.FileBrowser;
//import com.jcraft.jzlib.*;
//import record2.RecordForm;
//import org.netbeans.microedition.lcdui.

/**
 * @author ajib
 */
public class MCMMidlet extends MIDlet implements CommandListener{
    private static final String[] MENU_ITEMS =
    {"Judul", "Kategori", "Isi", "Gambar", "Suara", "Video", "Bantuan"},
    MENU_GAMBAR={"Ambil File", "Ambil dari kamera"},
    MENU_SUARA={"Rekam Suara", "Play"},
    MENU_FILM={"Rekam Video", "Play"}, MENU_KATEGORI=
    {"Politik", "Ekonomi", "Teknologi", "Olahraga", "Kriminal",
    "Kesehatan", "Hiburan"};

    private Display display;
    private List listMain, listGambar, listSuara, listFilm, listKtgr;
    private Command
    cmExit, cmBack, cmSelect, cmSubmit, cmNext, cmUpload, cmCapture, cmCheck, cmR
    ecMovie,
        cmStop, cmRec, cmSet, cmSimpan, cmLihat;
    private FormLogin fmLogin;
    private FormSetting fmSetting;
    private FormHelp fmHelp;
    private TextBox mTextBox, mResume;
    private FormJudul fmJudul;
    private TextField durasiRekam;

    private Form mForm_2, fmRecAudio, fmRecMovie, fmDurasirekam;
    private TextBox textBox;
    private FileBrowser gambarBrowser, videoBrowser, audioBrowser;
    //private Alert alert;
    String Judul=null, Text1, resume, fileGambar, fileVideo,
fileAudio, Text_asli;
    private Player mPlayer, mPlayer2, mPlayer3;
    private VideoControl mVideoControl;
    private RecordControl rc, rc2;
    private Image gambar;
    private ByteArrayOutputStream rawVideo, rawAudio;

    byte[] data, data2, Text;
    private char[] FileGambar;

```

```

String datatext,kategori=null;
//private Image gambar01;
private Image dataFilegambar;
private byte[] byteGambar, byteVideo, byteAudio;
private Image dataCapturegambar;

private StringItem messageItem;
private int p=1,q=1,r=1,s=1,lamaRekam=10;

private Login login;
private DateField dfTime;
public long waktu;
public FormRespon fmRespon;
public Gauge gIn;

public MCMMidlet()
{
    super();
    listMain = new List("Menu Utama",List.IMPLICIT,
MENU_ITEMS,null);
    //deklarasi command
    cmSelect = new Command("Open", Command.OK,0);
    cmBack = new Command("kembali", Command.CANCEL,1);
    cmExit = new Command("keluar", Command.EXIT,2);
    cmSubmit= new Command("tambah", Command.OK,3);
    cmNext = new Command("Resume",Command.OK,4);
    cmUpload= new Command("upload",Command.OK,5);
    cmCapture = new Command("foto",Command.OK,6);
    cmRecMovie= new Command("rekam", Command.OK,8);
    cmCheck=new Command("cekdoang",Command.OK,7);
    cmStop=new Command("selesai",Command.OK,8);
    cmRec=new Command("rekam",Command.OK,9);
    cmSet= new Command("Pengaturan",Command.OK,10);
    cmSimpan= new Command("Simpan",Command.OK,11);
    cmLihat= new Command("Berita baru",Command.OK,12);
    listMain.addCommand(cmExit);
    listMain.addCommand(cmSelect);
    listMain.addCommand(cmNext);
    listMain.addCommand(cmSet);
    listMain.addCommand(cmLihat);
    //buat cek doang
    //listMain.addCommand(cmCheck);

    listMain.setCommandListener(this);

    display = Display.getDisplay(this);

    fmLogin = new FormLogin(this, listMain);

    fmHelp = new FormHelp(this, listMain);

    fmJudul= new FormJudul(this, listMain);
    fmSetting = new FormSetting(this, listMain);
    fmRespon = new FormRespon(this, listMain);

    messageItem = new StringItem("Record", "Click record to start
recording.");

    //login = new Login();
}

public void startApp()
{

```

```

        if(chekLogin())
        {
            setDisplay(fmLogin); //yg bener yg ini....
            //setDisplay(listMain);
        }
        else
        {
            setDisplay(listMain);
        }
    }

    public void pauseApp() {
    }

    public void destroyApp(boolean unconditional)
    {
        listMain=null;
    }

    private boolean chekLogin() {
        RecordManager rm = new RecordManager("LOGIN");
        return (rm.numRecords()<2);
    }

    private void playSuara()
    {
        try {
            //ByteArrayInputStream recordedInputStream = new
            ByteArrayInputStream(recordedSoundArray);
            ByteArrayInputStream recordedInputStream = new
            ByteArrayInputStream(byteAudio);
            Player p2 = Manager.createPlayer(recordedInputStream,
            "audio/basic");
            p2.prefetch();
            p2.start();
            //throw new UnsupportedOperationException("Not yet
            implemented");
        } catch (IOException ex) {
            ex.printStackTrace();
        } catch (MediaException ex) {
            ex.printStackTrace();
        }
        //throw new UnsupportedOperationException("Not yet
        implemented");
    }
    private void playMovie()
    {
        try {
            Form fMovie = new Form("Movie");
            fMovie.addCommand(cmBack);
            //ByteArrayInputStream recordedInputStream = new
            ByteArrayInputStream(recordedVideoArray);
            ByteArrayInputStream recordedInputStream = new
            ByteArrayInputStream(byteVideo);
            Player mPlay = Manager.createPlayer(recordedInputStream,
            "video/3gpp");
            mPlay.realize();
        }
    }

```

```

        mVideoControl = (VideoControl) mPlay.getControl
("VideoControl");

        fMovie.append((Item) mVideoControl.initDisplayMode
(mVideoControl.USE_GUI_PRIMITIVE, null));

        mVideoControl.setDisplaySize(320, 240);

        mPlay.start();

        fMovie.setCommandListener(this);
        display.setCurrent(fMovie);
    } catch (IOException ex) {
        ex.printStackTrace();
    } catch (MediaException ex) {
        ex.printStackTrace();
    }
}

}
private void setDisplay(Displayable d)
{
    display.setCurrent(d);
}
public void setDisplay(Alert a, Displayable d)
{
    display.setCurrent(a, d);
}
public Displayable getMain()
{
    return listMain;
}
public void dapatJudul(String judul)
{
    Judul=judul;
    System.out.println(Judul);
}

public void textHelp()
{
    String help ="Login\nIsikanlah username dan password Anda.
Username" +
        "dan passord tersebut akan digunakan sebagai
identitas Anda " +
        "ketika meng-upload berita\n" +
        "Title : Untuk memasukkan judul\n" +
        "Text : Untuk memasukkan berita text\n" +
        "Image : Untuk memasukkan gambar, baik dari file atau
mengambil langsung " +
        "menggunakan kamera\n" +
        "Voice : Memasukkan suara, baik dari file atau
merekam secara langsung\n" +
        "Video : Memasukkan video, baik dari file atau
merekam secara langsung";
    mTextBox = new TextBox("Help",help,1024,TextField.ANY);
    mTextBox.addCommand(cmBack);
    mTextBox.setCommandListener(this);
    display.setCurrent(mTextBox);
}

}

public void showInfo(String respon, Displayable d)
{
    Alert alert = new Alert("Info",respon,null, AlertType.INFO);
    alert.setTimeout(Alert.FOREVER);
}

```

```

        //alert.addCommand(cmBack);
        display.setCurrent(alert,d);
        //startApp();
    }
    public Displayable getCurrentDisplay()
    {
        return display.getCurrent();
    }
    public void listKategori()
    {
        listKtgr = new List("Kategori Berita",List.IMPLICIT,
MENU_KATEGORI,null);
        listKtgr.addCommand(cmBack);
        listKtgr.addCommand(cmSelect);
        listKtgr.setCommandListener(this);
        display.setCurrent(listKtgr);
    }
    public void listPicture()
    {
        listGambar = new List("Menu Gambar",List.IMPLICIT,
MENU_GAMBAR,null);
        listGambar.addCommand(cmBack);
        listGambar.addCommand(cmSelect);
        listGambar.setCommandListener(this);
        display.setCurrent(listGambar);
    }
    public void listAudio()
    {
        listSuara = new List("Menu Suara",List.IMPLICIT,
MENU_SUARA,null);
        listSuara.addCommand(cmBack);
        listSuara.addCommand(cmSet);
        listSuara.addCommand(cmSelect);
        listSuara.setCommandListener(this);
        display.setCurrent(listSuara);
    }
    public void listVideo()
    {
        listFilm = new List("Menu Movie", List.IMPLICIT,
MENU_FILM,null);
        listFilm.addCommand(cmBack);
        listFilm.addCommand(cmSelect);
        listFilm.setCommandListener(this);
        display.setCurrent(listFilm);
    }

    public FileBrowser getPictureBrowser()
    {
        if (gambarBrowser == null) {
            //Form gambarForm= new Form("");
            gambarBrowser= new FileBrowser(getDisplay
());
            gambarBrowser.setCommandListener(this);
            gambarBrowser.addCommand
(FileBrowser.SELECT_FILE_COMMAND);
            gambarBrowser.addCommand(cmBack);
            gambarBrowser.setTitle("Pilih File
Gambar");
        }
        return gambarBrowser;
    }
}
//video browser

```

```

/*
public FileBrowser getVideoBrowser()
{
    if (videoBrowser == null) {
        //Form gambarForm= new Form("");
        videoBrowser= new FileBrowser(getDisplay
());
        videoBrowser.setCommandListener(this);
        videoBrowser.addCommand
(FileBrowser.SELECT_FILE_COMMAND);
        videoBrowser.addCommand(cmBack);
        videoBrowser.setTitle("Pilih File
Video");
    }
    return videoBrowser;
}
*/
/*
public FileBrowser getAudioBrowser()
{
    if (audioBrowser == null) {
        //Form gambarForm= new Form("");
        audioBrowser= new FileBrowser(getDisplay
());
        audioBrowser.setCommandListener(this);
        audioBrowser.addCommand
(FileBrowser.SELECT_FILE_COMMAND);
        audioBrowser.addCommand(cmBack);
        audioBrowser.setTitle("Pilih File
Suara");
    }
    return audioBrowser;
}
*/

private void showKamera() {
    try {
        mForm_2 = new Form("Kamera");
        mForm_2.addCommand(cmBack);
        mForm_2.addCommand(cmCapture);
        //mForm_2.addCommand(cmRecMovie);

        mForm_2.setCommandListener(this);

        mPlayer = Manager.createPlayer("capture://video");
        mPlayer.realize();

        mVideoControl = (VideoControl)mPlayer.getControl
("VideoControl");
        mForm_2.append(((Item)mVideoControl.initDisplayMode
(mVideoControl.USE_GUI_PRIMITIVE, null)));
        mVideoControl.setDisplaySize(320, 240);

        mPlayer.start();
        display.setCurrent(mForm_2);

    } catch (IOException ex) {
        ex.printStackTrace();
    } catch (MediaException ex) {
        ex.printStackTrace();
    }
}

```

```

}
private void movieRecord() {
    try {

        fmRecMovie = new Form("Kamera");
        fmRecMovie.addCommand(cmBack);
        fmRecMovie.addCommand(cmRecMovie);

        fmRecMovie.setCommandListener(this);

        mPlayer3 = Manager.createPlayer("capture://video");
        mPlayer3.realize();

        mVideoControl = (VideoControl)mPlayer3.getControl
("VideoControl");
        fmRecMovie.append(((Item)mVideoControl.initDisplayMode
(mVideoControl.USE_GUI_PRIMITIVE, null)));
        mVideoControl.setDisplaySize(320, 240);

        rc2 = (RecordControl) mPlayer3.getControl
("RecordControl");

        mPlayer3.start();
        display.setCurrent(fmRecMovie);

    } catch (IOException ex) {
        ex.printStackTrace();
    } catch (MediaException ex) {
        ex.printStackTrace();
    }
}

private void audioRecord()
{
    try {

        mPlayer2 = Manager.createPlayer("capture://audio?
encoding=pcm");
        mPlayer2.realize();
        rc = (RecordControl) mPlayer2.getControl
("RecordControl");
        rawAudio = new ByteArrayOutputStream();
        rc.setRecordStream(rawAudio);
        rc.startRecord();
        //fmRecAudio.append("recording...");
        mPlayer2.prefetch();
        mPlayer2.start();
        Thread.currentThread().sleep(lamaRekam);
        rc.stopRecord();
        rc.commit();
        byteAudio = rawAudio.toByteArray();
        mPlayer2.stop();
        mPlayer2.close();
        r=3;

    } catch (InterruptedException ex) {
        ex.printStackTrace();
    } catch (IOException ex) {
        ex.printStackTrace();
    } catch (MediaException ex) {

```

```

        ex.printStackTrace();
    }

}

private void showResumes(String insert)
{
    mResume = new TextBox("Resume",insert,2048,TextField.ANY);
    mResume.setCommandListener(this);
    mResume.addCommand(cmBack);
    mResume.addCommand(cmUpload);

    display.setCurrent(mResume);
}

public void commandAction(Command c, Displayable d)
{
    if(c==cmExit)
    {
        destroyApp(true);
        notifyDestroyed();
    }

    if(c==cmSet)
    {
        if(d==listMain)
        {
            setDisplay(fmSetting);
        }
        else if(d==listSuara)
        {
            fmDurasirekam= new Form("Durasi Rekam Suara");
            durasiRekam = new TextField("waktu rekaman(detik)
:","",4,TextField.NUMERIC);
            fmDurasirekam.append(durasiRekam);
            fmDurasirekam.addCommand(cmBack);
            fmDurasirekam.addCommand(cmSimpan);
            display.setCurrent(fmDurasirekam);
            fmDurasirekam.setCommandListener(this);
        }
    }
    else if(c==cmSimpan)
    {
        String str = durasiRekam.getString();
        lamaRekam=Integer.parseInt(str)*1000;
        System.out.println(lamaRekam);
        startApp();
    }
    else if(c==cmLihat)
    {
        setDisplay(fmRespon);
        gIn=new Gauge("interactive", true,100,0);
        fmRespon.append(gIn);

        PostSenderRes ss2 = new PostSenderRes(this);
        ss2.start() ;
    }
}

```



```
else if(c==cmCheck)
{
    //String binaryStr = "";
    /*
    byte[] unComp=Text.getBytes();
    System.out.println("Size before: "+unComp.length*8);
    byte[] Comp=null;
    LZWN lzw2= new LZWN(unComp);
    lzw2.compress();
    Comp=lzw2.hasil();
    System.out.println("Size belong: "+Comp.length*8);
    data=Comp;
    */

    /*
    System.out.println("Comp.length beyond: ");
    System.out.println(Comp.length);
    System.out.println("Data per byte: ");
    */

}
else if (c==cmNext)
{
    int cek = p*q*r*s;
    String gbr = null,suara = null,video = null;
    if(cek==1)
    {
        //post untuk berita saja
        int jd=Judul.length();
        jd +=kategori.length();
        jd +=Text.length;
        jd =(jd*8)/1000;
        jd +=8;
        gbr="kosong";
        suara="kosong";
        video="kosong";
        resume="Judul : "+Judul+"<" +kategori+">

\n"+Text_asli+"\n" +
            "Gambar:"+gbr+
            "\nSuara:"+suara+
            "\nVideo:"+video+
            "\nEstimasi Bytesize: "+jd+"Kbyte";
    }
    else if(cek==2)
    {
        //post untuk berita dengan gambar
        gbr="ada";
        suara="kosong";
        video="kosong";
        int jd=Judul.length();
        jd +=kategori.length();
        jd +=Text.length;
        jd =(jd*8)/1000;
        jd +=8;
        jd+=byteGambar.length/1000;
        resume="Judul : "+Judul+"<" +kategori+">

\n"+Text_asli+"\n" +
            "Gambar:"+gbr+
            "\nSuara:"+suara+
            "\nVideo:"+video+
            "\nEstimasi Bytesize: "+jd+"Kbyte";
        System.out.println(byteGambar.length);
    }
}
```

```

}
else if(cek==3)
{
    //post untuk berita dengan suara
    gbr="kosong";
    suara="ada";
    video="kosong";
    int jd=Judul.length();
    jd +=kategori.length();
    jd +=Text.length();
    jd =(jd*8)/1000;
    jd +=8;
    jd+=byteAudio.length/1000;
    resume="Judul : "+Judul+"<" +kategori+">
\n"+Text_asli+"\n" +
        "Gambar:"+gbr+
        "\nSuara:"+suara+
        "\nVideo:"+video+
        "\nEstimasi Bytesize: "+jd+"Kbyte";
}
else if(cek==4)
{
    //post untuk berita dengan video
    gbr="kosong";
    suara="kosong";
    video="ada";
    int jd=Judul.length();
    jd +=kategori.length();
    jd +=Text.length();
    jd =(jd*8)/1000;
    jd +=8;
    jd+=byteVideo.length/1000;
    resume="Judul : "+Judul+"<" +kategori+">
\n"+Text_asli+"\n" +
        "Gambar:"+gbr+
        "\nSuara:"+suara+
        "\nVideo:"+video+
        "\nEstimasi Bytesize: "+jd+"Kbyte";
}
else if(cek==6)
{
    //post untuk berita dengan gambar dengan suara
    gbr="ada";
    suara="ada";
    video="kosong";
    int jd=Judul.length();
    jd +=kategori.length();
    jd +=Text.length();
    jd =(jd*8)/1000;
    jd +=8;
    jd+=byteAudio.length/1000;
    jd+=byteGambar.length/1000;
    resume="Judul : "+Judul+"<" +kategori+">
\n"+Text_asli+"\n" +
        "Gambar:"+gbr+
        "\nSuara:"+suara+
        "\nVideo:"+video+
        "\nEstimasi Bytesize: "+jd+"Kbyte";
}
else if(cek==8)
{
    //post untuk berita dengan gambar dengan video

```

```

        gbr="ada";
        suara="kosong";
        video="ada";
        int jd=Judul.length();
            jd +=kategori.length();
            jd +=Text.length;
        jd+=byteGambar.length/1000;
        jd+=byteVideo.length/1000;
        resume="Judul : "+Judul+"<" +kategori+">
\n"+Text_asli+"\n" +
            "Gambar:"+gbr+
            "\nSuara:"+suara+
            "\nVideo:"+video+
            "\nEstimasi Bytesize: "+jd+"Kbyte";
    }
    else if(cek==12)
    {
        //post untuk berita dengan suara dengan video
        gbr="kosong";
        suara="ada";
        video="ada";
        int jd=Judul.length();
            jd +=kategori.length();
            jd +=Text.length;
        jd+=byteAudio.length/1000;
        jd+=byteVideo.length/1000;
        resume="Judul : "+Judul+"<" +kategori+">
\n"+Text_asli+"\n" +
            "Gambar:"+gbr+
            "\nSuara:"+suara+
            "\nVideo:"+video+
            "\nEstimasi Bytesize: "+jd+"Kbyte";
    }
    else if(cek==24)
    {
        //post untuk berita gambar + suara + video
        gbr="ada";
        suara="ada";
        video="ada";
        int jd=Judul.length();
            jd +=kategori.length();
            jd +=Text.length;
        jd+=byteGambar.length/1000;
        jd+=byteAudio.length/1000;
        jd+=byteVideo.length/1000;
        resume="Judul : "+Judul+"<" +kategori+">
\n"+Text_asli+"\n" +
            "Gambar:"+gbr+
            "\nSuara:"+suara+
            "\nVideo:"+video+
            "\nEstimasi Bytesize: "+jd+"Kbyte";
    }
    //System.out.println(resume);
    showResumes (resume);
}
else if(c==gambarBrowser.SELECT_FILE_COMMAND)
{
    fileGambar = gambarBrowser.getSelectedFileURL();
    //dapetin byte gambar

```

```

        try {
FileConnection fc =
        (FileConnection)Connector.open(fileGambar);

if (!fc.exists()) {
        throw new IOException("File does not exists");
}

InputStream fis = fc.openInputStream();

//int i = fis.available();
//byteGambar = new byte[i];
byteGambar = new byte[100000];
fis.read(byteGambar);

//int length = fis.read(b);
//Image image = Image.createImage(fis);
//dataFilegambar = image;
//byte[] dataImage=image.getClass()
//datamentah =
fis.close();
fc.close();
q=2;

} catch (Exception e)
{}
        startApp();
}
/*
else if(c==videoBrowser.SELECT_FILE_COMMAND)
{
        //try {
        fileVideo = videoBrowser.getSelectedFileURL();
        //dapetin byte gambar

        try {
FileConnection fc =
        (FileConnection)Connector.open(fileVideo);
if (!fc.exists()) {
        throw new IOException("File does not exists");
}
//ByteArrayInputStream bis = fc.
//int tempat = fc.
InputStream fis = fc.openInputStream();
//int i = fis.available();
//int length = fis.read(byte[1024] b);

byteVideo = new byte[20000];
fis.read(byteVideo);
fis.close();
fc.close();
s=4;
} catch (Exception e)
{}
        startApp();
}
/*
//ByteArrayInputStream bis = (ByteArrayInputStream)
getClass().getResourceAsStream(fileVideo);
Form fMovie = new Form("Movie");
fMovie.addCommand(cmBack);
//ByteArrayInputStream recordedInputStream = new
ByteArrayInputStream(recordedVideoArray);
//ByteArrayInputStream recordedInputStream = new
ByteArrayInputStream(byteVideo);

```

```

        Player mPlay = Manager.createPlayer(getClass
().getResourceAsStream(fileVideo), "video/3gpp");
        mPlay.realize();
        mVideoControl = (VideoControl) mPlay.getControl
("VideoControl");
        fMovie.append((Item) mVideoControl.initDisplayMode
(mVideoControl.USE_GUI_PRIMITIVE, null));
        mVideoControl.setDisplaySize(320, 240);
        mPlay.start();
        fMovie.setCommandListener(this);
        display.setCurrent(fMovie);
    } catch (IOException ex) {
        ex.printStackTrace();
    } catch (MediaException ex) {
        ex.printStackTrace();
    }
}
*/
/*
else if(c==audioBrowser.SELECT_FILE_COMMAND)
{
    fileAudio = audioBrowser.getSelectedFileURL();
    //dapetin byte gambar
    try {
        FileConnection fc =
            (FileConnection)Connector.open(fileAudio);

        if (!fc.exists()) {
            throw new IOException("File does not exists");
        }

        InputStream fis = fc.openInputStream();
        int i = fis.available();
        //DataOutputStream dos = new DataOutputStream(fis);
        //byte[] dataImage=fis.

        //int length = fis.read(byte[1024] b);
        byteAudio = new byte[i];
        fis.read(byteAudio);

        //int length = fis.read(b);
        //Image image = Image.createImage(fis);
        //dataFilegambar = image;
        //byte[] dataImage=image.getClass()
        //datamentah =
        fis.close();
        fc.close();

        r=3;

    } catch (Exception e)
    {}
    startApp();
}
*/
/*
else if(c==cmStop || d==fmRecAudio)
{
    try {
        rc.stopRecord();
        rc.commit();
        //recordedSoundArray = rawAudio.toByteArray();
        byteAudio = rawAudio.toByteArray();

```

```

        mPlayer2.stop();
        mPlayer2.close();
    } catch (MediaException ex) {
        ex.printStackTrace();
    } catch (IOException ex) {
        ex.printStackTrace();
    }
}
*/
//else if(c==cmSelect && d==listMain)
else if(c==cmSelect)
{
    if(d==listMain)
    {
        int menuIndex = listMain.getSelectedIndex();

        switch(menuIndex)
        {
            case 0:
            {
                //System.out.println("Insert Title");
                setDisplay(fmJudul);//listMain seharusnya Form
                Isi Judul

                break;
            }
            case 1:
            {
                //System.out.println("Kategori");
                listKategori();
                //setDisplay(fmJudul);//listMain seharusnya Form
                Isi Judul

                break;
            }
            case 2:
            {
                //System.out.println("Insert Text");
                showTextBox();
                //
                //System.out.println(Text);
                //setDisplay(listMain);//listMain seharusnya
                Form Text

                break;
            }
            case 3:
            {
                //System.out.println("Insert Picture");
                listPicture();
                //setDisplay(listMain);//listMain seharusnya
                Form Insert Picture

                break;
            }
            case 4:
            {
                //System.out.println("Insert Voice");
                listAudio();//listMain seharusnya Form Insert
                Voice

                break;
            }
            case 5:
            {
                //System.out.println("Insert Video");
                listVideo();
                //setDisplay(listMain);//listMain seharusnya

```

Form Insert Video

```
        break;
    }
    case 6:
    {
        //System.out.println("Help");
        //setDisplay(fmHelp);//listMain seharusnya
Form Help
        textHelp();
        break;
    }
}
else if(d==listGambar)
{
    int indek = listGambar.getSelectedIndex();
    switch (indek)
    {
        case 0:
        {
            //System.out.println("ambil dari direktori");
            switchDisplayable(null, getPictureBrowser());
            //System.out.println(getPictureBrowser
().getSelectedFileURL());
            //fileGambar=getPictureBrowser
().getSelectedFileURL();
            break;
        }
        case 1:
        {
            //System.out.println("ambil dari kamera");
            showKamera();

            break;
        }
    }
}
else if(d==listSuara)
{
    int indek = listSuara.getSelectedIndex();
    switch (indek)
    {
        case 0:
        {
            //System.out.println("ambil dari direktori");
            //switchDisplayable(null, getAudioBrowser());
            //System.out.println("rekam");

            audioRecord();

            break;
        }
        case 1:
        {
            //System.out.println("rekam");
            playSuara();

            break;
        }
    }
}
else if(d==listFilm)
{
    int indek = listFilm.getSelectedIndex();
```

```

switch (indek)
{
    case 0:
    {
        //System.out.println("ambil dari direktori");
        //switchDisplayable(null, getVideoBrowser());
        //System.out.println("rekam");
        movieRecord();
        break;
    }
    case 1:
    {
        //System.out.println("rkam");
        playMovie();
        break;
    }
}
}
else if(d==listKtgr)
{
    int indek = listKtgr.getSelectedIndex();
    switch (indek)
    {
        case 0:
        {
            kategori="1";
            startApp();
        }
        break;
        case 1:
        {
            kategori="2";
            startApp();
        }
        break;
        case 2:
        {
            kategori="3";
            startApp();
        }
        break;
        case 3:
        {
            kategori="4";
            startApp();
        }
        break;
        case 4:
        {
            kategori="5";
            startApp();
        }
        break;
        case 5:
        {
            kategori="6";
            startApp();
        }
        break;
        case 6:
        {
            kategori="7";
            startApp();
        }
    }
}
}

```



```

        }
        break;
    }
}
else if(c==cmBack || d==mTextBox)
{
    startApp();
}
else if(c==cmSubmit|| d==textBox)
{
    byte[] unComp=textBox.getString().getBytes();
    System.out.println("Size before: "+unComp.length*8);
    Text_asli=textBox.getString();
    byte[] Comp=null;
    LZWNew lzw2= new LZWNew(unComp);
    lzw2.compress();
    Comp=lzw2.hasil();
    System.out.println("Size after: "+Comp.length*8);
    if(unComp.length<Comp.length)
    {
        Text=unComp;
    }
    else
    {
        Text=Comp;
    }
    //data=Comp;

    System.out.println(Text.toString());
    startApp();
}

else if(c==cmUpload)
{
    waktu=System.currentTimeMillis();
    //PostSenderExp ss2 = new PostSenderExp(this,
data,Judul, Text, kategori);
    //ss2.start();

    //System.out.println("uploading...");
    if(Judul==null || Text==null || kategori==null)
    {
        Alert alert = new Alert("Awas", "Judul, Kategori, dan
Isi harus diisi!!", null, AlertType.WARNING);
        alert.setTimeout(Alert.FOREVER);
        display.setCurrent(alert);
    }
    else
    {
        int cek=p*q*r*s;
        System.out.println(cek);
        if(cek==1)
        {
            //post untuk berita saja
            PostSender1 ss2 = new PostSender1(this, Judul, Text,
kategori);
            ss2.start();
        }
        else if(cek==2)
        {
            //post untuk berita dengan gambar
            PostSender2 ss2 = new PostSender2(this, byteGambar,
Judul, Text, kategori);

```

```

        ss2.start();
    }
    else if(cek==3)
    {
        //post untuk berita dengan suara
        PostSender3 ss2 = new PostSender3(this, byteAudio,
Judul, Text, kategori);
        ss2.start() ;
    }
    else if(cek==4)
    {
        //post untuk berita dengan video
        PostSender4 ss2 = new PostSender4(this, byteVideo,
Judul, Text, kategori);
        ss2.start();
    }
    else if(cek==6)
    {
        //post untuk berita dengan gambar dengan suara
        PostSender5 ss2 = new PostSender5(this, byteGambar,
byteAudio, Judul, Text, kategori);
        ss2.start();
    }
    else if(cek==8)
    {
        //post untuk berita dengan gambar dengan video
        PostSender6 ss2 = new PostSender6(this, byteGambar,
byteVideo, Judul, Text, kategori);
        ss2.start();
    }
    else if(cek==12)
    {
        //post untuk berita dengan suara dengan video
        PostSender7 ss2 = new PostSender7(this, byteAudio,
byteVideo, Judul, Text, kategori);
        ss2.start();
    }
    else if(cek==24)
    {
        //post untuk berita gambar + suara + video
        PostSender8 ss2 = new PostSender8(this, byteGambar,
byteAudio, byteVideo, Judul, Text, kategori);
        ss2.start();
    }
    }
    else if(c==cmCapture)
    {
        new Thread()
        {

            public void run() {
                try {
                    byte[] raw = mVideoControl.getSnapshot
("encoding=jpeg");
                    Image image = Image.createImage(raw, 0, raw.length);
                    dataCapturegambar=image;
                    byteGambar=raw;
                    q=2;
                }
                catch (MediaException ex) {
                    ex.printStackTrace();
                }
            }
        }
    }
}

```

```

        }.start();
    }

    else if(c==cmRecMovie)
    {
        fmRecMovie.removeCommand(cmRecMovie);
        fmRecMovie.addCommand(cmStop);
        rawVideo = new ByteArrayOutputStream();
        rc2.setRecordStream(rawVideo);
        rc2.startRecord();
        //Thread.currentThread().sleep(5000);
    }

    else if(c==cmStop)
    {
        try {
            rc2.stopRecord();
            rc2.commit();
            //2recordedVideoArray = rawVideo.toByteArray();
            byteVideo = rawVideo.toByteArray();
            mPlayer3.stop();
            mPlayer3.close();
            s=4;
            //d=4;

        } catch (MediaException ex) {
            ex.printStackTrace();
        } catch (IOException ex) {
            ex.printStackTrace();
        }
    }
}

private void showTextBox() {
    textBox = new TextBox("Isi",null,2048,TextField.ANY);
    textBox.setCommandListener(this);
    textBox.addCommand(cmBack);
    textBox.addCommand(cmSubmit);
    display.setCurrent(textBox);
}

public void switchDisplayable(Alert alert, Displayable
nextDisplayable) {
    // write pre-switch user code here
    Display display = getDisplay();
    if (alert == null) {
        display.setCurrent(nextDisplayable);
    } else {
        display.setCurrent(alert, nextDisplayable);
    }
    // write post-switch user code here
}

public Display getDisplay () {
    return Display.getDisplay(this);
}
}

```



```

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package midletMCM;

import javax.microedition.lcdui.Command;
import javax.microedition.lcdui.CommandListener;
import javax.microedition.lcdui.Displayable;
import javax.microedition.lcdui.Form;
import javax.microedition.lcdui.List;
import javax.microedition.lcdui.TextBox;
import javax.microedition.lcdui.TextField;

/**
 *
 * @author ajib
 */
class FormHelp extends Form implements CommandListener{
    private MCMMidlet midlet;
    private Displayable previousDisplay;
    private Command cmBack;
    String help;

    public FormHelp(MCMMidlet midlet, Displayable previousDisplay)
    {
        super("Login");
        this.midlet=midlet;
        this.previousDisplay=previousDisplay;
        cmBack = new Command("back",Command.BACK,0);

        help = "Petunjuk penggunaan ...";

        //TextBox mTextBox = new TextBox
        ("Help",help,20,TextField.ANY);
        TextField mTextField = new TextField("petunjuk","ini adalah
        petunjuk",20, TextField.ANY);

        append(mTextField);
        addCommand(cmBack);
    }

    public void commandAction(Command c, Displayable d)
    {
        if(c==cmBack)
        {
            midlet.startApp();
        }
    }
}
}

```

```

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package midletMCM;

import javax.microedition.lcdui.Command;
import javax.microedition.lcdui.CommandListener;
import javax.microedition.lcdui.Display;
import javax.microedition.lcdui.Displayable;
import javax.microedition.lcdui.Form;
import javax.microedition.lcdui.TextField;

/**
 *
 * @author ajib
 */
class FormJudul extends Form implements CommandListener{

    private MCMMidlet midlet;
    private Command cmBack, cmSubmit;
    private TextField mTextField;
    private Displayable previousDisplay;
    private Display display;

    public FormJudul(MCMMidlet midlet, Displayable previousdisplay)
    {
        super("Judul");
        this.midlet=midlet;
        this.previousDisplay=previousdisplay;

        mTextField= new TextField("Judul",null,100,TextField.ANY);
        cmBack= new Command("back",Command.BACK,0);
        cmSubmit = new Command("tambah",Command.OK,0);

        append(mTextField);

        addCommand(cmBack);
        addCommand(cmSubmit);
        setCommandListener(this);
        //display.setCurrent(this);

    }

    public void commandAction(Command c, Displayable d) {
        if(c==cmBack)
        {
            midlet.startApp();
        }
        if(c==cmSubmit)
        {
            midlet.dapatJudul(mTextField.getString());
            midlet.startApp();
        }
    }

}

```

```

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package midletMCM;

import javax.microedition.lcdui.Alert;
import javax.microedition.lcdui.AlertType;
import javax.microedition.lcdui.Command;
import javax.microedition.lcdui.CommandListener;
import javax.microedition.lcdui.Display;
import javax.microedition.lcdui.Displayable;
import javax.microedition.lcdui.Form;
import javax.microedition.lcdui.TextField;

/**
 *
 * @author ajib
 */
class FormLogin extends Form implements CommandListener
{
    private TextField
    tfUsername,tfPassword,tfNama,tfNomortelp,tfEmail;

    private final String RS="LOGIN";
    private MCMMidlet midlet;
    private Displayable previousDisplay;
    private Display display;

    private Command cmExit,cmLogin;

    private Login login;
    public String status;

    public FormLogin(MCMMidlet midlet, Displayable previousDisplay)
    {
        super("Form Registrasi");
        this.midlet=midlet;
        this.previousDisplay=previousDisplay;

        tfUsername = new TextField("Username*", "", 35, TextField.ANY);
        tfPassword = new TextField
        ("Password*", "", 35, TextField.PASSWORD);
        tfNama = new TextField("Nama Lengkap*", "", 50, TextField.ANY);
        tfNomortelp = new TextField("No
        HP*", "", 35, TextField.PHONENUMBER);
        tfEmail = new TextField("Email", "", 50, TextField.ANY);

        cmExit = new Command("keluar", Command.EXIT, 0);
        cmLogin = new Command("simpan", Command.OK, 0);

        append(tfUsername);
        append(tfPassword);
        append(tfNama);
        append(tfNomortelp);
        append(tfEmail);
        append("*)Harap diisi dengan benar!\nInfo selanjutnya bisa
        Anda kunjungi http://www.mcmedia.com");

        addCommand(cmExit);
        addCommand(cmLogin);
        setCommandListener(this);
        login = new Login();
    }
}

```

```

}
//hapus field
public void hapusField()
{
    tfUsername.setString("");
    tfPassword.setString("");
}

public void commandAction(Command c, Displayable d)
{
    if(c==cmExit)
    {
        midlet.destroyApp(true);
        midlet.notifyDestroyed();
        //System.exit(1);
        //notifyDestroyed();
    }
    else if(c==cmLogin)
    {
        String username = tfUsername.getString().trim();
        String password = tfPassword.getString().trim();
        String nama = tfNama.getString().trim();
        String nomor = tfNomortelp.getString().trim();
        String email = tfEmail.getString().trim();
        //
        if(username.equals("")||password.equals("")||nama.equals
        ("")||nomor.equals(""))
        {
            dataNotValid();
        }
        else
        {
            login.setUsername(username);
            login.setPassword(password);
            login.setNamaLengkap(nama);
            login.setNoHP(nomor);
            login.setEmail(email);
            login.setAlamatweb
            ("http://www.ee.ui.ac.id/telkom/mcm2");

            //LZWCompression lc = new LZWCompression();
            //Register to Website
            PostSender ss = new PostSender(midlet,username,
            password, nama, nomor, email);
            ss.start();
        }
        midlet.startApp();
    }
}
private void dataNotValid()
{
    Alert alert1 = new Alert("Alert","Isilah username,\npassword,
\nnama lengkap, dan \nno HP dengan benar!",null,AlertType.INFO);
    alert1.setTimeout(3000);
    midlet.setDisplay(alert1, this);
}
}

```


}



```

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package midletMCM;

import javax.microedition.lcdui.Alert;
import javax.microedition.lcdui.AlertType;
import javax.microedition.lcdui.Command;
import javax.microedition.lcdui.CommandListener;
import javax.microedition.lcdui.Display;
import javax.microedition.lcdui.Displayable;
import javax.microedition.lcdui.Form;
import javax.microedition.lcdui.TextField;

/**
 *
 * @author ajib
 */
class FormRespon extends Form implements CommandListener
{
    private TextField
    tfUsername,tfPassword,tfNama,tfNomortelp,tfEmail;

    private final String RS="LOGIN";
    private MCMMidlet midlet;
    private Displayable previousDisplay;
    private Display display;

    private Command cmBack;

    private Login login;
    public String status;

    public FormRespon(MCMMidlet midlet, Displayable previousDisplay)
    {
        super("Berita Terkini");
        this.midlet=midlet;
        this.previousDisplay=previousDisplay;
        cmBack = new Command("keluar", Command.BACK,0);
        //append("*)Harap diisi dengan benar!\nInfo selanjutnya bisa
        Anda kunjungi http://www.mcmedia.com");

        addCommand(cmBack);

        setCommandListener(this);
    }
    public void commandAction(Command c, Displayable d)
    {
        if(c==cmBack)
        {
            midlet.startApp();
        }
    }
    private void dataNotValid()
    {
        Alert alert1 = new Alert("Alert", "Downloading
        berita...",null,AlertType.INFO);
        alert1.setTimeout(3000);
        midlet.setDisplay(alert1, this);
    }
}

```

}

}



```

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package midletMCM;

import javax.microedition.lcdui.Alert;
import javax.microedition.lcdui.AlertType;
import javax.microedition.lcdui.Command;
import javax.microedition.lcdui.CommandListener;
import javax.microedition.lcdui.Display;
import javax.microedition.lcdui.Displayable;
import javax.microedition.lcdui.Form;
import javax.microedition.lcdui.TextField;

/**
 *
 * @author ajib
 */
class FormSetting extends Form implements CommandListener
{
    private TextField
    tfUsername,tfPassword,tfNama,tfNomortelp,tfEmail,tfWeb;

    //private final String RS="LOGIN";
    private MCMidlet midlet;
    private Displayable previousDisplay;
    private Display display;

    private Command cmExit,cmLogin;

    private Login login;
    public String status;

    public FormSetting(MCMidlet midlet, Displayable previousDisplay)
    {
        super("Edit data");
        this.midlet=midlet;
        this.previousDisplay=previousDisplay;
        RecordManager rm = new RecordManager("LOGIN");

        tfUsername = new TextField("Username",rm.getRecord
(1),35,TextField.ANY);
        tfPassword = new TextField("Password",rm.getRecord
(2),35,TextField.PASSWORD);
        tfNama = new TextField("Nama Lengkap",rm.getRecord
(3),50,TextField.ANY);
        tfNomortelp = new TextField("No HP",rm.getRecord
(4),35,TextField.PHONENUMBER);
        tfEmail = new TextField("Email",rm.getRecord
(5),50,TextField.EMAILADDR);
        tfWeb = new TextField("Alamat Web",rm.getRecord
(6),50,TextField.ANY);

        cmExit = new Command("keluar", Command.EXIT,0);
        cmLogin = new Command("simpan", Command.OK,0);

        append(tfUsername);
        append(tfPassword);
        append(tfNama);
        append(tfNomortelp);
        append(tfEmail);
        append(tfWeb);
        //append("Bila tidak punya email, kosongkan saja\nUntuk info
lebih lengkap silahkan kunjung http://www.mcm.com/admin");
    }
}

```

```

        addCommand(cmExit);
        addCommand(cmLogin);
        setCommandListener(this);
        login = new Login();
    }
    //hapus field
    public void hapusField()
    {
        tfUsername.setString("");
        tfPassword.setString("");
    }

    public void commandAction(Command c, Displayable d)
    {
        if(c==cmExit)
        {
            midlet.startApp();

            //System.exit(1);
            //notifyDestroyed();
        }
        else if(c==cmLogin)
        {
            String username = tfUsername.getString().trim();
            String password = tfPassword.getString().trim();
            String nama = tfNama.getString().trim();
            String nomor = tfNomortelp.getString().trim();
            String email = tfEmail.getString().trim();
            String web = tfWeb.getString().trim();
            //
            if(username.equals("")||password.equals("")||nomor.equals
            (""))
            {
                dataNotValid();
            }
            else
            {
                login.setUsername(username);
                login.setPassword(password);
                login.setNamaLengkap(nama);
                login.setNoHP(nomor);
                login.setEmail(email);
                login.setAlamatweb(web);

                //LZWCompression lc = new LZWCompression();
                //Register to Website
                PostSender ss = new PostSender(midlet,username,
                password, nama, nomor, email);
                ss.start();

                midlet.startApp();
            }
        }
    }
    private void dataNotValid()
    {
        Alert alert1 = new Alert("Alert","Isilah username dan
        password",null,AlertType.INFO);
    }

```

```
alert1.setTimeout(3000);  
midlet.setDisplay(alert1, this);
```

```
}
```

```
}
```



```

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package midletMCM;

import java.io.IOException;
import java.io.InputStream;

import javax.microedition.io.Connector;
import javax.microedition.io.HttpConnection;
import javax.microedition.lcdui.Alert;
import javax.microedition.lcdui.AlertType;
import javax.microedition.lcdui.Command;
import javax.microedition.lcdui.Display;
import javax.microedition.lcdui.Displayable;
import javax.microedition.lcdui.Form;
import javax.microedition.lcdui.Image;

/**
 *
 * @author ajib
 */
class PostSender implements Runnable
{
    //private MCMMidlet midlet;
    private String username, nama, nomor, email;
    private String password;
    private Command exit;
    private Form mForm_3;
    private String username1;
    private String password1;
    //private String URL =
    "http://www.ee.ui.ac.id/telkom/mcm2/config/post_sender.php";
    private String URL =
    "http://localhost/asa/config/post_sender.php";
    private MCMMidlet midlet;
    private Displayable previousDisplay;
    public String result;
    private Display display;
    //private PostingMidlet midlet;
    byte[] satu;
    int angka;
    private Image image;
    //private FormLogin fmLogin;
    //private Login login;

    public PostSender(MCMMidlet midlet, String username, String
password, String nama, String nomor, String email)
    {

        this.username1 = username;
        this.password1 = password;
        this.nama = nama;
        this.nomor = nomor;
        this.midlet = midlet;
        this.email=email;

    }

    public void run()
    {
        try
        {

```

```

        sendPost();
    }
    catch (Exception e)
    {
        e.printStackTrace();
        System.err.println("Error: " + e.toString());
        networkError(e.toString());
    }
}

public void start()
{
    Thread t = new Thread(this);
    try
    {
        t.start();
    }
    catch (Exception e)
    {
        System.err.println("Error: " + e.toString());
        networkError(e.toString());
    }
}

private void sendPost() throws IOException
{
    InputStream is = null;
    StringBuffer sb = null;
    HttpURLConnection http = null;

    try
    {
        // append the query string onto the URL
        URL += "?username=" + username1 + "&password="+
password1+"&nama="+ nama+"&nomor="+ nomor+"&email="+email;
        // replace not allowed char in the URL
        URL = EncodeURL(URL);
        // establish the connection
        http = (HttpURLConnection) Connector.open(URL);
        // set the request method as GET
        http.setRequestMethod(HttpURLConnection.GET);
        // server response
        if (http.getResponseCode() ==
HttpURLConnection.HTTP_OK)
        {
            sb = new StringBuffer();
            int ch;
            is = http.openInputStream();
            while ((ch = is.read()) != -1)
            {
                sb.append((char) ch);
            }
        }
        else
        {
            System.out.println("Network error");
            networkError();
        }
    }
    catch (Exception e)
    {
        System.err.println("Error: " + e.toString());
        networkError(e.toString());
    }
    finally

```



```

        {
            if (is != null)
                is.close();
            if (sb != null)
                result = sb.toString();
            else
                networkError();
            if (http != null)
                http.close();
        }
        if (result != "")
        {
            System.out.println(result);
            midlet.showInfo("Registrasi
Berhasil",getCurrentDisplay());
        }

        else
        {
            networkError();
        }
    }

    private Displayable getCurrentDisplay()
    {
        // get the current display
        Displayable d = midlet.getCurrentDisplay();
        // if it is an Alert set the next display as the previous
one
        if (d.getClass().getName().equals
("javax.microedition.lcdui.Alert"))
        {
            d = previousDisplay;
        }
        return d;
    }

    private String EncodeURL(String URL)
    {
        URL = replace(URL, 'Ã ', "%E0");
        URL = replace(URL, 'Ã"', "%E8");
        URL = replace(URL, 'Ã@', "%E9");
        URL = replace(URL, 'Ã-', "%EC");
        URL = replace(URL, 'Ã²', "%F2");
        URL = replace(URL, 'Ã¹', "%F9");
        URL = replace(URL, '$', "%24");
        URL = replace(URL, '#', "%23");
        URL = replace(URL, 'Ã£', "%A3");
        URL = replace(URL, '@', "%40");
        URL = replace(URL, '\\', "%27");
        URL = replace(URL, ' ', "%20");

        return URL;
    }

    private String replace(String source, char oldChar, String
dest)

```

```
{
    String ret = "";
    for (int i = 0; i < source.length(); i++)
    {
        if (source.charAt(i) != oldChar)
            ret += source.charAt(i);
        else
            ret += dest;
    }
    return ret;
}

private void networkError()
{
    midlet.showInfo("Network error, try later",
getCurrentDisplay());
}
private void networkError(String msg)
{
    midlet.showInfo(msg, getCurrentDisplay());
}
}
```

```

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package midletMCM;

import java.io.IOException;
import java.io.InputStream;

import java.io.OutputStream;
import javax.microedition.io.Connector;
import javax.microedition.io.HttpConnection;
import javax.microedition.lcdui.Command;
import javax.microedition.lcdui.Display;
import javax.microedition.lcdui.Displayable;
import javax.microedition.lcdui.Form;

/**
 *
 * @author ajib
 */
class PostSender1 implements Runnable
{
    private final String CrLf = "\r\n";
    RecordManager rm = new RecordManager("LOGIN");
    private Command exit;
    private Form mForm_3;
    private String dirGambar, title;
    private byte[] text;
    //private final String URL = rm.getRecord(6);
    private final String URL = "http://localhost/asa/post.php";
    private MCMidlet midlet;
    private Displayable previousDisplay;
    private String result,kategori;
    private Display display;
    private String cek;
    private long waktu2;

    public PostSender1(MCMidlet midlet, String title, byte[] text,
String kategori)
    {
        this.midlet =midlet;
        this.title = title;
        this.text = text;
        this.kategori=kategori;
    }

    public void run()
    {
        try
        {
            sendPost();
        }
        catch (Exception e)
        {
            e.printStackTrace();
            System.err.println("Error: 1" + e.toString());
        }
    }

    public void start()
    {

```

```

new Thread(this).start();
try
{
    //t.start();
}
catch (Exception e)
{
    System.err.println("Error: 2" + e.toString());
    //networkError(e.toString());
}
}

private void sendPost() throws IOException
{
    HttpURLConnection conn = null;
    OutputStream os = null;
    InputStream is = null;
    HttpURLConnection http = null;
    //RecordManager rm = new RecordManager("LOGIN");
    String username = rm.getRecord(1);
    StringBuffer sb;
    //String username = "admin";
    try
    {
        conn = (HttpURLConnection)Connector.open(URL);
        conn.setRequestMethod(HttpURLConnection.POST);
        //proses upload

        //username
        String msg1="";
        msg1 += CrLf +
"-----4664151417711" + CrLf;
        msg1 += "Content-Type: text/plain"+ CrLf;
        //msg1 += "Content-Transfer-Encoding:
7bit"+ CrLf;
        msg1 += "Content-Disposition: inline;
name=\"username\""+CrLf;
        msg1 += ""+CrLf+username;

        //kategori
        String msg2="";
        msg2 += CrLf +
"-----4664151417711" + CrLf;
        msg2 += "Content-Type: multipart/form-
data"+ CrLf;
        //msg2 += "Content-Transfer-Encoding:
7bit"+ CrLf;
        msg2 += "Content-Disposition: inline;
name=\"kategori\""+CrLf;
        msg2 += ""+CrLf+kategori;
        //judul
        String msg3="";
        msg3 += CrLf +
"-----4664151417711" + CrLf;
        msg3 += "Content-Type: multipart/form-
data"+ CrLf;
        //msg3 += "Content-Transfer-Encoding:
7bit"+ CrLf;
        msg3 += "Content-Disposition: inline;
name=\"judul\""+CrLf;

```

```

msg3 += ""+CrLf+title;
//isi
String msg4="";
msg4 += CrLf +
"-----4664151417711" + CrLf;
msg4 += "Content-Type: multipart/form-
data"+ CrLf;
//msg3 += "Content-Transfer-Encoding:
7bit"+ CrLf;
msg4 += "Content-Disposition: inline;
name=\"isi\""+CrLf;
msg4 += ""+CrLf;
/*
String msg5 = "";
msg5 += CrLf
+"-----4664151417711" + CrLf;
msg5 += "Content-Disposition: form-data;
name=\"uploadedfile\"; filename=\"\" +FILE + "\"\" + CrLf;
msg5 += "Content-Type: image/jpeg" +
CrLf;
msg5 += CrLf;
*/
String msg6 = "";
msg6 += CrLf +
"-----4664151417711--" + CrLf;

conn.setRequestProperty("Content-Type",
"multipart/form-data; boundary=-----
4664151417711");
// might not need to specify the content-
length when sending chunked data.
// conn.setRequestProperty("Content-
Length", String.valueOf((message1.length() + message2.length() +
imgData.length)));

System.out.println("open os");
os = conn.getOutputStream();

//write username
System.out.println(msg1);
os.write(msg1.getBytes());
//write kategori
System.out.println(msg2);
os.write(msg2.getBytes());
//write judul
System.out.println(msg3);
os.write(msg3.getBytes());
//write isi
System.out.println(msg4);
os.write(msg4.getBytes());
os.write(text);

//tutup
System.out.println(msg6);
os.write(msg6.getBytes());
os.flush();
os.close();

System.out.println("open is");
/*
is = conn.openInputStream();

```

```

        char buff = 512;
        int len;
        byte []data = new byte[buff];
        do{
            System.out.println("READ");
            len = is.read(data);
            if(len > 0){
                System.out.println(new String
(data, 0, len));
            }
        }while(len>0);
        /*
        //System.out.println("DONE");

        // server response
        /*
        if (http.getResponseCode() ==
HttpConnection.HTTP_OK)
        {
            sb = new StringBuffer();
            int ch;
            is = http.openInputStream();
            while ((ch = is.read()) != -1)
                sb.append((char) ch);
        }
        else
        {
            System.out.println("Network error");
            //networkError();
        }
        */
    }
    finally
    {
        //if (is != null)
        //is.close();
        //if (sb != null)
        //result = sb.toString();
        //else
        //networkError();
        //if (http != null)
        //http.close();
    }

    if (result != "")
    {
        //satu = result.getBytes();
        //satu.
        System.out.println(result);
        //midlet = new MCMMidlet();
        waktu2=System.currentTimeMillis()-midlet.waktu;
        //midlet.showInfo(Long.toString
(waktu2),getCurrentDisplay());
        midlet.showInfo("upload
berhasil...",getCurrentDisplay());
    }

    //midlet.showInfo(result, getCurrentDisplay());

```

```

else
{
    midlet.showInfo("gagal...",getCurrentDisplay
());
    //networkError();
}
//midlet.showInfo(Long.toString
(waktu2),getCurrentDisplay());
}
private Displayable getCurrentDisplay()
{
    // get the current display
    Displayable d = midlet.getCurrentDisplay();
    // if it is an Alert set the next display as the previous
one
    if (d.getClass().getName().equals
("javax.microedition.lcdui.Alert"))
    {
        d = previousDisplay;
    }
    return d;
}
}

```

```

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package midletMCM;

import java.io.IOException;
import java.io.InputStream;

import java.io.OutputStream;
import javax.microedition.io.Connector;
import javax.microedition.io.HttpConnection;
import javax.microedition.lcdui.Command;
import javax.microedition.lcdui.Display;
import javax.microedition.lcdui.Displayable;
import javax.microedition.lcdui.Form;

/**
 *
 * @author ajib
 */
class PostSender2 implements Runnable
{
    private final String CrLf = "\r\n";
    RecordManager rm = new RecordManager("LOGIN");
    //private String password;
    private Command exit;
    private StringBuffer sb;
    private Form mForm_3;
    private String dirGambar, title, kategori;
    private byte[] text;
    //private final String URL = rm.getRecord(6);
    private final String URL = "http://localhost/asa/post.php";
    //private PostingMidlet midlet;
    private MCMMidlet midlet;
    private Displayable previousDisplay;
    private String result;
    private Display display;
    //private PostingMidlet midlet;
    //byte[] satu;
    byte[] byteGambar;
    private long waktu2;
    //int angka;
    //private Image image;

    public PostSender2(MCMMidlet midlet, byte[] byteGambar, String
title, byte[] text, String kategori)
    {
        this.midlet =midlet;
        this.byteGambar=byteGambar;
        this.title = title;
        this.text = text;
        this.kategori = kategori;
        //this.dirGambar=dirGambar;
    }

    public void run()
    {
        try
        {
            sendPost();

```



```

    }
    catch (Exception e)
    {
        e.printStackTrace();
        System.err.println("Error: 1" + e.toString());
        //networkError(e.toString());
    }
}

public void start()
{
    new Thread(this).start();
    try
    {
        //t.start();
    }
    catch (Exception e)
    {
        System.err.println("Error: 2" + e.toString());
        //networkError(e.toString());
    }
}

private void sendPost() throws IOException
{
    String FILE="file:///root1/"+Integer.toString
(title.hashCode())+".jpg";
    HttpURLConnection conn = null;
    OutputStream os = null;
    InputStream is = null;
    HttpURLConnection http = null;
    //RecordManager rm = new RecordManager("LOGIN");
    String username = rm.getRecord(1);
    //is.read(byteGambar);
    try
    {
        // append the query string onto the URL
        //URL += "?title=" + title + "&text="+
text+"&gambar="+gambar;
        // replace not allowed char in the URL
        //URL = EncodeURL(URL);
        // establish the connection
        conn = (HttpURLConnection)Connector.open(URL);
        conn.setRequestMethod(HttpURLConnection.POST);
        //uploading process
        //username
        String msg1="";
        msg1 += CrLf +
"-----4664151417711" + CrLf;
        msg1 += "Content-Type: text/plain"+ CrLf;
        //msg1 += "Content-Transfer-Encoding:
7bit"+ CrLf;
        msg1 += "Content-Disposition: inline;
name=\"username\""+CrLf;
        msg1 += ""+CrLf+username;
        //msg1 += ""+CrLf+"admin";

        //kategori
        String msg2="";
        msg2 += CrLf +
"-----4664151417711" + CrLf;
        msg2 += "Content-Type: multipart/form-
data"+ CrLf;
        //msg2 += "Content-Transfer-Encoding:

```

```

7bit"+ CrLf;
name="\kategori\""+CrLf;
msg2 += "Content-Disposition: inline;
msg2 += ""+CrLf+kategori;
//judul
String msg3="";
msg3 += CrLf +
"-----4664151417711" + CrLf;
msg3 += "Content-Type: multipart/form-
data"+ CrLf;
7bit"+ CrLf;
//msg3 += "Content-Transfer-Encoding:
msg3 += "Content-Disposition: inline;
name="\judul\""+CrLf;
msg3 += ""+CrLf+title;
//isi
String msg4="";
msg4 += CrLf +
"-----4664151417711" + CrLf;
msg4 += "Content-Type: multipart/form-
data"+ CrLf;
7bit"+ CrLf;
//msg3 += "Content-Transfer-Encoding:
msg4 += "Content-Disposition: inline;
name="\isi\""+CrLf;
msg4 += ""+CrLf;
String msg5 = "";
msg5 += CrLf
+"-----4664151417711" + CrLf;
msg5 += "Content-Disposition: form-data;
name="\filegambar\"; filename=\""+FILE + "\""+ CrLf;
msg5 += "Content-Type: image/jpeg" +
CrLf;
msg5 += CrLf;
String msg6 = "";
msg6 += CrLf +
"-----4664151417711--" + CrLf;
String message3 = "";
message3 += CrLf;
message3 += "" + CrLf;
conn.setRequestProperty("Content-Type",
"multipart/form-data; boundary=-----
4664151417711");
// might not need to specify the content-
length when sending chunked data.
// conn.setRequestProperty("Content-
Length", String.valueOf((message1.length() + message2.length() +
imgData.length)));
System.out.println("open os");
os = conn.getOutputStream();
//write username
System.out.println(msg1);
os.write(msg1.getBytes());
//write kategori
System.out.println(msg2);
os.write(msg2.getBytes());
//write judul
System.out.println(msg3);

```

```

os.write(msg3.getBytes());
//write isi
System.out.println(msg4);
os.write(msg4.getBytes());
os.write(text);
/*
System.out.println(msg6);
os.write(msg6.getBytes());
*/

// SEND THE IMAGE

//write image
System.out.println(msg5);
os.write(msg5.getBytes());

int index = 0;
int size = 1024;
do{
    System.out.println("write:" + index);
    if((index+size)>byteGambar.length){
        size = byteGambar.length - index;
    }
    os.write(byteGambar, index, size);
    index+=size;
    //progress(byteGambar.length, index);
}while(index<byteGambar.length);
System.out.println("written:" + index);

System.out.println(msg6);
os.write(msg6.getBytes());

os.flush();

System.out.println("open is");
is = conn.openInputStream();

char buff = 512;
int len;
byte []data = new byte[buff];
do{
    System.out.println("READ");
    len = is.read(data);

    if(len > 0){
        System.out.println(new String
(data, 0, len));
    }
}while(len>0);

System.out.println("DONE");

// server response

if (http.getResponseCode() ==
HttpConnection.HTTP_OK)

```

```

        {
            sb = new StringBuffer();
            int ch;
            is = http.openInputStream();
            while ((ch = is.read()) != -1)
                sb.append((char) ch);
        }
        else
        {
            System.out.println("Network error");
            //networkError();
        }
    }
    catch (Exception e)
    {
        System.err.println("Error:3 " + e.toString());
        //networkError(e.toString());
    }
    finally
    {
        //if (is != null)
            is.close();
        //if (sb != null)
            //result = sb.toString();
        //else
            //networkError();
        //if (http != null)
            //http.close();
    }

    if (result != "")
    {
        //satu = result.getBytes();
        //satu.
        System.out.println(result);
        //midlet = new MCMidlet();
        waktu2=System.currentTimeMillis()-midlet.waktu;

        midlet.showInfo("Berhasil...",getCurrentDisplay());
        //System.out.println(satu);
        //System.out.println(angka);
        //PostingMidlet midlet = new PostingMidlet();
        //midlet.Tampil(result);
        //midlet.showInfo(result, getCurrentDisplay());
        //midlet.showInfo(result, previousDisplay);
        //mForm_3 = new Form("Status");
        // mForm_3.append(result);
        //mForm_3.addCommand(exit);
        //mForm_3.setCommandListener(mForm_3);
        //display.setCurrent(mForm_3);
    }

        //midlet.showInfo(result, getCurrentDisplay());

    else
    {
        midlet.showInfo("gagal",getCurrentDisplay());
        //networkError();
    }

}

private Displayable getCurrentDisplay()
{

```

```
// get the current display
Displayable d = midlet.getCurrentDisplay();
// if it is an Alert set the next display as the previous
one
    if (d.getClass().getName().equals
("javax.microedition.lcdui.Alert"))
    {
        d = previousDisplay;
    }
    return d;
}
}
```



```

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package midletMCM;

import java.io.IOException;
import java.io.InputStream;

import java.io.OutputStream;
import javax.microedition.io.Connector;
import javax.microedition.io.HttpConnection;
import javax.microedition.lcdui.Command;
import javax.microedition.lcdui.Display;
import javax.microedition.lcdui.Displayable;
import javax.microedition.lcdui.Form;

/**
 *
 * @author ajib
 */
class PostSender3 implements Runnable
{
    RecordManager rm = new RecordManager("LOGIN");
    private final String CrLf = "\r\n";
    //private String password;
    private Command exit;
    private Form mForm_3;
    private String dirGambar, title;
    private byte[] text;
    //private final String URL = rm.getRecord(6);
    private final String URL = "http://localhost/asa/post.php";
    //private PostingMidlet midlet;
    private MCMMidlet midlet;
    private Displayable previousDisplay;
    private String result,kategori;
    private Display display;
    private String cek;
    private long waktu2;
    byte[] byteAudio;

    public PostSender3(MCMMidlet midlet, byte[] byteAudio, String
    title, byte[] text, String kategori)
    {
        this.midlet =midlet;
        this.byteAudio=byteAudio;
        this.title = title;
        this.text = text;
        this.kategori=kategori;

        //this.dirGambar=dirGambar;
    }

    public void run()
    {
        try
        {
            sendPost();
        }
        catch (Exception e)
        {

```

```

        e.printStackTrace();
        System.err.println("Error: 1" + e.toString());
        //networkError(e.toString());
    }
}

public void start()
{
    new Thread(this).start();
    try
    {
        //t.start();
    }
    catch (Exception e)
    {
        System.err.println("Error: 2" + e.toString());
        //networkError(e.toString());
    }
}

private void sendPost() throws IOException
{
    String FILE="file:///root1/"+Integer.toString
(title.hashCode())+".wav";
    HttpURLConnection conn = null;
    OutputStream os = null;
    InputStream is = null;
    HttpURLConnection http = null;

    String username = rm.getRecord(1);
    //String username = "admin";
    try
    {
        // append the query string onto the URL
        //URL += "?title=" + title + "&text="+
text+"&gambar="+gambar;
        // replace not allowed char in the URL
        //URL = EncodeURL(URL);
        // establish the connection
        conn = (HttpURLConnection)Connector.open(URL);
        conn.setRequestMethod(HttpURLConnection.POST);
        //uploading process

        // the image is sent between the messages
ni the multipart message.

        //username
        String msg1="";
        msg1 += CrLf +
"-----4664151417711" + CrLf;
        msg1 += "Content-Type: text/plain"+ CrLf;
        //msg1 += "Content-Transfer-Encoding:
7bit"+ CrLf;
        msg1 += "Content-Disposition: inline;
name=\"username\""+CrLf;
        msg1 += ""+CrLf+username;

        //kategori
        String msg2="";
        msg2 += CrLf +
"-----4664151417711" + CrLf;
        msg2 += "Content-Type: multipart/form-

```

```

data"+ CrLf;
//msg2 += "Content-Transfer-Encoding:
7bit"+ CrLf;
msg2 += "Content-Disposition: inline;
name=\"kategori\""+CrLf;
msg2 += ""+CrLf+kategori;
//judul
String msg3="";
msg3 += CrLf +
"-----4664151417711" + CrLf;
msg3 += "Content-Type: multipart/form-
data"+ CrLf;
//msg3 += "Content-Transfer-Encoding:
7bit"+ CrLf;
msg3 += "Content-Disposition: inline;
name=\"judul\""+CrLf;
msg3 += ""+CrLf;
//isi
String msg4="";
msg4 += CrLf +
"-----4664151417711" + CrLf;
msg4 += "Content-Type: multipart/form-
data"+ CrLf;
//msg3 += "Content-Transfer-Encoding:
7bit"+ CrLf;
msg4 += "Content-Disposition: inline;
name=\"isi\""+CrLf;
msg4 += ""+CrLf+text;

String msg5 = "";
msg5 += CrLf
+"-----4664151417711" + CrLf;
msg5 += "Content-Disposition: form-data;
name=\"filesuara\"; filename=\"\" +FILE + "\"\" + CrLf;
msg5 += "Content-Type: image/jpeg" +
CrLf;
msg5 += CrLf;

String msg6 = "";
msg6 += CrLf +
"-----4664151417711--" + CrLf;
String message3 = "";
message3 += CrLf;
message3 += "" + CrLf;

conn.setRequestProperty("Content-Type",
"multipart/form-data; boundary=-----
4664151417711");
// might not need to specify the content-
length when sending chunked data.
// conn.setRequestProperty("Content-
Length", String.valueOf((message1.length() + message2.length() +
imgData.length)));

System.out.println("open os");
os = conn.openOutputStream();

//write username
System.out.println(msg1);
os.write(msg1.getBytes());
//write kategori
System.out.println(msg2);
os.write(msg2.getBytes());

```



```

//write judul
System.out.println(msg3);
os.write(msg3.getBytes());
//write isi
System.out.println(msg4);
os.write(msg4.getBytes());
os.write(text);
/*
System.out.println(msg6);
os.write(msg6.getBytes());
*/

// SEND THE IMAGE

//write image
System.out.println(msg5);
os.write(msg5.getBytes());

int index = 0;
int size = 1024;
do{
    System.out.println("write:" + index);
    if((index+size)>byteAudio.length){
        size = byteAudio.length - index;
    }
    os.write(byteAudio, index, size);
    index+=size;
    //progress(byteGambar.length, index);
// update the progress bar.

}while(index<byteAudio.length);
System.out.println("written:" + index);

System.out.println(msg6);
os.write(msg6.getBytes());

//System.out.println(String.valueOf
(imgData.length));
//os.write(("ini data").getBytes());
//os.write(imgData);

//String message3 = "username";
//os.write(message3.getBytes());
os.flush();

System.out.println("open is");
is = conn.openInputStream();

char buff = 512;
int len;
byte []data = new byte[buff];
do{
    System.out.println("READ");
    len = is.read(data);

    if(len > 0){

```

```

        System.out.println(new String
(data, 0, len));
    }
    }while(len>0);

    System.out.println("DONE");

    // server response
    /*
    if (http.getResponseCode() ==
HttpConnection.HTTP_OK)
    {
        sb = new StringBuffer();
        int ch;
        is = http.openInputStream();
        while ((ch = is.read()) != -1)
            sb.append((char) ch);
    }
    else
    {
        System.out.println("Network error");
        networkError();
    }
    */
}
catch (Exception e)
{
    System.err.println("Error:3 " + e.toString());
    //networkError(e.toString());
}
finally
{
    //if (is != null)
    is.close();
    //if (sb != null)
    //result = sb.toString();
    //else
    //networkError();
    //if (http != null)
    //http.close();
}

if (result != "")
{
    //satu = result.getBytes();
    //satu.
    System.out.println(result);
    waktu2=System.currentTimeMillis()-midlet.waktu;
    //midlet.showInfo(Long.toString
(waktu2),getCurrentDisplay());
    midlet.showInfo("upload
berhasil...",getCurrentDisplay());
    //midlet.showInfo(result,getCurrentDisplay());
}

//midlet.showInfo(result, getCurrentDisplay());

else
{
    midlet.showInfo(result,getCurrentDisplay());
    //networkError();
}
}
}

```

```
private Displayable getCurrentDisplay()
{
    // get the current display
    Displayable d = midlet.getCurrentDisplay();
    // if it is an Alert set the next display as the previous
one
    if (d.getClass().getName().equals
("javax.microedition.lcdui.Alert"))
    {
        d = previousDisplay;
    }
    return d;
}
}
```

```

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package midletMCM;

import java.io.IOException;
import java.io.InputStream;

import java.io.OutputStream;
import javax.microedition.io.Connector;
import javax.microedition.io.HttpConnection;
import javax.microedition.lcdui.Command;
import javax.microedition.lcdui.Display;
import javax.microedition.lcdui.Displayable;
import javax.microedition.lcdui.Form;

/**
 *
 * @author ajib
 */
class PostSender4 implements Runnable
{
    private final String CrLf = "\r\n";
    RecordManager rm = new RecordManager("LOGIN");
    //private String password;
    private Command exit;
    private Form mForm_3;
    private String dirGambar, title;
    private byte[] text;
    //private final String URL = rm.getRecord(6);
    private final String URL = "http://localhost/asa/post.php";
    //private PostingMidlet midlet;
    private MCMMidlet midlet;
    private Displayable previousDisplay;
    private String result, kategori;
    private Display display;
    private String cek;
    byte[] byteVideo;
    private long waktu2;
    public PostSender4(MCMMidlet midlet, byte[] byteVideo, String
title, byte[] text, String kategori)
    {

        this.midlet =midlet;
        this.byteVideo=byteVideo;
        this.title = title;
        this.text = text;
        this.kategori=kategori;

        //this.dirGambar=dirGambar;
    }

    public void run()
    {
        try
        {
            sendPost();
        }
        catch (Exception e)
        {
            e.printStackTrace();
        }
    }
}

```

```

        System.err.println("Error: 1" + e.toString());
        //networkError(e.toString());
    }
}

public void start()
{
    new Thread(this).start();
    try
    {
        //t.start();
    }
    catch (Exception e)
    {
        System.err.println("Error: 2" + e.toString());
        //networkError(e.toString());
    }
}

private void sendPost() throws IOException
{
    String FILE="file:///root1/"+Integer.toString
(title.hashCode())+".3gp";
    HttpURLConnection conn = null;
    OutputStream os = null;
    InputStream is = null;
    HttpURLConnection http = null;

    String username = rm.getRecord(1);
    //String username = "admin";
    try
    {
        // append the query string onto the URL
        //URL += "?title=" + title + "&text="+
text+"&gambar="+gambar;
        // replace not allowed char in the URL
        //URL = EncodeURL(URL);
        // establish the connection
        conn = (HttpURLConnection)Connector.open(URL);
        conn.setRequestMethod(HttpURLConnection.POST);
        //uploading process

        // the image is sent between the messages
ni the multipart message.

        //username
        String msg1="";
        msg1 += CrLf +
"-----4664151417711" + CrLf;
        msg1 += "Content-Type: text/plain"+ CrLf;
        //msg1 += "Content-Transfer-Encoding:
7bit"+ CrLf;
        msg1 += "Content-Disposition: inline;
name=\"username\""+CrLf;
        //msg1 += ""+CrLf+username;
        msg1 += ""+CrLf+"admin";

        //kategori
        String msg2="";
        msg2 += CrLf +
"-----4664151417711" + CrLf;
        msg2 += "Content-Type: multipart/form-

```

```

data"+ CrLf;
//msg2 += "Content-Transfer-Encoding:
7bit"+ CrLf;
msg2 += "Content-Disposition: inline;
name=\"kategori\""+CrLf;
msg2 += ""+CrLf+kategori;
//judul
String msg3="";
msg3 += CrLf +
"-----4664151417711" + CrLf;
msg3 += "Content-Type: multipart/form-
data"+ CrLf;
//msg3 += "Content-Transfer-Encoding:
7bit"+ CrLf;
msg3 += "Content-Disposition: inline;
name=\"judul\""+CrLf;
msg3 += ""+CrLf+title;
//isi
String msg4="";
msg4 += CrLf +
"-----4664151417711" + CrLf;
msg4 += "Content-Type: multipart/form-
data"+ CrLf;
//msg3 += "Content-Transfer-Encoding:
7bit"+ CrLf;
msg4 += "Content-Disposition: inline;
name=\"isi\""+CrLf;
msg4 += ""+CrLf;

String msg5 = "";
msg5 += CrLf
+"-----4664151417711" + CrLf;
msg5 += "Content-Disposition: form-data;
name=\"filevideo\"; filename=\"\" +FILE + "\"\" + CrLf;
msg5 += "Content-Type: image/jpeg" +
CrLf;
msg5 += CrLf;

String msg6 = "";
msg6 += CrLf +
"-----4664151417711--" + CrLf;

conn.setRequestProperty("Content-Type",
"multipart/form-data; boundary=-----
4664151417711");
// might not need to specify the content-
length when sending chunked data.
// conn.setRequestProperty("Content-
Length", String.valueOf((message1.length() + message2.length() +
imgData.length)));

System.out.println("open os");
os = conn.getOutputStream();

//write username
System.out.println(msg1);
os.write(msg1.getBytes());
//write kategori
System.out.println(msg2);
os.write(msg2.getBytes());
//write judul
System.out.println(msg3);

```

```

os.write(msg3.getBytes());
//write isi
System.out.println(msg4);
os.write(msg4.getBytes());
os.write(text);
/*
System.out.println(msg6);
os.write(msg6.getBytes());
*/

// SEND THE IMAGE

//write image
System.out.println(msg5);
os.write(msg5.getBytes());

int index = 0;
int size = 1024;
do{
    System.out.println("write:" + index);
    if((index+size)>byteVideo.length){
        size = byteVideo.length - index;
    }
    os.write(byteVideo, index, size);
    index+=size;
    //progress(byteVideo.length, index);
}while(index<byteVideo.length);
System.out.println("written:" + index);

System.out.println(msg6);
os.write(msg6.getBytes());

//System.out.println(String.valueOf
(imgData.length));
//os.write(("ini data").getBytes());
//os.write(imgData);

//String message3 = "username";
//os.write(message3.getBytes());
os.flush();

System.out.println("open is");
is = conn.openInputStream();

char buff = 512;
int len;
byte []data = new byte[buff];
do{
    System.out.println("READ");
    len = is.read(data);

    if(len > 0){
        System.out.println(new String
(data, 0, len));

```

```

        }
        }while(len>0);

        System.out.println("DONE");

        // server response
        /*
        if (http.getResponseCode() ==
HttpConnection.HTTP_OK)
        {
            sb = new StringBuffer();
            int ch;
            is = http.openInputStream();
            while ((ch = is.read()) != -1)
                sb.append((char) ch);
        }
        else
        {
            System.out.println("Network error");
            networkError();
        }
        */
    }
    catch (Exception e)
    {
        System.err.println("Error:3 " + e.toString());
        //networkError(e.toString());
    }
    finally
    {
        //if (is != null)
            is.close();
        //if (sb != null)
            //result = sb.toString();
        //else
            //networkError();
        //if (http != null)
            //http.close();
    }

    if (result != "")
    {
        //satu = result.getBytes();
        //satu.
        System.out.println(result);
        waktu2=System.currentTimeMillis()-midlet.waktu;
        midlet.showInfo("upload
berhasil...",getCurrentDisplay());
        //midlet.showInfo(result,getCurrentDisplay());
    }

        //midlet.showInfo(result, getCurrentDisplay());
    else
    {
        midlet.showInfo(result,getCurrentDisplay());
        //networkError();
    }
}

private Displayable getCurrentDisplay()
{
    // get the current display

```



```
        Displayable d = midlet.getCurrentDisplay();
        // if it is an Alert set the next display as the previous
one
        if (d.getClass().getName().equals
("javax.microedition.lcdui.Alert"))
        {
            d = previousDisplay;
        }
        return d;
    }
}
```



```

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package midletMCM;

import java.io.IOException;
import java.io.InputStream;

import java.io.OutputStream;
import javax.microedition.io.Connector;
import javax.microedition.io.HttpConnection;
import javax.microedition.lcdui.Command;
import javax.microedition.lcdui.Display;
import javax.microedition.lcdui.Displayable;
import javax.microedition.lcdui.Form;

/**
 *
 * @author ajib
 */
class PostSender5 implements Runnable
{
    private final String CrLf = "\r\n";
    RecordManager rm = new RecordManager("LOGIN");
    private Command exit;
    private Form mForm_3;
    private String dirGambar, title;
    private byte[] text;
    //private final String URL = rm.getRecord(6);
    private final String URL = "http://localhost/asa/post.php";
    //private PostingMidlet midlet;
    private MCMidlet midlet;
    private Displayable previousDisplay;
    private String result,kategori;
    private Display display;
    private String cek;
    byte[] byteGambar, byteAudio;
    private long waktu2;

    public PostSender5(MCMidlet midlet, byte[] byteGambar,byte[]
byteAudio, String title, byte[] text, String kategori)
    {

        this.midlet =midlet;
        this.byteAudio=byteAudio;
        this.byteGambar=byteGambar;
        this.title = title;
        this.text = text;
        this.kategori=kategori;

        //this.dirGambar=dirGambar;
    }

    public void run()
    {
        try
        {
            sendPost();
        }
        catch (Exception e)
        {

```

```

        e.printStackTrace();
        System.err.println("Error: 1" + e.toString());
        //networkError(e.toString());
    }
}

public void start()
{
    new Thread(this).start();
    try
    {
        //t.start();
    }
    catch (Exception e)
    {
        System.err.println("Error: 2" + e.toString());
        //networkError(e.toString());
    }
}

private void sendPost() throws IOException
{
    String FILE1="file:///root1/"+Integer.toString
(title.hashCode())+".jpg";
    String FILE2="file:///root1/"+Integer.toString
(title.hashCode())+".wav";
    HttpURLConnection conn = null;
    OutputStream os = null;
    InputStream is = null;
    HttpURLConnection http = null;

    String username = rm.getRecord(1);
    //String username = "admin";
    try
    {
        // append the query string onto the URL
        //URL += "?title=" + title + "&text="+
text+"&gambar="+gambar;
        // replace not allowed char in the URL
        //URL = EncodeURL(URL);
        // establish the connection
        conn = (HttpURLConnection)Connector.open(URL);
        conn.setRequestMethod(HttpURLConnection.POST);
        //uploading process

        // the image is sent between the messages
ni the multipart message.

        //username
String msg1="";
msg1 += CrLf +
"-----4664151417711" + CrLf;
msg1 += "Content-Type: text/plain"+ CrLf;
//msg1 += "Content-Transfer-Encoding:
7bit"+ CrLf;
msg1 += "Content-Disposition: inline;
name=\"username\""+CrLf;
msg1 += ""+CrLf+username;

        //kategori
String msg2="";
msg2 += CrLf +

```

```

"-----4664151417711" + CrLf;
msg2 += "Content-Type: multipart/form-
data"+ CrLf;
//msg2 += "Content-Transfer-Encoding:
7bit"+ CrLf;
msg2 += "Content-Disposition: inline;
name=\"kategori\""+CrLf;
msg2 += ""+CrLf+kategori;
//judul
String msg3="";
msg3 += CrLf +
"-----4664151417711" + CrLf;
msg3 += "Content-Type: multipart/form-
data"+ CrLf;
//msg3 += "Content-Transfer-Encoding:
7bit"+ CrLf;
msg3 += "Content-Disposition: inline;
name=\"judul\""+CrLf;
msg3 += ""+CrLf+title;
//isi
String msg4="";
msg4 += CrLf +
"-----4664151417711" + CrLf;
msg4 += "Content-Type: multipart/form-
data"+ CrLf;
//msg3 += "Content-Transfer-Encoding:
7bit"+ CrLf;
msg4 += "Content-Disposition: inline;
name=\"isi\""+CrLf;
msg4 += ""+CrLf;
String msg5 = "";
msg5 += CrLf
+"-----4664151417711" + CrLf;
msg5 += "Content-Disposition: form-data;
name=\"filegambar\"; filename=\"\" +FILE1 + "\"\" + CrLf;
msg5 += "Content-Type: image/jpeg" +
CrLf;
msg5 += CrLf;
String msg6 = "";
msg6 += CrLf
+"-----4664151417711" + CrLf;
msg6 += "Content-Disposition: form-data;
name=\"filesuara\"; filename=\"\" +FILE2 + "\"\" + CrLf;
msg6 += "Content-Type: image/jpeg" +
CrLf;
msg6 += CrLf;
String msg7 = "";
msg7 += CrLf +
"-----4664151417711--" + CrLf;

conn.setRequestProperty("Content-Type",
"multipart/form-data; boundary=-----
4664151417711");
// might not need to specify the content-
length when sending chunked data.
// conn.setRequestProperty("Content-
Length", String.valueOf((message1.length() + message2.length() +
imgData.length)));

System.out.println("open os");
os = conn.openOutputStream();

```

```

//write username
System.out.println(msg1);
os.write(msg1.getBytes());
//write kategori
System.out.println(msg2);
os.write(msg2.getBytes());
//write judul
System.out.println(msg3);
os.write(msg3.getBytes());
//write isi
System.out.println(msg4);
os.write(msg4.getBytes());
os.write(text);
/*
System.out.println(msg6);
os.write(msg6.getBytes());
*/

// SEND THE IMAGE

//write image
System.out.println(msg5);
os.write(msg5.getBytes());

int index = 0;
int size = 1024;
do{
    System.out.println("write:" + index);
    if((index+size)>byteGambar.length){
        size = byteGambar.length - index;
    }
    os.write(byteGambar, index, size);
    index+=size;
    //progress(byteVideo.length, index);
// update the progress bar.

}while(index<byteGambar.length);
System.out.println("written:" + index);

System.out.println(msg6);
os.write(msg6.getBytes());
int index2 = 0;
int size2 = 1024;
do{
    System.out.println("write:" +
index2);
    if((index2+size2)>byteAudio.length){
        size2 = byteAudio.length -
index2;
    }
    os.write(byteAudio, index2, size2);
    index2+=size2;
    //progress(byteVideo.length, index);
// update the progress bar.

}while(index2<byteAudio.length);
System.out.println("written:" + index2);
System.out.println(msg7);
os.write(msg7.getBytes());

```

```

        (imgData.length));

        //System.out.println(String.valueOf
        //os.write(("ini data").getBytes());
        //os.write(imgData);

        //String message3 = "username";
        //os.write(message3.getBytes());
        os.flush();

        System.out.println("open is");
        is = conn.openInputStream();

        char buff = 512;
        int len;
        byte []data = new byte[buff];
        do{
            System.out.println("READ");
            len = is.read(data);

            if(len > 0){
                System.out.println(new String
                (data, 0, len));
            }
        }while(len>0);

        System.out.println("DONE");

        // server response
        /*
        if (http.getResponseCode() ==
        HttpURLConnection.HTTP_OK)
        {
            sb = new StringBuffer();
            int ch;
            is = http.openInputStream();
            while ((ch = is.read()) != -1)
                sb.append((char) ch);
        }
        else
        {
            System.out.println("Network error");
            networkError();
        }
        */

    }
    catch (Exception e)
    {
        System.err.println("Error:3 " + e.toString());
        //networkError(e.toString());
    }
    finally
    {
        //if (is != null)
        is.close();
        //if (sb != null)
        //result = sb.toString();
        //else
        //networkError();
        //if (http != null)

```

```

        //http.close();
    }

    if (result != "")
    {
        //satu = result.getBytes();
        //satu.
        System.out.println(result);
        waktu2=System.currentTimeMillis()-midlet.waktu;
        midlet.showInfo("upload
berhasil...",getCurrentDisplay());
        //midlet.showInfo(result,getCurrentDisplay());
    }

    else
    {
        midlet.showInfo(result,getCurrentDisplay());
        //networkError();
    }
}

private Displayable getCurrentDisplay()
{
    // get the current display
    Displayable d = midlet.getCurrentDisplay();
    // if it is an Alert set the next display as the previous
one
    if (d.getClass().getName().equals
("javax.microedition.lcdui.Alert"))
    {
        d = previousDisplay;
    }
    return d;
}
}
}

```

```

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package midletMCM;

import java.io.IOException;
import java.io.InputStream;

import java.io.OutputStream;
import javax.microedition.io.Connector;
import javax.microedition.io.HttpConnection;
import javax.microedition.lcdui.Command;
import javax.microedition.lcdui.Display;
import javax.microedition.lcdui.Displayable;
import javax.microedition.lcdui.Form;

/**
 *
 * @author ajib
 */
class PostSender6 implements Runnable
{
    private final String CrLf = "\r\n";
    RecordManager rm = new RecordManager("LOGIN");
    private Command exit;
    private Form mForm_3;
    private String dirGambar, title;
    private byte[] text;
    //private final String URL = rm.getRecord(6);
    private final String URL = "http://localhost/asa/post.php";
    //private PostingMidlet midlet;
    private MCMMidlet midlet;
    private Displayable previousDisplay;
    private String result,kategori;
    private Display display;
    private String cek;
    byte[] byteGambar, byteVideo;
    private long waktu2;

    public PostSender6(MCMMidlet midlet, byte[] byteGambar,byte[]
byteVideo, String title, byte[] text, String kategori)
    {

        this.midlet =midlet;
        this.byteVideo=byteVideo;
        this.byteGambar=byteGambar;
        this.title = title;
        this.text = text;
        this.kategori=kategori;

        //this.dirGambar=dirGambar;
    }

    public void run()
    {
        try
        {
            sendPost();
        }
        catch (Exception e)
        {

```



```

        e.printStackTrace();
        System.err.println("Error: 1" + e.toString());
        //networkError(e.toString());
    }
}

public void start()
{
    new Thread(this).start();
    try
    {
        //t.start();
    }
    catch (Exception e)
    {
        System.err.println("Error: 2" + e.toString());
        //networkError(e.toString());
    }
}

private void sendPost() throws IOException
{
    String FILE1="file:///root1/"+Integer.toString
(title.hashCode())+".jpg";
    String FILE2="file:///root1/"+Integer.toString
(title.hashCode())+".3gp";
    HttpURLConnection conn = null;
    OutputStream os = null;
    InputStream is = null;
    HttpURLConnection http = null;

    String username = rm.getRecord(1);
    //String username = "admin";
    try
    {
        // append the query string onto the URL
        //URL += "?title=" + title + "&text="+
text+"&gambar="+gambar;
        // replace not allowed char in the URL
        //URL = EncodeURL(URL);
        // establish the connection
        conn = (HttpURLConnection)Connector.open(URL);
        conn.setRequestMethod(HttpURLConnection.POST);
        //uploading process

        // the image is sent between the messages
ni the multipart message.

        //username
        String msg1="";
        msg1 += CrLf +
"-----4664151417711" + CrLf;
        msg1 += "Content-Type: text/plain"+ CrLf;
        //msg1 += "Content-Transfer-Encoding:
7bit"+ CrLf;
        msg1 += "Content-Disposition: inline;
name=\"username\""+CrLf;
        msg1 += ""+CrLf+username;

        //kategori
        String msg2="";
        msg2 += CrLf +

```

```

"-----4664151417711" + CrLf;
msg2 += "Content-Type: multipart/form-
data"+ CrLf;
//msg2 += "Content-Transfer-Encoding:
7bit"+ CrLf;
msg2 += "Content-Disposition: inline;
name=\"kategori\""+CrLf;
msg2 += ""+CrLf+kategori;
//judul
String msg3="";
msg3 += CrLf +
"-----4664151417711" + CrLf;
msg3 += "Content-Type: multipart/form-
data"+ CrLf;
//msg3 += "Content-Transfer-Encoding:
7bit"+ CrLf;
msg3 += "Content-Disposition: inline;
name=\"judul\""+CrLf;
msg3 += ""+CrLf+title;
//isi
String msg4="";
msg4 += CrLf +
"-----4664151417711" + CrLf;
msg4 += "Content-Type: multipart/form-
data"+ CrLf;
//msg3 += "Content-Transfer-Encoding:
7bit"+ CrLf;
msg4 += "Content-Disposition: inline;
name=\"isi\""+CrLf;
msg4 += ""+CrLf;
String msg5 = "";
msg5 += CrLf
+"-----4664151417711" + CrLf;
msg5 += "Content-Disposition: form-data;
name=\"filegambar\"; filename=\"\" +FILE1 + "\"\" + CrLf;
msg5 += "Content-Type: image/jpeg" +
CrLf;
msg5 += CrLf;
String msg6 = "";
msg6 += CrLf
+"-----4664151417711" + CrLf;
msg6 += "Content-Disposition: form-data;
name=\"filevideo\"; filename=\"\" +FILE2 + "\"\" + CrLf;
msg6 += "Content-Type: image/jpeg" +
CrLf;
msg6 += CrLf;
String msg7 = "";
msg7 += CrLf +
"-----4664151417711--" + CrLf;

conn.setRequestProperty("Content-Type",
"multipart/form-data; boundary=-----
4664151417711");
// might not need to specify the content-
length when sending chunked data.
// conn.setRequestProperty("Content-
Length", String.valueOf((message1.length() + message2.length() +
imgData.length)));

System.out.println("open os");
os = conn.openOutputStream();

```

```

//write username
System.out.println(msg1);
os.write(msg1.getBytes());
//write kategori
System.out.println(msg2);
os.write(msg2.getBytes());
//write judul
System.out.println(msg3);
os.write(msg3.getBytes());
//write isi
System.out.println(msg4);
os.write(msg4.getBytes());
os.write(text);
/*
System.out.println(msg6);
os.write(msg6.getBytes());
*/

// SEND THE IMAGE

//write image
System.out.println(msg5);
os.write(msg5.getBytes());

int index = 0;
int size = 1024;
do{
    System.out.println("write:" + index);
    if((index+size)>byteGambar.length){
        size = byteGambar.length - index;
    }
    os.write(byteGambar, index, size);
    index+=size;
    //progress(byteVideo.length, index);
// update the progress bar.

}while(index<byteGambar.length);
System.out.println("written:" + index);

System.out.println(msg6);
os.write(msg6.getBytes());
int index2 = 0;
int size2 = 1024;
do{
    System.out.println("write:" +
index2);
    if((index2+size2)>byteVideo.length){
        size2 = byteVideo.length -
index2;
    }
    os.write(byteVideo, index2, size2);
    index2+=size2;
    //progress(byteVideo.length, index);
// update the progress bar.

}while(index2<byteVideo.length);
System.out.println("written:" + index2);
System.out.println(msg7);
os.write(msg7.getBytes());

```

```

        (imgData.length));

        //System.out.println(String.valueOf
        //os.write(("ini data").getBytes());
        //os.write(imgData);

        //String message3 = "username";
        //os.write(message3.getBytes());
        os.flush();

        System.out.println("open is");
        is = conn.openInputStream();

        char buff = 512;
        int len;
        byte []data = new byte[buff];
        do{
            System.out.println("READ");
            len = is.read(data);

            if(len > 0){
                System.out.println(new String
                (data, 0, len));
            }
        }while(len>0);

        System.out.println("DONE");

        // server response
        /*
        if (http.getResponseCode() ==
        HttpURLConnection.HTTP_OK)
        {
            sb = new StringBuffer();
            int ch;
            is = http.openInputStream();
            while ((ch = is.read()) != -1)
                sb.append((char) ch);
        }
        else
        {
            System.out.println("Network error");
            networkError();
        }
        */

    }
    catch (Exception e)
    {
        System.err.println("Error:3 " + e.toString());
        //networkError(e.toString());
    }
    finally
    {
        //if (is != null)
        is.close();
        //if (sb != null)
        //result = sb.toString();
        //else
        //networkError();
        //if (http != null)

```

```

        //http.close();
    }

    if (result != "")
    {
        //satu = result.getBytes();
        //satu.
        System.out.println(result);
        waktu2=System.currentTimeMillis()-midlet.waktu;
        midlet.showInfo("upload
berhasil...",getCurrentDisplay());
        //midlet.showInfo(result,getCurrentDisplay());
    }

        //midlet.showInfo(result, getCurrentDisplay());

    else
    {
        midlet.showInfo(result,getCurrentDisplay());
        //networkError();
    }
}

private Displayable getCurrentDisplay()
{
    // get the current display
    Displayable d = midlet.getCurrentDisplay();
    // if it is an Alert set the next display as the previous
one
    if (d.getClass().getName().equals
("javax.microedition.lcdui.Alert"))
    {
        d = previousDisplay;
    }
    return d;
}

}

```

```

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package midletMCM;

import java.io.IOException;
import java.io.InputStream;

import java.io.OutputStream;
import javax.microedition.io.Connector;
import javax.microedition.io.HttpConnection;
import javax.microedition.lcdui.Command;
import javax.microedition.lcdui.Display;
import javax.microedition.lcdui.Displayable;
import javax.microedition.lcdui.Form;

/**
 *
 * @author ajib
 */
class PostSender7 implements Runnable
{
    private final String CrLf = "\r\n";
    RecordManager rm = new RecordManager("LOGIN");
    private Command exit;
    private Form mForm_3;
    private String dirGambar, title;
    private byte[] text;
    //private final String URL = rm.getRecord(6);
    private final String URL = "http://localhost/asa/post.php";
    //private PostingMidlet midlet;
    private MCMMidlet midlet;
    private Displayable previousDisplay;
    private String result,kategori;
    private Display display;
    private String cek;
    byte[] byteAudio, byteVideo;
    private long waktu2;

    public PostSender7(MCMMidlet midlet, byte[] byteAudio,byte[]
byteVideo, String title, byte[] text, String kategori)
    {

        this.midlet =midlet;
        this.byteVideo=byteVideo;
        this.byteAudio=byteAudio;
        this.title = title;
        this.text = text;
        this.kategori=kategori;

        //this.dirGambar=dirGambar;
    }

    public void run()
    {
        try
        {
            sendPost();
        }
        catch (Exception e)
        {

```

```

        e.printStackTrace();
        System.err.println("Error: 1" + e.toString());
        //networkError(e.toString());
    }
}

public void start()
{
    new Thread(this).start();
    try
    {
        //t.start();
    }
    catch (Exception e)
    {
        System.err.println("Error: 2" + e.toString());
        //networkError(e.toString());
    }
}

private void sendPost() throws IOException
{
    String FILE1="file:///root1/"+Integer.toString
(title.hashCode())+".wav";
    String FILE2="file:///root1/"+Integer.toString
(title.hashCode())+".3gp";
    HttpURLConnection conn = null;
    OutputStream os = null;
    InputStream is = null;
    HttpURLConnection http = null;

    String username = rm.getRecord(1);
    //String username = "admin";
    try
    {
        // append the query string onto the URL
        //URL += "?title=" + title + "&text="+
text+"&gambar="+gambar;
        // replace not allowed char in the URL
        //URL = EncodeURL(URL);
        // establish the connection
        conn = (HttpURLConnection)Connector.open(URL);
        conn.setRequestMethod(HttpURLConnection.POST);
        //uploading process

        // the image is sent between the messages
ni the multipart message.

        //username
        String msg1="";
        msg1 += CrLf +
"-----4664151417711" + CrLf;
        msg1 += "Content-Type: text/plain"+ CrLf;
        //msg1 += "Content-Transfer-Encoding:
7bit"+ CrLf;
        msg1 += "Content-Disposition: inline;
name=\"username\""+CrLf;
        msg1 += ""+CrLf+username;

        //kategori
        String msg2="";
        msg2 += CrLf +

```

```

"-----4664151417711" + CrLf;
msg2 += "Content-Type: multipart/form-
data"+ CrLf;
//msg2 += "Content-Transfer-Encoding:
7bit"+ CrLf;
msg2 += "Content-Disposition: inline;
name=\"kategori\""+CrLf;
msg2 += ""+CrLf+kategori;
//judul
String msg3="";
msg3 += CrLf +
"-----4664151417711" + CrLf;
msg3 += "Content-Type: multipart/form-
data"+ CrLf;
//msg3 += "Content-Transfer-Encoding:
7bit"+ CrLf;
msg3 += "Content-Disposition: inline;
name=\"judul\""+CrLf;
msg3 += ""+CrLf+title;
//isi
String msg4="";
msg4 += CrLf +
"-----4664151417711" + CrLf;
msg4 += "Content-Type: multipart/form-
data"+ CrLf;
//msg3 += "Content-Transfer-Encoding:
7bit"+ CrLf;
msg4 += "Content-Disposition: inline;
name=\"isi\""+CrLf;
msg4 += ""+CrLf;
String msg5 = "";
msg5 += CrLf
+"-----4664151417711" + CrLf;
msg5 += "Content-Disposition: form-data;
name=\"filesuara\"; filename=\"\" +FILE1 + "\"\" + CrLf;
msg5 += "Content-Type: image/jpeg" +
CrLf;
msg5 += CrLf;
String msg6 = "";
msg6 += CrLf
+"-----4664151417711" + CrLf;
msg6 += "Content-Disposition: form-data;
name=\"filevideo\"; filename=\"\" +FILE2 + "\"\" + CrLf;
msg6 += "Content-Type: image/jpeg" +
CrLf;
msg6 += CrLf;
String msg7 = "";
msg7 += CrLf +
"-----4664151417711--" + CrLf;

conn.setRequestProperty("Content-Type",
"multipart/form-data; boundary=-----
4664151417711");
// might not need to specify the content-
length when sending chunked data.
// conn.setRequestProperty("Content-
Length", String.valueOf((message1.length() + message2.length() +
imgData.length)));

System.out.println("open os");
os = conn.openOutputStream();

```



```

//write username
System.out.println(msg1);
os.write(msg1.getBytes());
//write kategori
System.out.println(msg2);
os.write(msg2.getBytes());
//write judul
System.out.println(msg3);
os.write(msg3.getBytes());
//write isi
System.out.println(msg4);
os.write(msg4.getBytes());
os.write(text);
/*
System.out.println(msg6);
os.write(msg6.getBytes());
*/

//write image
System.out.println(msg5);
os.write(msg5.getBytes());

int index = 0;
int size = 1024;
do{
    System.out.println("write:" + index);
    if((index+size)>byteAudio.length){
        size = byteAudio.length - index;
    }
    os.write(byteAudio, index, size);
    index+=size;
    //progress(byteVideo.length, index);
// update the progress bar.

}while(index<byteAudio.length);
System.out.println("written:" + index);

System.out.println(msg6);
os.write(msg6.getBytes());

int index2 = 0;
int size2 = 1024;
do{
    System.out.println("write:" +
index2);
    if((index2+size2)>byteVideo.length){
        size2 = byteVideo.length -
index2;
    }
    os.write(byteVideo, index2, size2);
    index2+=size2;
    //progress(byteVideo.length, index);
// update the progress bar.

}while(index2<byteVideo.length);
System.out.println("written:" + index2);
System.out.println(msg7);

```

```

os.write(msg7.getBytes());

//System.out.println(String.valueOf
(imgData.length));
//os.write(("ini data").getBytes());
//os.write(imgData);

//String message3 = "username";
//os.write(message3.getBytes());
os.flush();

System.out.println("open is");
is = conn.openInputStream();

char buff = 512;
int len;
byte []data = new byte[buff];
do{
    System.out.println("READ");
    len = is.read(data);

    if(len > 0){
        System.out.println(new String
(data, 0, len));
    }
}while(len>0);

System.out.println("DONE");

// server response
/*
if (http.getResponseCode() ==
HttpURLConnection.HTTP_OK)
{
    sb = new StringBuffer();
    int ch;
    is = http.openInputStream();
    while ((ch = is.read()) != -1)
        sb.append((char) ch);
}
else
{
    System.out.println("Network error");
    networkError();
}
*/
}
catch (Exception e)
{
    System.err.println("Error:3 " + e.toString());
    //networkError(e.toString());
}
finally
{
    //if (is != null)
    is.close();
    //if (sb != null)
    //result = sb.toString();
    //else
    //networkError();
}

```

```

        //if (http != null)
            //http.close();
    }

    if (result != "")
    {
        //satu = result.getBytes();
        //satu.
        System.out.println(result);
        waktu2=System.currentTimeMillis()-midlet.waktu;
        midlet.showInfo("upload
berhasil...",getCurrentDisplay());
        //midlet.showInfo(result,getCurrentDisplay());
    }

    else
    {
        midlet.showInfo(result,getCurrentDisplay());
        //networkError();
    }
}

private Displayable getCurrentDisplay()
{
    // get the current display
    Displayable d = midlet.getCurrentDisplay();
    // if it is an Alert set the next display as the previous
one
    if (d.getClass().getName().equals
("javax.microedition.lcdui.Alert"))
    {
        d = previousDisplay;
    }
    return d;
}
}
}

```

```

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package midletMCM;

import java.io.IOException;
import java.io.InputStream;

import java.io.OutputStream;
import javax.microedition.io.Connector;
import javax.microedition.io.HttpConnection;
import javax.microedition.lcdui.Command;
import javax.microedition.lcdui.Display;
import javax.microedition.lcdui.Displayable;
import javax.microedition.lcdui.Form;

/**
 *
 * @author ajib
 */
class PostSender8 implements Runnable
{
    private final String CrLf = "\r\n";
    RecordManager rm = new RecordManager("LOGIN");
    private Command exit;
    private Form mForm_3;
    private String dirGambar, title;
    private byte[] text;
    //private final String URL = rm.getRecord(6);
    private final String URL = "http://localhost/asa/post.php";
    //private PostingMidlet midlet;
    private MCMMidlet midlet;
    private Displayable previousDisplay;
    private String result,kategori;
    private Display display;
    private String cek;
    byte[] byteAudio, byteVideo, byteGambar;
    private long waktu2;

    public PostSender8(MCMMidlet midlet, byte[] byteGambar,byte[]
byteAudio,byte[] byteVideo, String title, byte[] text, String
kategori)
    {
        this.midlet =midlet;
        this.byteGambar=byteGambar;
        this.byteVideo=byteVideo;
        this.byteAudio=byteAudio;
        this.title = title;
        this.text = text;
        this.kategori=kategori;
        //this.dirGambar=dirGambar;
    }

    public void run()
    {
        try
        {
            sendPost();
        }
    }
}

```

```

        catch (Exception e)
        {
            e.printStackTrace();
            System.err.println("Error: 1" + e.toString());
            //networkError(e.toString());
        }
    }

    public void start()
    {
        new Thread(this).start();
        try
        {
            //t.start();
        }
        catch (Exception e)
        {
            System.err.println("Error: 2" + e.toString());
            //networkError(e.toString());
        }
    }

    private void sendPost() throws IOException
    {
        String FILE1="file:///root1/"+Integer.toString
(title.hashCode())+".jpg";
        String FILE2="file:///root1/"+Integer.toString
(title.hashCode())+".wav";
        String FILE3="file:///root1/"+Integer.toString
(title.hashCode())+".3gp";
        HttpURLConnection conn = null;
        OutputStream os = null;
        InputStream is = null;
        HttpURLConnection http = null;

        String username = rm.getRecord(1);
        //String username = "admin";
        try
        {
            // append the query string onto the URL
            //URL += "?title=" + title + "&text="+
text+"&gambar="+gambar;
            // replace not allowed char in the URL
            //URL = EncodeURL(URL);
            // establish the connection
            conn = (HttpURLConnection)Connector.open(URL);
            conn.setRequestMethod(HttpURLConnection.POST);
            //uploading process

            // the image is sent between the messages
ni the multipart message.

            //username
            String msg1="";
            msg1 += CrLf +
"-----4664151417711" + CrLf;
            msg1 += "Content-Type: text/plain"+ CrLf;
            //msg1 += "Content-Transfer-Encoding:
7bit"+ CrLf;
            msg1 += "Content-Disposition: inline;
name=\"username\""+CrLf;
            msg1 += ""+CrLf+username;

```

```

//kategori
String msg2="";
msg2 += CrLf +
"-----4664151417711" + CrLf;
data"+ CrLf;
msg2 += "Content-Type: multipart/form-
//msg2 += "Content-Transfer-Encoding:
7bit"+ CrLf;
msg2 += "Content-Disposition: inline;
name=\"kategori\""+CrLf;
msg2 += ""+CrLf+kategori;
//judul
String msg3="";
msg3 += CrLf +
"-----4664151417711" + CrLf;
data"+ CrLf;
msg3 += "Content-Type: multipart/form-
//msg3 += "Content-Transfer-Encoding:
7bit"+ CrLf;
msg3 += "Content-Disposition: inline;
name=\"judul\""+CrLf;
msg3 += ""+CrLf+title;
//isi
String msg4="";
msg4 += CrLf +
"-----4664151417711" + CrLf;
data"+ CrLf;
msg4 += "Content-Type: multipart/form-
//msg3 += "Content-Transfer-Encoding:
7bit"+ CrLf;
msg4 += "Content-Disposition: inline;
name=\"isi\""+CrLf;
msg4 += ""+CrLf;
String msg5 = "";
msg5 += CrLf
+"-----4664151417711" + CrLf;
name=\"filegambar\"; filename=\"\" +FILE1 + "\"\" + CrLf;
msg5 += "Content-Disposition: form-data;
msg5 += "Content-Type: image/jpeg" +
CrLf;
msg5 += CrLf;
String msg6 = "";
msg6 += CrLf
+"-----4664151417711" + CrLf;
name=\"filesuara\"; filename=\"\" +FILE2 + "\"\" + CrLf;
msg6 += "Content-Disposition: form-data;
msg6 += "Content-Type: image/jpeg" +
CrLf;
msg6 += CrLf;
String msg7 = "";
msg7 += CrLf
+"-----4664151417711" + CrLf;
name=\"filevideo\"; filename=\"\" +FILE3 + "\"\" + CrLf;
msg7 += "Content-Disposition: form-data;
msg7 += "Content-Type: image/jpeg" +
CrLf;
msg7 += CrLf;
String msg8 = "";
msg8 += CrLf +
"-----4664151417711--" + CrLf;

```

```

        conn.setRequestProperty("Content-Type",
"multipart/form-data; boundary=-----
4664151417711");

        // might not need to specify the content-
length when sending chunked data.
        // conn.setRequestProperty("Content-
Length", String.valueOf((message1.length() + message2.length() +
imgData.length)));

        System.out.println("open os");
        os = conn.getOutputStream();

        //write username
        System.out.println(msg1);
        os.write(msg1.getBytes());
        //write kategori
        System.out.println(msg2);
        os.write(msg2.getBytes());
        //write judul
        System.out.println(msg3);
        os.write(msg3.getBytes());
        //write isi
        System.out.println(msg4);
        os.write(msg4.getBytes());
        os.write(text);
        /*
        System.out.println(msg6);
        os.write(msg6.getBytes());
        */

        //write image
        System.out.println(msg5);
        os.write(msg5.getBytes());

        int index = 0;
        int size = 1024;
        do{
            System.out.println("write:" + index);
            if((index+size)>byteGambar.length){
                size = byteGambar.length - index;
            }
            os.write(byteGambar, index, size);
            index+=size;
            //progress(byteGambar.length, index);

// update the progress bar.

        }while(index<byteGambar.length);
        System.out.println("written:" + index);

        System.out.println(msg6);
        os.write(msg6.getBytes());

        int index2 = 0;
        int size2 = 1024;
        do{
            System.out.println("write:" +
index2);

```

```

        if((index2+size2)>byteAudio.length){
            size2 = byteAudio.length -
index2;
        }
        os.write(byteAudio, index2, size2);
        index2+=size2;
        //progress(byteVideo.length, index);
// update the progress bar.
    }while(index2<byteAudio.length);
    System.out.println("written:" + index2);

    System.out.println(msg7);
    os.write(msg7.getBytes());

    int index3 = 0;
    int size3 = 1024;
    do{
        System.out.println("write:" +
index3);
        if((index3+size3)>byteVideo.length){
            size3 = byteVideo.length -
index3;
        }
        os.write(byteVideo, index3, size3);
        index3+=size3;
        //progress(byteVideo.length, index);
// update the progress bar.

    }while(index3<byteVideo.length);
    System.out.println("written:" + index3);

    System.out.println(msg8);
    os.write(msg8.getBytes());

    //System.out.println(String.valueOf
(indexData.length));
    //os.write(("ini data").getBytes());
    //os.write(imgData);

    //String message3 = "username";
    //os.write(message3.getBytes());
    os.flush();

    System.out.println("open is");
    is = conn.openInputStream();

    char buff = 512;
    int len;
    byte []data = new byte[buff];
    do{
        System.out.println("READ");
        len = is.read(data);

        if(len > 0){
            System.out.println(new String
(indexData, 0, len));
        }
    }while(len>0);

    System.out.println("DONE");

```



```

        // server response
        /*
        if (http.getResponseCode() ==
HttpConnection.HTTP_OK)
        {
            sb = new StringBuffer();
            int ch;
            is = http.openInputStream();
            while ((ch = is.read()) != -1)
                sb.append((char) ch);
        }
        else
        {
            System.out.println("Network error");
            networkError();
        }
        */
    }
    catch (Exception e)
    {
        System.err.println("Error:3 " + e.toString());
        //networkError(e.toString());
    }
    finally
    {
        //if (is != null)
        is.close();
        //if (sb != null)
        //result = sb.toString();
        //else
        //networkError();
        //if (http != null)
        //http.close();
    }

    if (result != "")
    {
        //satu = result.getBytes();
        //satu.
        System.out.println(result);
        waktu2=System.currentTimeMillis()-midlet.waktu;
        midlet.showInfo("upload
berhasil...",getCurrentDisplay());
        //midlet.showInfo(result,getCurrentDisplay());
    }

        //midlet.showInfo(result, getCurrentDisplay());

    else
    {
        midlet.showInfo(result,getCurrentDisplay());
        //networkError();
    }
}

private Displayable getCurrentDisplay()
{
    // get the current display
    Displayable d = midlet.getCurrentDisplay();
    // if it is an Alert set the next display as the previous
one
    if (d.getClass().getName().equals
("javax.microedition.lcdui.Alert"))

```

```
    {  
        d = previousDisplay;  
    }  
    return d;  
}
```



```

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package midletMCM;

import java.io.IOException;
import java.io.InputStream;

import javax.microedition.io.Connector;
import javax.microedition.io.HttpConnection;
import javax.microedition.lcdui.Alert;
import javax.microedition.lcdui.AlertType;
import javax.microedition.lcdui.Command;
import javax.microedition.lcdui.Display;
import javax.microedition.lcdui.Displayable;
import javax.microedition.lcdui.Form;
import javax.microedition.lcdui.Image;

/**
 *
 * @author ajib
 */
class PostSenderRes implements Runnable
{
    //private MCMMidlet midlet;
    private String username, nama, nomor, email;
    private String password;
    private Command exit;
    private Form mForm_3;
    private String username1;
    private String password1;
    private String URL = "http://localhost/asa/respon.php";
    //private String URL =
    "http://www.ee.ui.ac.id/telkom/mcm2/respon.php";

    private MCMMidlet midlet;
    private Displayable previousDisplay;
    public String result;
    private Display display;
    //private PostingMidlet midlet;
    byte[] satu;
    int angka;
    private Image image;
    //private FormLogin fmLogin;
    //private Login login;

    public PostSenderRes(MCMMidlet midlet)
    {
        this.midlet = midlet;
    }

    public void run()
    {
        try
        {
            sendPost();
        }
        catch (Exception e)
        {
            e.printStackTrace();
            System.err.println("Error1: " + e.toString());
            networkError(e.toString());
        }
    }
}

```

```

public void start()
{
    Thread t = new Thread(this);
    try
    {
        t.start();
    }
    catch (Exception e)
    {
        System.err.println("Error2: " + e.toString());
        networkError(e.toString());
    }
}

private void sendPost() throws IOException
{
    InputStream is = null;
    StringBuffer sb = null;
    HttpURLConnection http = null;

    try
    {
        // append the query string onto the URL
        //URL += "?username=" + username1 + "&password="+
password1+"&nama="+ nama+"&nomor="+ nomor+"&email="+email;
        // replace not allowed char in the URL
        URL = EncodeURL(URL);
        // establish the connection
        http = (HttpURLConnection) Connector.open(URL);
        // set the request method as GET
        http.setRequestMethod(HttpURLConnection.GET);
        //http.setRequestProperty("Content-Type",
"text/plain");
        // server response
        int rc = http.getResponseCode();
        if (http.getResponseCode() ==
HttpURLConnection.HTTP_OK)
        {
            sb = new StringBuffer();
            int ch;
            is = http.openInputStream();
            int max = is.available();
            //midlet.gIn.setMaxValue(max);
            while ((ch = is.read()) != -1)
            {
                sb.append((char) ch);

                progress(max, sb.length());
            }
        }
        else
        {
            System.out.println("Network error sini" +
rc);
            networkError();
        }
    }
    catch (Exception e)
    {
        System.err.println("Error: " + e.toString());
        System.err.println("Error: salah2");
        networkError(e.toString());
    }
}

```

```

    }
    finally
    {
        if (is != null)
            is.close();
        if (sb != null)
            result = sb.toString();
        else
            networkError();
        if (http != null)
            http.close();
    }

    if (result != "")
    {
        System.out.println(result);
        //midlet.fmRespon.re
        //midlet.gIn.
        midlet.fmRespon.append(result);
    }
    else
    {
        System.out.println(result+"salah");
        networkError();
    }
}

private Displayable getCurrentDisplay()
{
    // get the current display
    Displayable d = midlet.getCurrentDisplay();
    // if it is an Alert set the next display as the previous
    one
    if (d.getClass().getName().equals
("javax.microedition.lcdui.Alert"))
    {
        d = previousDisplay;
    }
    return d;
}

private void progress(int total, int current)
{
    int percent = (int) (100 * ((float)current/(float)total));
    //midlet.gIn.setMaxValue(total);
    midlet.gIn.setValue(percent);
}

private String EncodeURL(String URL)
{
    URL = replace(URL, 'Ã ', "%E0");
    URL = replace(URL, 'Ã", "%E8");
    URL = replace(URL, 'Ã@', "%E9");
    URL = replace(URL, 'Ã-', "%EC");
    URL = replace(URL, 'Ã²', "%F2");
    URL = replace(URL, 'Ã¹', "%F9");
    URL = replace(URL, '$', "%24");
    URL = replace(URL, '#', "%23");
    URL = replace(URL, 'Â£', "%A3");
    URL = replace(URL, '@', "%40");
    URL = replace(URL, '\\', "%27");
}

```

```
        URL = replace(URL, ' ', "%20");
        return URL;
    }

    private String replace(String source, char oldChar, String
dest)
    {
        String ret = "";
        for (int i = 0; i < source.length(); i++)
        {
            if (source.charAt(i) != oldChar)
                ret += source.charAt(i);
            else
                ret += dest;
        }
        return ret;
    }

    private void networkError()
    {
        midlet.showInfo("Network error, try later",
getCurrentDisplay());
    }
    private void networkError(String msg)
    {
        midlet.showInfo(msg, getCurrentDisplay());
    }
}
```

```

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package midletMCM;

import javax.microedition.rms.RecordStore;

/**
 *
 * @author ajib
 */
class RecordManager {
    private String rsName;
    private RecordStore rs;

    //constructor
    public RecordManager(String rsName)
    {
        this.rsName = rsName;
    }

    //open a RS
    private void openRS()
    {
        try {
            rs = RecordStore.openRecordStore(rsName, true);
            //rs = RecordStore.
        }
        catch(Exception e)
        {
            System.err.println(e);
        }
    }

    //close a RS
    private void closeRS()
    {
        try {
            if(rs != null)
                rs.closeRecordStore();
        }
        catch(Exception e)
        {
            System.err.println(e);
        }
    }

    //add a record to the RS
    public void addRecord(String record)
    {
        openRS();

        byte[] rec = record.getBytes();
        try {
            rs.addRecord(rec, 0, rec.length);
        }
        catch(Exception e)
        {
            System.err.println(e);
        }
        finally
        {
            closeRS();
        }
    }
}

```

```

}

//replace a record in the RS
public void setRecord(int id, String record)
{
    openRS();

    byte[] rec = record.getBytes();
    try {
        rs.setRecord(id, rec, 0, rec.length);
    }
    catch(Exception e)
    {
        System.err.println(e);
    }
    finally
    {
        closeRS();
    }
}

//return a record of the RS
public String getRecord(int id)
{
    openRS();
    String ret = "";
    try {
        ret = new String(rs.getRecord(id));
    }
    catch(Exception e)
    {
        System.err.println(e);
    }
    finally
    {
        closeRS();
    }
    return ret;
}

//return the number of records stored in the RS
public int numRecords()
{
    openRS();
    int num = -1;
    try
    {
        num = rs.getNumRecords();
    }
    catch(Exception e)
    {
        System.err.println(e);
    }
    finally
    {
        closeRS();
    }
    return num;
}

//delete the RS
public void deleteRS()
{
    try {

```



```
        RecordStore.deleteRecordStore(rsName);  
    }  
    catch(Exception e)  
    {  
        System.err.println(e);  
    }  
}
```



```

body{
    font-family: Tahoma;
    text-align: center;
}
#header {
    position: relative;
    background-image: url(../admin/images/header.jpg);
    background-repeat: no-repeat;
    margin-right: auto;
    margin-left: auto;
    width: 780px;
    border: 2px solid #006600;
    padding-top: 70px;
    text-align: left;
}
#content {
    margin-left: 230px;
    padding: 20px 10px 0 0;
}
#content p {
    font-size: 80%;
    line-height: 1.8em;
    padding-left: 2em;
}
#menu {
    position: absolute;
    top: 90px;
    left: 0;
    width: 180px;
}
#menu ul {
    list-style: none;
    margin: 0;
    padding: 0;
    border: none;
}
#menu li {
    width: 180px;
    border-bottom: 1px solid #006600;
    margin: 0;
    padding: 0;
    font-size: 80%;
    vertical-align: bottom;
}
#menu a:link, #menu a:visited {
    display: block;
    padding: 5px 5px 5px 0.5em;
    border-left: 12px solid #006600;
    border-right: 1px solid #006600;
    background-color: #CAD6EC;
    color: #006600;
    text-decoration: none;
}
#menu a:hover {
    background-color: #006600;
    color: #FFFFFF;
}
a:link,a:visited {
    color:#006600;
}
a:hover {
    color: #FF6600;
    text-decoration:none;
}

```

```

h2 {
    font: normal 120% Georgia;
    color: #006600;
    background-color: transparent;
    border-bottom: 1px dotted #006600;
}
table {
    font-family: Tahoma;
    font-size: 8pt;
    border-width: 1px;
    border-style: solid;
    border-color: #999999;
    border-collapse: collapse;
    margin: 10px 0px;
}
th {
    color: #FFFFFF;
    font-size: 7pt;
    text-transform: uppercase;
    text-align: center;
    padding: 0.5em;
    border-width: 1px;
    border-style: solid;
    border-color: #969BA5;
    border-collapse: collapse;
    background-color: #006600;
}
td {
    padding: 0.5em;
    color: #333333;
    vertical-align: top;
    border-width: 1px;
    border-style: solid;
    border-color: #969BA5;
    border-collapse: collapse;
}
input,textarea,select{
    font-family: Tahoma;
    font-size: 8pt;
}
#footer {
    padding: 20px 0 10px 255px;
    font-size: 70%;
    color: #FFFFFF;
    background-color: #006600;
}

```

```

<?php
class Paging
{
// Fungsi untuk mengecek halaman dan posisi data
function cariPosisi($batas)
{
if(empty($_GET[halaman])){
    $posisi=0;
    $_GET[halaman]=1;
}
else{
    $posisi = ($_GET[halaman]-1) * $batas;
}
return $posisi;
}

// Fungsi untuk menghitung total halaman
function jumlahHalaman($jmldata, $batas)
{
    $jmlhalaman = ceil($jmldata/$batas);
    return $jmlhalaman;
}

// Fungsi untuk link halaman 1,2,3 ... Next, Prev, First, Last
function navHalaman($halaman_aktif, $jmlhalaman)
{
    $link_halaman = "";

// Link First dan Previous
if ($halaman_aktif > 1)
{
    $link_halaman .= " <a href=$_SERVER[PHP_SELF]?module=$_GET[module]
&halaman=1><< First</a> | ";
}

if (($halaman_aktif-1) > 0)
{
    $previous = $halaman_aktif-1;
    $link_halaman .= "<a href=$_SERVER[PHP_SELF]?module=$_GET[module]
&halaman=$previous>< Previous</a> | ";
}

// Link halaman 1,2,3, ...
for ($i=1; $i<=$jmlhalaman; $i++)
{
    if ($i == $halaman_aktif)
    {
        $link_halaman .= "<b>$i</b> | ";
    }
    else
    {
        $link_halaman .= "<a href=$_SERVER[PHP_SELF]?module=$_GET[module]
&halaman=$i>$i</a> | ";
    }
    $link_halaman .= " ";
}

// Link Next dan Last
if ($halaman_aktif < $jmlhalaman)
{
    $next=$halaman_aktif+1;
    $link_halaman .= " <a href=$_SERVER[PHP_SELF]?module=$_GET[module]
&halaman=$next>Next >></a> ";
}

if (($halaman_aktif != $jmlhalaman) && ($jmlhalaman != 0))

```

```
{
$link_halaman .= " | <a href=$_SERVER[PHP_SELF]?module=$_GET[module]
&halaman=$jmlhalaman>Last >></a> ";
}
return $link_halaman;
}
}
?>
```



```

<?php
function combotgl($awal, $akhir, $var, $default){
echo "<select name=$var>";
echo "<option value=0 selected>$default</option>";
for ($i=$awal; $i<=$akhir; $i++){
    echo "<option value=$i>$i</option>";
}
echo "</select> ";
}

function combobln($awal, $akhir, $var, $default){
include "../config/library.php";
echo "<select name=$var>";
echo "<option value=0 selected>$default</option>";
for ($bln=$awal; $bln<=$akhir; $bln++){
    echo "<option value=$bln>$nama_bln[$bln]</option>";
}
echo "</select> ";
}

function combotgl2($awal, $akhir, $var, $terpilih){
echo "<select name=$var>";
for ($i=$awal; $i<=$akhir; $i++){
    if ($i==$terpilih)
        echo "<option value=$i selected>$i</option>";
    else
        echo "<option value=$i>$i</option>";
}
echo "</select> ";
}

function combobln2($awal, $akhir, $var, $terpilih){
include "../config/library.php";
echo "<select name=$var>";
for ($bln=$awal; $bln<=$akhir; $bln++){
    if ($bln==$terpilih)
        echo "<option value=$bln selected>$nama_bln[$bln]</option>";
    else
        echo "<option value=$bln>$nama_bln[$bln]</option>";
}
echo "</select> ";
}

?>

```

```
<?php
function tgl_indo($tgl){
    $tanggal = substr($tgl,8,2);
    $bulan = getBulan(substr($tgl,5,2));
    $tahun = substr($tgl,0,4);
    return $tanggal.' '.$bulan.' '.$tahun;
}

function getBulan($bln){
    switch ($bln){
        case 1:
            return "Januari";
            break;
        case 2:
            return "Februari";
            break;
        case 3:
            return "Maret";
            break;
        case 4:
            return "April";
            break;
        case 5:
            return "Mei";
            break;
        case 6:
            return "Juni";
            break;
        case 7:
            return "Juli";
            break;
        case 8:
            return "Agustus";
            break;
        case 9:
            return "September";
            break;
        case 10:
            return "Oktober";
            break;
        case 11:
            return "November";
            break;
        case 12:
            return "Desember";
            break;
    }
}
?>
```

```
<?php
  header('location:media.php?module=home');
?>
```




```

<table width="100%" cellpadding=5>
<?php
include "config/koneksi.php";
include "config/fungsi_indotgl.php";
include "config/library.php";
include "config/class_paging.php";

// Bagian Home
if ($_GET[module]=='home'){
    echo "<tr><td align=center><img src=images/welcome.jpg><br><br>
</td></tr>";

    // Tampilkan 3 berita terbaru
    echo "<tr><td class=judul_head>&#187; Berita Terkini</td></tr>";
    $sterkini= mysql_query("SELECT * FROM berita,user
                            WHERE user.id_user=berita.id_user
                            ORDER BY id_berita DESC LIMIT 5");

    while($t=mysql_fetch_array($sterkini)){
        if($t[suara]!=''){
            $filesuara=$t[suara];
            $iconsuara="iconau.jpg";
        }
        if($t[video]!=''){
            {
                $filevideo=$t[video];
                $iconvideo="iconvid.jpg";
            }
            $tgl = tgl_indo($t[tanggal]);
            echo "<tr><td class=isi_kecil>$t[hari], $tgl</td></tr>";
            echo "<tr><td class=judul><a href=?
module=detailberita&id=$t[id_berita]>$t[judul]</a></td></tr>";
            echo "<tr><td class=isi_kecil>Ditulis Oleh : $t
[nama_lengkap]<a href='uploads/$filesuara'><img src='uploads/
$iconsuara' width=20 height=20 hspace=10 border=0></a>
<a href='uploads/$filevideo'><img
src='uploads/$iconvideo' width=20 height=20 hspace=10 border=0></a>
</td></tr>";
            echo "<tr><td class=isi>";
                if ($t[gambar]!=''){
                    echo "<img src='uploads/$t[gambar]' width=150
height=120 hspace=10 border=0 align=left>";
                }

                $kalimat=strtok(nl2br($t[isi_berita])," ");
                for ($i=1;$i<=50;$i++){
                    echo ($kalimat);
                    echo (" "); // Spasi antar kalimat
                    $kalimat=strtok(" "); // Potong per kalimat
                    //hapus isi penanda
                    $filesuara="";
                    $iconsuara="";
                    $filevideo="";
                    $iconvideo="";
                }
                echo " ... <a href=?module=detailberita&id=$t[id_berita]>
Selengkapnya</a>
                <br><br><hr color=white></td></tr>";
            }

    // Tampilkan 5 berita sebelumnya
    echo "<tr><td><img src=images/berita_sebelumnya.jpg></td></tr>";
    $sebelum=mysql_query("SELECT * FROM berita
                            ORDER BY id_berita DESC LIMIT 3,5");

    while($s=mysql_fetch_array($sebelum)){

```



```

        echo "<tr><td class=judul>${d[judul]}</td></tr>";
        echo "<tr><td class=isi_kecil>Ditulis Oleh : ${d[nama_lengkap]}<a
href='uploads/${filesuara}'><img src='uploads/${iconsuara}' width=20
height=20 hspace=10 border=0></a>
        <a href='uploads/${filevideo}'><img src='uploads/
${iconvideo}' width=20 height=20 hspace=10 border=0></a></td></tr>";
        echo "<tr><td class=isi>";
            if (${d[gambar]!='') {
                echo "<img src='uploads/${d[gambar]}' hspace=10 border=0
align=left>";
            }
            $isi_berita=nl2br(${d[isi_berita]});
            echo "${isi_berita}</td></tr>";
            echo "<tr><td class=kembali><br>
[ <a href=javascript:history.go(-1)>Kembali</a> ]</td></tr>";

// Apabila berita dibuka, maka tambahkan counternya
mysql_query("UPDATE berita SET counter=${d[counter]}+1
WHERE id_berita='${_GET[id]}'");
}

// Bagian Berita
elseif ($_GET[module]=='berita'){
    echo "<tr><td class=judul_head>&#187; Berita</td></tr>";

    $p = new Paging;
    $batas = 8;
    $posisi = $p->cariPosisi($batas);

    $sql = "SELECT * FROM berita,user
WHERE user.id_user=berita.id_user
ORDER BY id_berita DESC LIMIT $posisi,$batas";
    $hasil = mysql_query($sql);

    while($r=mysql_fetch_array($hasil)){
        $tgl = tgl_indo($r[tanggal]);
        echo "<tr><td class=isi_kecil>${r[hari]}, $tgl</td></tr>";
        echo "<tr><td class=judul><a href=$_SERVER[PHP_SELF]?
module=detailberita&id=${r[id_berita]}>${r[judul]}</a></td></tr>";
        echo "<tr><td class=isi>";
        $kalimat=strtok(nl2br(${r[isi_berita]})," ");
        for ($i=1;$i<=50;$i++){
            echo ($kalimat);
            echo (" "); // Spasi antar kalimat
            $kalimat=strtok(" "); // Potong per kalimat
        }
        echo " ... <a href=$_SERVER[PHP_SELF]?module=detailberita&id=${r
[id_berita]}>Selengkapnya</a><br><br>
<hr color=white></td></tr>";
    }

    $jmlldata = mysql_num_rows(mysql_query("SELECT * FROM berita"));
    $jmlhalaman = $p->jumlahHalaman($jmlldata, $batas);
    $linkHalaman = $p->navHalaman($_GET[halaman], $jmlhalaman);

    echo "<tr><td class=kembali>${linkHalaman}</td></tr>";
}

// Bagian Agenda
elseif ($_GET[module]=='agenda'){
    echo "<tr><td class=judul_head>&#187; Agenda</td></tr>";
}

```

```

$p      = new Paging;
$batas = 10;
$posisi = $p->cariPosisi($batas);

$sql    = "SELECT * FROM agenda,user
          WHERE user.id_user=agenda.id_user
          ORDER BY id_agenda DESC LIMIT $posisi,$batas";
$hasil = mysql_query($sql);

while($r=mysql_fetch_array($hasil)){
    $tgl_mulai   = tgl_indo($r[tgl_mulai]);
    $tgl_selesai = tgl_indo($r[tgl_selesai]);
    $isi_agenda=nl2br($r[isi_agenda]);

    echo "<tr><td class=isi_kecil>$tgl_mulai s/d $tgl_selesai</td>
</tr>";
    echo "<tr><td class=judul>$r[tema]</td></tr>";
    echo "<tr><td class=isi><b>Topik</b>      : $isi_agenda</td>
</tr>";
    echo "<tr><td class=isi><b>Tempat</b>     : $r[tempat]</td>
</tr>";
    echo "<tr><td class=isi><b>Pengirim</b>  : $r[nama_lengkap]<hr
color=white></td></tr>";
}

$jmlldata      = mysql_num_rows(mysql_query("SELECT * FROM agenda"));
$jmlhalaman    = $p->jumlahHalaman($jmlldata, $batas);
$linkHalaman   = $p->navHalaman($_GET[halaman], $jmlhalaman);

echo "<tr><td class=kembali>$linkHalaman</td></tr>";
}

// Detail Agenda
elseif ($_GET[module]=='detailagenda'){
    $detail=mysql_query("SELECT * FROM agenda,user
                       WHERE user.id_user=agenda.id_user
                       AND id_agenda='".$_GET[id]'");
    $d      = mysql_fetch_array($detail);
    $tgl_mulai   = tgl_indo($d[tgl_mulai]);
    $tgl_selesai = tgl_indo($d[tgl_selesai]);
    $isi_agenda=nl2br($d[isi_agenda]);

    echo "<tr><td class=isi_kecil>$tgl_mulai s/d $tgl_selesai</td>
</tr>";
    echo "<tr><td class=judul>$d[tema]</td></tr>";
    echo "<tr><td class=isi><b>Topik</b>      : $isi_agenda</td>
</tr>";
    echo "<tr><td class=isi><b>Tempat</b>     : $d[tempat]</td>
</tr>";
    echo "<tr><td class=isi><b>Pengirim</b>  : $d[nama_lengkap]<hr
color=white></td></tr>";

    echo "<tr><td class=kembali><br>
[ <a href=javascript:history.go(-1)>Kembali</a> ]</td></tr>";
}

// Bagian Pengumuman
elseif ($_GET[module]=='pengumuman'){
    echo "<tr><td class=judul_head>&#187; Pengumuman</td></tr>";

    $p      = new Paging;
    $batas = 10;
    $posisi = $p->cariPosisi($batas);

```

```

$sql = "SELECT * FROM pengumuman,user
WHERE user.id_user=pengumuman.id_user
ORDER BY id_pengumuman DESC LIMIT $posisi,$batas";

$hasil = mysql_query($sql);

while($r=mysql_fetch_array($hasil)){
    $tgl = tgl_indo($r[tanggal]);
    $isi = nl2br($r[isi]);

    echo "<tr><td class=isi_kecil>$tgl</td></tr>";
    echo "<tr><td class=judul>$r[judul]</td></tr>";
    echo "<tr><td class=isi>$isi</td></tr>";
    echo "<tr><td class=isi><b>Pengirim</b> : $r[nama_lengkap]<hr
color=white></td></tr>";
}

$jmlldata = mysql_num_rows(mysql_query("SELECT * FROM
pengumuman"));
$jmlhalaman = $p->jumlahHalaman($jmlldata, $batas);
$linkHalaman = $p->navHalaman($_GET[halaman], $jmlhalaman);

echo "<tr><td class=kembali>$linkHalaman</td></tr>";
}

// Detail Pengumuman
elseif ($_GET[module]=='detailpengumuman'){
    $detail=mysql_query("SELECT * FROM pengumuman,user
WHERE user.id_user=pengumuman.id_user
AND id_pengumuman='$_GET[id]'");
    $d = mysql_fetch_array($detail);
    $tgl = tgl_indo($d[tanggal]);
    $isi = nl2br($d[isi]);

    echo "<tr><td class=isi_kecil>$tgl</td></tr>";
    echo "<tr><td class=judul>$d[judul]</td></tr>";
    echo "<tr><td class=isi>$isi</td></tr>";
    echo "<tr><td class=isi><b>Pengirim</b> : $d[nama_lengkap]<hr
color=white></td></tr>";

    echo "<tr><td class=kembali><br>
[ <a href=javascript:history.go(-1)>Kembali</a> ]</td></tr>";
}

// Bagian Hubungi Kami
elseif ($_GET[module]=='hubungi'){
    echo "<tr><td class=judul_head>&#187; Hubungi Kami</td></tr>";

    echo "<tr><td class=isi>Silahkan hubungi kami secara online:</td>
</tr>";

    echo "<form method=POST action='?module=kirimemail'>
<tr><td class=isi>Nama : <input type=text name=nama size=
35></td></tr>
<tr><td class=isi>E-mail : <input type=text name=email size=
35></td></tr>
<tr><td class=isi>Subjek: <input type=text name=subjek size=
50></td></tr>
<tr><td class=isi>Pesan : <br><textarea name=pesan rows=13
cols=70></textare></td></tr>
<tr><td><input type=submit value=Kirim></td></tr>
</form>";
}

```

```

echo "<tr><td class=kembali><br>
    [ <a href=javascript:history.go(-1)>Kembali</a> ]</td></tr>";
}

// Bagian Kirim Email
elseif ($_GET[module]=='kirimemail'){
    mysql_query("INSERT INTO hubungi(nama,
        email,
        subjek,
        pesan,
        tanggal)
        VALUES('$_POST[nama]',
        '$_POST[email]',
        '$_POST[subjek]',
        '$_POST[pesan]',
        '$tgl_sekarang')");

    echo "<tr><td class=judul_head>&#187; Status Email</td></tr>
    <tr><td class=isi>Email telah sukses terkirim dan segera akan
    kami balas</td></tr>
    <tr><td class=kembali><br>
    [ <a href=index.php>Kembali</a> ]</td></tr>";
}

// Bagian Profil
elseif ($_GET[module]=='profil'){
    echo "<tr><td class=judul_head>&#187; Profil Lembaga</td></tr>";

    $profil = mysql_query("SELECT * FROM modul WHERE
    id_modul='11'");
    $r      = mysql_fetch_array($profil);

    echo "<tr><td class=isi>";
    if ($r[gambar]!=''){
        echo "<img src='uploads/$r[gambar]' hspace=10 border=0
    align=left>";
    }
    $isi_profil=nl2br($r[static_content]);
    echo "$isi_profil</td></tr>";

    echo "<tr><td class=kembali><br>
    [ <a href=javascript:history.go(-1)>Kembali</a> ]</td></tr>";
}

// Bagian Hasil Pencarian
elseif ($_GET[module]=='hasilcari'){
    echo "<tr><td class=judul_head>&#187; Hasil Pencarian</td></tr>";

    // Hanya mencari berita, apabila diperlukan bisa ditambahkan utk
    mencari agenda, pengumuman, dll
    $cari = mysql_query("SELECT * FROM berita WHERE isi_berita
    LIKE '%$_POST[kata]%'");
    $jumlah = mysql_num_rows($cari);

    if ($jumlah > 0){
        echo "<tr><td class=isi>
        <br>Ditemukan <b>$jumlah</b> berita dengan kata <b>$_POST
    [kata]</b> : <ul>";

        while($r=mysql_fetch_array($cari)){

```

```

        echo "<li><a href=?module=detailberita&id=$r[id_berita]>$r
[judul]</a></li>";
    }
    echo "</ul></td></tr>";
}
else{
    echo "<tr><td class=judul>
        Tidak ditemukan berita dengan kata <b>$_POST[kata]</b></td>
</tr>";
}

    echo "<tr><td class=kembali><br>
    [ <a href=javascript:history.go(-1)>Kembali</a> ]</td></tr>";
}
//Bagian berita politik
elseif($_GET[module]=='politik')
{
    echo "<tr><td class=judul_head>&#187; Berita Politik</td>
</tr>";

    $p      = new Paging;
    $batas  = 8;
    $posisi = $p->cariPosisi($batas);

    $sql    = "SELECT * FROM berita,user
        WHERE user.id_user=berita.id_user && berita.id_kategori=1
        ORDER BY id_berita DESC LIMIT $posisi,$batas";
    $hasil  = mysql_query($sql);

    while($r=mysql_fetch_array($hasil)){
        $tgl = tgl_indo($r[tanggal]);
        echo "<tr><td class=isi_kecil>$r[hari], $tgl</td></tr>";
        echo "<tr><td class=judul><a href=$_SERVER[PHP_SELF]?
module=detailberita&id=$r[id_berita]>$r[judul]</a></td></tr>";
        echo "<tr><td class=isi>";
        $kalimat=strtok(nl2br($r[isi_berita])," ");
        for ($i=1;$i<=50;$i++){
            echo ($kalimat);
            echo (" "); // Spasi antar kalimat
            $kalimat=strtok(" "); // Potong per kalimat
        }
        echo " ... <a href=$_SERVER[PHP_SELF]?module=detailberita&id=$r
[id_berita]>Selengkapnya</a><br><br>
        <hr color=white></td></tr>";
    }

    $jmlldata      = mysql_num_rows(mysql_query("SELECT * FROM berita"));
    $jmlhalaman    = $p->jumlahHalaman($jmlldata, $batas);
    $linkHalaman   = $p->navHalaman($_GET[halaman], $jmlhalaman);

    echo "<tr><td class=kembali>$linkHalaman</td></tr>";
}
//Bagian berita ekonomi
elseif($_GET[module]=='ekonomi')
{
    echo "<tr><td class=judul_head>&#187; Berita Politik</td>
</tr>";

    $p      = new Paging;
    $batas  = 8;
    $posisi = $p->cariPosisi($batas);

    $sql    = "SELECT * FROM berita,user
        WHERE user.id_user=berita.id_user && berita.id_kategori=2

```

```

ORDER BY id_berita DESC LIMIT $posisi,$batas";
$hasil = mysql_query($sql);

while($r=mysql_fetch_array($hasil)){
    $tgl = tgl_indo($r[tanggal]);
    echo "<tr><td class=isi_kecil>$r[hari], $tgl</td></tr>";
    echo "<tr><td class=judul><a href=$_SERVER[PHP_SELF]?
module=detailberita&id=$r[id_berita]>$r[judul]</a></td></tr>";
    echo "<tr><td class=isi>";
    $kalimat=strtok(nl2br($r[isi_berita])," ");
    for ($i=1;$i<=50;$i++){
        echo ($kalimat);
        echo (" "); // Spasi antar kalimat
        $kalimat=strtok(" "); // Potong per kalimat
    }
    echo " ... <a href=$_SERVER[PHP_SELF]?module=detailberita&id=$r
[id_berita]>Selengkapnya</a><br><br>
        <hr color=white></td></tr>";
    }

    $jmlldata = mysql_num_rows(mysql_query("SELECT * FROM berita"));
    $jmlhalaman = $p->jumlahHalaman($jmlldata, $batas);
    $linkHalaman = $p->navHalaman($_GET[halaman], $jmlhalaman);

    echo "<tr><td class=kembali>$linkHalaman</td></tr>";
}
//Bagian berita teknologi
elseif($_GET[module]=='teknologi')
{
    echo "<tr><td class=judul_head>&#187; Berita Politik</td>
</tr>";

    $p = new Paging;
    $batas = 8;
    $posisi = $p->cariPosisi($batas);

    $sql = "SELECT * FROM berita,user
WHERE user.id_user=berita.id_user && berita.id_kategori=3
ORDER BY id_berita DESC LIMIT $posisi,$batas";
    $hasil = mysql_query($sql);

    while($r=mysql_fetch_array($hasil)){
        $tgl = tgl_indo($r[tanggal]);
        echo "<tr><td class=isi_kecil>$r[hari], $tgl</td></tr>";
        echo "<tr><td class=judul><a href=$_SERVER[PHP_SELF]?
module=detailberita&id=$r[id_berita]>$r[judul]</a></td></tr>";
        echo "<tr><td class=isi>";
        $kalimat=strtok(nl2br($r[isi_berita])," ");
        for ($i=1;$i<=50;$i++){
            echo ($kalimat);
            echo (" "); // Spasi antar kalimat
            $kalimat=strtok(" "); // Potong per kalimat
        }
        echo " ... <a href=$_SERVER[PHP_SELF]?module=detailberita&id=$r
[id_berita]>Selengkapnya</a><br><br>
            <hr color=white></td></tr>";
    }

    $jmlldata = mysql_num_rows(mysql_query("SELECT * FROM berita"));
    $jmlhalaman = $p->jumlahHalaman($jmlldata, $batas);
    $linkHalaman = $p->navHalaman($_GET[halaman], $jmlhalaman);

    echo "<tr><td class=kembali>$linkHalaman</td></tr>";
}

```



```

//Bagian berita olahraga
elseif($_GET[module]=='olahraga')
{
    echo "<tr><td class=judul_head>&#187; Berita Politik</td>
</tr>";

    $p      = new Paging;
    $batas  = 8;
    $posisi = $p->cariPosisi($batas);

    $sql    = "SELECT * FROM berita,user
              WHERE user.id_user=berita.id_user && berita.id_kategori=4
              ORDER BY id_berita DESC LIMIT $posisi,$batas";
    $hasil  = mysql_query($sql);

    while($r=mysql_fetch_array($hasil)){
        $tgl = tgl_indo($r[tanggal]);
        echo "<tr><td class=isi_kecil>$r[hari], $tgl</td></tr>";
        echo "<tr><td class=judul><a href=$_SERVER[PHP_SELF]?
module=detailberita&id=$r[id_berita]>$r[judul]</a></td></tr>";
        echo "<tr><td class=isi>";
        $kalimat=strtok(nl2br($r[isi_berita]), " ");
        for ($i=1;$i<=50;$i++){
            echo ($kalimat);
            echo (" "); // Spasi antar kalimat
            $kalimat=strtok(" "); // Potong per kalimat
        }
        echo " ... <a href=$_SERVER[PHP_SELF]?module=detailberita&id=$r
[id_berita]>Selengkapnya</a><br><br>
<hr color=white></td></tr>";

    }

    $jmlldata      = mysql_num_rows(mysql_query("SELECT * FROM berita"));
    $jmlhalaman    = $p->jumlahHalaman($jmlldata, $batas);
    $linkHalaman   = $p->navHalaman($_GET[halaman], $jmlhalaman);

    echo "<tr><td class=kembali>$linkHalaman</td></tr>";
}
//Bagian berita kriminal
elseif($_GET[module]=='kriminal')
{
    echo "<tr><td class=judul_head>&#187; Berita Politik</td>
</tr>";

    $p      = new Paging;
    $batas  = 8;
    $posisi = $p->cariPosisi($batas);

    $sql    = "SELECT * FROM berita,user
              WHERE user.id_user=berita.id_user && berita.id_kategori=5
              ORDER BY id_berita DESC LIMIT $posisi,$batas";
    $hasil  = mysql_query($sql);

    while($r=mysql_fetch_array($hasil)){
        $tgl = tgl_indo($r[tanggal]);
        echo "<tr><td class=isi_kecil>$r[hari], $tgl</td></tr>";
        echo "<tr><td class=judul><a href=$_SERVER[PHP_SELF]?
module=detailberita&id=$r[id_berita]>$r[judul]</a></td></tr>";
        echo "<tr><td class=isi>";
        $kalimat=strtok(nl2br($r[isi_berita]), " ");
        for ($i=1;$i<=50;$i++){
            echo ($kalimat);
            echo (" "); // Spasi antar kalimat
            $kalimat=strtok(" "); // Potong per kalimat
        }
    }
}

```

```

    echo " ... <a href=$_SERVER[PHP_SELF]?module=detailberita&id=$r
[id_berita]>Selengkapnya</a><br><br>
        <hr color=white></td></tr>";

    }

    $jmlldata      = mysql_num_rows(mysql_query("SELECT * FROM berita"));
    $jmlhalaman    = $p->jumlahHalaman($jmlldata, $batas);
    $linkHalaman   = $p->navHalaman($_GET[halaman], $jmlhalaman);

    echo "<tr><td class=kembali>$linkHalaman</td></tr>";
}
//Bagian berita kesehatan
elseif($_GET[module]=='kesehatan')
{
    echo "<tr><td class=judul_head>&#187; Berita Politik</td>
</tr>";

    $p      = new Paging;
    $batas  = 8;
    $posisi = $p->cariPosisi($batas);

    $sql    = "SELECT * FROM berita,user
              WHERE user.id_user=berita.id_user && berita.id_kategori=6
              ORDER BY id_berita DESC LIMIT $posisi,$batas";
    $hasil  = mysql_query($sql);

    while($r=mysql_fetch_array($hasil)){
        $tgl = tgl_indo($r[tanggal]);
        echo "<tr><td class=isi_kecil>$r[hari], $tgl</td></tr>";
        echo "<tr><td class=judul><a href=$_SERVER[PHP_SELF]?
module=detailberita&id=$r[id_berita]>$r[judul]</a></td></tr>";
        echo "<tr><td class=isi>";
        $kalimat=strtok(nl2br($r[isi_berita]), " ");
        for ($i=1;$i<=50;$i++){
            echo ($kalimat);
            echo (" "); // Spasi antar kalimat
            $kalimat=strtok(" "); // Potong per kalimat
        }
        echo " ... <a href=$_SERVER[PHP_SELF]?module=detailberita&id=$r
[id_berita]>Selengkapnya</a><br><br>
            <hr color=white></td></tr>";

    }

    $jmlldata      = mysql_num_rows(mysql_query("SELECT * FROM berita"));
    $jmlhalaman    = $p->jumlahHalaman($jmlldata, $batas);
    $linkHalaman   = $p->navHalaman($_GET[halaman], $jmlhalaman);

    echo "<tr><td class=kembali>$linkHalaman</td></tr>";
}
//Bagian berita hiburan
elseif($_GET[module]=='hiburan')
{
    echo "<tr><td class=judul_head>&#187; Berita Politik</td>
</tr>";

    $p      = new Paging;
    $batas  = 8;
    $posisi = $p->cariPosisi($batas);

    $sql    = "SELECT * FROM berita,user
              WHERE user.id_user=berita.id_user && berita.id_kategori=7
              ORDER BY id_berita DESC LIMIT $posisi,$batas";
    $hasil  = mysql_query($sql);

```

```

while($r=mysql_fetch_array($hasil)){
    $tgl = tgl_indo($r[tanggal]);
    echo "<tr><td class=isi_kecil>$r[hari], $tgl</td></tr>";
    echo "<tr><td class=judul><a href=$_SERVER[PHP_SELF]?
module=detailberita&id=$r[id_berita]>$r[judul]</a></td></tr>";
    echo "<tr><td class=isi>";
    $kalimat=strtok(nl2br($r[isi_berita]), " ");
    for ($i=1;$i<=50;$i++){
        echo ($kalimat);
        echo (" "); // Spasi antar kalimat
        $kalimat=strtok(" "); // Potong per kalimat
    }
    echo " ... <a href=$_SERVER[PHP_SELF]?module=detailberita&id=$r
[id_berita]>Selengkapnya</a><br><br>
        <hr color=white></td></tr>";
    }

    $jmlldata      = mysql_num_rows(mysql_query("SELECT * FROM berita"));
    $jmlhalaman    = $p->jumlahHalaman($jmlldata, $batas);
    $linkHalaman   = $p->navHalaman($_GET[halaman], $jmlhalaman);

    echo "<tr><td class=kembali>$linkHalaman</td></tr>";
}
?>
</table>

```

```
<?php
$server = "localhost";
$username = "root";
$password = "";
$database = "dbmedia";

// Koneksi dan memilih database di server
mysql_connect($server,$username,$password) or die("Koneksi gagal");
mysql_select_db($database) or die("Database tidak bisa dibuka");
?>
```



```
<?php
$seminggu = array
("Minggu", "Senin", "Selasa", "Rabu", "Kamis", "Jumat", "Sabtu");
$hari = date("w");
$hari_ini = $seminggu[$hari];

$tgl_sekarang = date("Ymd");
$thn_sekarang = date("Y");
$jam_sekarang = date("H:i:s");

$nama_bln=array(1=> "Januari", "Februari", "Maret", "April", "Mei",
"Juni", "Juli", "Agustus", "September",
"Oktober", "November", "Desember");
?>
```

```
<?php
/**
 * This PHP implementation of the LZW
 compression algorithm allows for
 * the compression of bytes using the
 Adobe PDF characteristics according
 * to the "PDF Reference Second
 Edition" by Adobe Systems Incorporated
 (2000)
 *
 * This code is primarily a conversion
 from Mark Nelson's LZW.c which can be
 found here:
 *
 http://marknelson.us/attachments/lzw-
 data-compression/lzw.c
 * Do not contact Mr. Nelson regarding
 this package, because this code was
 written by Samuel Shull
 *
 */
/**
 * @package LZW
 * @example /lzwexample.php Example
 usage of this class.
 * @category Numbers
 * @author Sam Shull
 <samshull@samshull.com>
 * @copyright Copyright (c) 2007, Sam
 Shull
 * @license
 http://www.samshull.com/bsdlicense.txt
 BSD License
 * @link
 http://samshull.com/lzwexample.php
 * @version 0.9
 * @access public
 */
class LZW{
/**
 * Table for storing codes
 *

```

```

* @var array
* @access protected
*/
    var $code_value = array();
/**
* Table for storing prefixes to codes
*
* @var array
* @access protected
*/
    var $prefix_code = array();

/**
* Table for storing individual
characters
*
* @var array
* @access protected
*/
    var $append_character = array();
    var $hasil=array();
/**
* Output
*
* @var string
* @access protected
*/
    var $out = "";
/**
* Total size of table of values
*
* @var integer
* @access protected
*/
    var $TABLE_SIZE = 5021;
/**
* Number of bits available for encoding
*
* @var integer
* @access protected
*/

```

```

        var $output_bit_count = 0;
        var $looper=0;
        var $pointer=0;
/**
 * The actual bits for encoding
 *
 * @var string
 * @access protected
 */
        var $output_bit_buffer = "0";
/**
 * Next code in the table
 *
 * @var integer
 * @access protected
 */
        var $next_code = 258;
/**
 * Decoding: the table
 *
 * @var array
 * @access protected
 */
        var $sTable = array();
/**
 * Data to be decoded
 *
 * @var string
 * @access protected
 */
        var $data = NULL;
        var $dataxxx ;
/**
 * Decoding: next code (same as
 * $next_code)
 *
 * @var integer
 * @access protected
 */
        var $tIdx;
/**
 * bits in next code

```



```

*
* @var integer
* @access protected
*/
    var $bitsToGet = 9;
/**
* Position holder within data string
*
* @var string
* @access protected
*/
    var $bytePointer;
/**
* Position holder for bits in data
string
*
* @var string
* @access protected
*/
    var $bitPointer;
/**
* Next value to be decoded
*
* @var integer
* @access protected
*/
    var $nextData = 0;
/**
* Next number of bits to be decoded
*
* @var string
* @access protected
*/
    var $nextBits = 0;
/**
* Table of max bit values per number of
bits
*
* @var string
* @access protected
*/
    var $andTable = array(511, 1023,

```

```

2047, 4095);
/**
 * Method: compress
 * The primary method used by this
class, accepts only a string as input
and
 * returns the string compressed.
 */
/**
 * Method: find_match - if PHP5 mark as
private or protected
 * Finds the matching index of the
character with the table
 * @param string $hash_prefix
 * @param char $hash_character
 * @return int
 */
/**
// The following methods are adapted
directly from FPDF - Version 1.1 and are
only
    included here in order to effect the
decompression of the above encoding
algorithm
//
// Copyright 2004,2005 Setasign - Jan
Slabon
//
// Licensed under the Apache License,
Version 2.0 (the "License");
// you may not use this file except in
compliance with the License.
// You may obtain a copy of the License
at
//
//
http://www.apache.org/licenses/LICENSE-
2.0
//
// Unless required by applicable law or
agreed to in writing, software

```

```

// distributed under the License is
distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF
ANY KIND, either express or implied.
// See the License for the specific
language governing permissions and
// limitations under the License.
/**
    function bbb($aray)
    {
        if($aray[0]>255)
        {
            $poin=$code_value
[$aray[0]];
            $p[0]=$prefix_code
[$poin];
            $p[1]=
$append_character[$poin];
            $this->bbb($p);
        }
        else
        {
            $decoder.=chr($aray
[0]);
        }
        if($aray[1]>255)
        {
            $angka=$code_value
[$aray[1]];
            $p[0]=$prefix_code
[$angka];
            $p[1]=
$append_character[$angka];
            $this->bbb($p);
        }
        else
        {
            $decoder.=chr($aray
[1]);
        }
    }

```

```

    }
    // $dataxxx=128;
    function decode($dataxxx)
    {
        $pointer=0;
        $looper=0;
        $point=0;
        for ($p=0; $p<65536; $p++)
        {
            $code_value[$p]=NULL;
            $prefix_code[$p]=NULL;
            $append_character[$p]
=
=NULL;
        }
        while($looper<(strlen
($dataxxx)-3))
        //while($dataxxx[$point]!==-
1)
        {
            $byte=ord($dataxxx
[$looper]);
            if($byte==3)
            {
                $byte=0;
            }
            $bit_buffer|=( $byte
<<24);
            $looper++;
            $byte=ord($dataxxx
[$looper]);
            if($byte==3)
            {
                $byte=0;
            }
            $bit_buffer|=( $byte
<<16);
            $out=( $bit_buffer>>
20);
            // $hasil.=chr($out);

```

```

$hasil[$pointer]=$out;
$result.="-".$out;
$bit_buffer=
$bit_buffer<<12;
$looper++;
$byte=ord($dataxxx
[$looper]);

if($byte==3)
{
    $byte=0;
}

$bit_buffer|=( $byte
<<20);
$out=( $bit_buffer>>
20);

// $hasil.=chr($out);
$pointer++;
$hasil[$pointer]=$out;
$result.="-".$out;
// $bit_buffer=
$bit_buffer<<12;
$bit_buffer=0;
$looper++;
$pointer++;

}
//buat outputnya
// $d=0;
// $n=0;

$w=$hasil[0];
// $nextcode=256;
$index=($w<<4)^($hasil[0]);
$code_value[256]=$index;
$prefix_code[$index]=$w;
$append_character[$index]=
$hasil[0];

// $w=$hasil[0];
// $decoding=chr($w);

```

```

        $nextcode=256;

        //$w=$hasil[0];
        //$n++;
        for ($n=1;$n<$pointer;$n++)
        {
            $index=($w<<4)^($hasil
[$n]);

            if($prefix_code
[$index]==NULL)
            {
                $nextcode++;
                $code_value
[$nextcode]=$index;
                $prefix_code
[$index]=$w;

                $append_character[$index]=$hasil
[$n];
                $index=0;
            }
            //decoding baru
            /*
            if($w>255)
            {
                $tanda=
$code_value[$w];
                $asa[0]=
$prefix_code[$tanda];
                $asa[1]=
$append_character[$tanda];
                $this->bbb
($asa);
            }
            else
            {
                $decoder.=chr
($w);
            }
            */
            //hasil decoding $w

```

```

        if ($w<=255)
        {
            $decoding .=chr
($w);
        }
        else if ($w>255)
        {
            $petunjuk=
$code_value[$w];
            if ($prefix_code
[$petunjuk]>255)
            {
                $petunjuk2
=$code_value[$prefix_code[$petunjuk]];
                if
($prefix_code[$petunjuk2]>255)
                {
                    $petunjuk21=$code_value
[$prefix_code[$petunjuk2]];
                    if
($prefix_code[$petunjuk21]>255)
                    {
                        $petunjuk211=$code_value
[$prefix_code[$petunjuk21]];
                        $decoding.=chr($prefix_code
[$petunjuk211]);
                        $decoding.=chr($append_character
[$petunjuk211]);
                    }
                    else
                    {
                        $decoding.=chr($prefix_code
[$petunjuk21]);
                    }
                    if
($append_character[$petunjuk21]>255)

```

```

{
    $petunjuk212=$code_value
[$append_character[$petunjuk21]];
    $decoding.=chr($prefix_code
[$petunjuk212]);
    $decoding.=chr($append_character
[$petunjuk212]);
}
else
{
    $decoding.=chr($append_character
[$petunjuk21]);
}
}
else
{
    $decoding.=chr($prefix_code
[$petunjuk2]);
}
if
($append_character[$petunjuk2]>255)
{
    $petunjuk22=$code_value
[$append_character[$petunjuk2]];
    if
($prefix_code[$petunjuk22]>255)
{
        $petunjuk221=$code_value
[$prefix_code[$petunjuk22]];
        $decoding.=chr($prefix_code
[$petunjuk221]);
    }
}
}
}
}

```



```

        $decoding.=chr($append_character
[$petunjuk221]);
    }
    else
    {
        $decoding.=chr($prefix_code
[$petunjuk22]);
    }
    if
($append_character[$petunjuk22]>255)
    {
        $petunjuk222=$code_value
[$append_character[$petunjuk22]];
        $decoding.=chr($prefix_code
[$petunjuk222]);
        $decoding.=chr($append_character
[$petunjuk222]);
    }
    else
    {
        $decoding.=chr($append_character
[$petunjuk22]);
    }
    }
    else
    {
        $decoding.=chr($append_character
[$petunjuk2]);
    }
    }
    else
    {
        $decoding.
=chr($prefix_code[$petunjuk]);
    }
}

```

```

        if
($append_character[$petunjuk]>255)
        {
            $petunjuk3
            =$code_value[$append_character
            [$petunjuk]];
            if
            ($prefix_code[$petunjuk3]>255)
            {
                $petunjuk31=$code_value
                [$prefix_code[$petunjuk3]];
                if
                ($prefix_code[$petunjuk31]>255)
                {
                    $petunjuk311=$code_value
                    [$prefix_code[$petunjuk31]];
                    $decoding.=chr($prefix_code
                    [$petunjuk311]);
                    $decoding.=chr($append_character
                    [$petunjuk311]);
                }
                else
                {
                    $decoding.=chr($prefix_code
                    [$petunjuk31]);
                }
            }
            if
            ($append_character[$petunjuk31]>255)
            {
                $petunjuk312=$code_value
                [$append_character[$petunjuk31]];
                $decoding.=chr($prefix_code
                [$petunjuk312]);
                $decoding.=chr($append_character

```

```

[$petunjuk312]);
}
else
{
    $decoding.=chr($append_character
[$petunjuk31]);
}
}
else
{
    $decoding.=chr($prefix_code
[$petunjuk3]);
}
if
($append_character[$petunjuk3]>255)
{
    $petunjuk32=$code_value
[$append_character[$petunjuk3]];
if
($prefix_code[$petunjuk32]>255)
{
    $petunjuk321=$code_value
[$prefix_code[$petunjuk32]];
    $decoding.=chr($prefix_code
[$petunjuk321]);
    $decoding.=chr($append_character
[$petunjuk321]);
}
else
{
    $decoding.=chr($prefix_code
[$petunjuk22]);
}
if
($append_character[$petunjuk32]>255)

```

```

{
    $petunjuk322=$code_value
[$append_character[$petunjuk32]];
    $decoding.=chr($prefix_code
[$petunjuk322]);
    $decoding.=chr($append_character
[$petunjuk322]);
}
else
{
    $decoding.=chr($append_character
[$petunjuk32]);
}
}
else
{
    $decoding.=chr($append_character
[$petunjuk3]);
}
}
else
{
    $decoding.
=chr($append_character[$petunjuk]);
}
}
    $w=$hasil[$n];
}
    //$ddd=$dataxxx[0].$dataxxx
[1].$dataxxx[2].$dataxxx[3].$dataxxx[4].
$dataxxx[5].$dataxxx[6].$dataxxx[7].
$dataxxx[8];
    //return strlen($dataxxx);
    return $decoding;
}

```

}

?>



```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<title>***Mobile Citizen Media***</title>
<link href="config/style.css" rel="stylesheet" type="text/css">

<style type="text/css">
<!--
body {
    background-image: url(images/background.gif);
    background-repeat: repeat-y;
    background-position: center;
    margin-left: 0px;
    margin-top: 0px;
    margin-right: 0px;
    margin-bottom: 0px;
}
-->
</style></head>

<body>
<table width="780" border="0" align="center" cellpadding="0"
cellspacing="0">
<tr>
<td colspan="2"></td>
</tr>
<tr>
<td width="200" valign="top" bgcolor="#006600">

<?php include "kiri.php"; ?>

<p align="center">&nbsp;</p>
</td>
<td width="580" valign="top" bgcolor="#FFFFFF"><p>

<?php include "kanan.php"; ?>

<br>
</p> </td>
</tr>
<tr>
<td height="30" colspan="2"></td>
</tr>
</table>
</body>
</html>

```

```

<?php
//require("../config/koneksi.php");
include "koneksi.php";
include "config/library.php";
include "lzwc.php";
//include "lzwc_asli.php";
$target_path = "uploads/";

$username = trim($_POST['username']);
$judul=$_POST['judul'];
$kategori=$_POST['kategori'];
$isip=$_POST['isi'];
//$isi=bindec($isi);
if(ord($isi[0])<16)
{
    $l = new LZW;
    $isi = $l->decode($isi);
}
else
{
    $isi=$isi;
}

//$isi=chr($isi);
//decode to string
//kamus
/*
for($i = 0; $i < 256; $i++) //populate dictionary with ascii
characters
    $startDictionary[$i] = chr($i);
//decode
function decodeLZW($string, $bits) {
    global $startDictionary;

    $dictionary = $startDictionary;

    $tokens = str_split($string, $bits); //tokenize the string -
split every at every $bits, where $bits is the size of the bits used
to encode one symbol.

    $decodedString = $dictionary[bindec($tokens[0])];
//function bindec - decode binary string to a decimal

    $word = $dictionary[bindec($tokens[0])];

    for($i = 1; $i < count($tokens); $i++) {

        $x = bindec($tokens[$i]);
        $element = $dictionary[$x];

        if(!$element) {

            $element = $word + $word{0};

        }

        $decodedString .= $element;

        $dictionary[] = "$word{$element{0}}";
        $word = $element;

    }

    return $decodedString;
}
}

```

```

//proses decoding
    $input=$isi;
    //$bits = substr($input, 0, 16);    //remove the first 16 bits
    $input = substr($input, 16);

    //$isi = decodeLZW($input, bindec($bits)); //decode ($bits is
converted to a decimal)

//decoding selesai
*/
$gambar=$_FILES['filegambar']['name'];

$suara=$_FILES['filesuara']['name'];
$video=$_FILES['filevideo']['name'];
if (empty($gambar) && empty($suara) && empty($video))
{
    mysql_query("INSERT INTO berita(judul,
                                id_kategori,
                                isi_berita,
                                id_user,
                                jam,
                                tanggal,
                                hari)
                                VALUES ('$judul',
                                '$kategori',
                                '$isi',
                                '$username',
                                '$jam_sekarang',
                                '$tgl_sekarang',
                                '$hari_ini')");
}
else if(!empty($gambar)&& empty($suara) && empty($video))
{
    $direktori = $target_path .$_FILES['filegambar']['name'];

    //echo $target_path;
    //echo "<br>";
    //echo $_FILES['uploadedfile']['name'];
    //echo "<br>";
    //echo $_FILES['uploadedfile']['tmp_name'];
    move_uploaded_file($_FILES['filegambar']['tmp_name'], $direktori);

    //if() {
    //mysql_query("insert into berita(id_user,gambar,judul,isi_berita)
    values('$username','$target_path2','$judul','$isi')");
    mysql_query("INSERT INTO berita(judul,
                                id_kategori,
                                isi_berita,
                                id_user,
                                jam,
                                tanggal,
                                hari,
                                gambar)
                                VALUES ('$judul',
                                '$kategori',
                                '$isi',
                                '$username',
                                '$jam_sekarang',
                                '$tgl_sekarang',
                                '$hari_ini',
                                '$gambar')");

    //header('location:index.php');
    ///} else{
    //    echo "There was an error uploading the file, please try

```



```

again!";
}
else if(!empty($suara) && empty($gambar) && empty($video))
{
    $direktori = $target_path .$_FILES['filesuara']['name'];
    move_uploaded_file($_FILES['filesuara']['tmp_name'],
$direktori);
    mysql_query("INSERT INTO berita(judul,
                                id_kategori,
                                isi_berita,
                                id_user,
                                jam,
                                tanggal,
                                hari,
                                suara)
                VALUES('$judul',
                        '$kategori',
                        '$isi',
                        '$username',
                        '$jam_sekarang',
                        '$tgl_sekarang',
                        '$hari_ini',
                        '$suara')");
}
else if(!empty($video) && empty($suara) && empty($gambar))
{
    $direktori = $target_path .$_FILES['filevideo']['name'];
    move_uploaded_file($_FILES['filevideo']['tmp_name'],
$direktori);
    mysql_query("INSERT INTO berita(judul,
                                id_kategori,
                                isi_berita,
                                id_user,
                                jam,
                                tanggal,
                                hari,
                                video)
                VALUES('$judul',
                        '$kategori',
                        '$isi',
                        '$username',
                        '$jam_sekarang',
                        '$tgl_sekarang',
                        '$hari_ini',
                        '$video')");
}
else if(!empty($suara) && !empty($gambar) && empty($video))
{
    $direktori = $target_path .$_FILES['filesuara']['name'];
    $direktori2 = $target_path .$_FILES['filegambar']['name'];
    move_uploaded_file($_FILES['filesuara']['tmp_name'],
$direktori);
    move_uploaded_file($_FILES['filegambar']['tmp_name'],
$direktori2);
    mysql_query("INSERT INTO berita(judul,
                                id_kategori,
                                isi_berita,
                                id_user,
                                jam,
                                tanggal,
                                hari,
                                gambar,
                                suara)
                VALUES('$judul',
                        '$kategori',

```

```

        '$isi',
        '$username',
        '$jam_sekarang',
        '$tgl_sekarang',
        '$hari_ini',
        '$gambar',
        '$suara')");
    }
else if(empty($suara) && !empty($gambar) && !empty($video))
{
    $direktori = $target_path .$_FILES['filevideo']['name'];
    $direktori2 = $target_path .$_FILES['filegambar']['name'];
    move_uploaded_file($_FILES['filevideo']['tmp_name'],
    $direktori);
    move_uploaded_file($_FILES['filegambar']['tmp_name'],
    $direktori2);
    mysql_query("INSERT INTO berita(judul,
        id_kategori,
        isi_berita,
        id_user,
        jam,
        tanggal,
        hari,
        gambar,
        video)
        VALUES('$judul',
        '$kategori',
        '$isi',
        '$username',
        '$jam_sekarang',
        '$tgl_sekarang',
        '$hari_ini',
        '$gambar',
        '$video')");
    }
else if(!empty($suara) && empty($gambar) && !empty($video))
{
    $direktori = $target_path .$_FILES['filevideo']['name'];
    $direktori2 = $target_path .$_FILES['filesuara']['name'];
    move_uploaded_file($_FILES['filevideo']['tmp_name'],
    $direktori);
    move_uploaded_file($_FILES['filesuara']['tmp_name'],
    $direktori2);
    mysql_query("INSERT INTO berita(judul,
        id_kategori,
        isi_berita,
        id_user,
        jam,
        tanggal,
        hari,
        suara,
        video)
        VALUES('$judul',
        '$kategori',
        '$isi',
        '$username',
        '$jam_sekarang',
        '$tgl_sekarang',
        '$hari_ini',
        '$suara',
        '$video')");
    }
else if(!empty($suara) && !empty($gambar) && !empty($video))
{
    $direktori = $target_path .$_FILES['filesuara']['name'];
    $direktori2 = $target_path .$_FILES['filegambar']['name'];

```

```
$direktori3 = $target_path .$_FILES['filevideo']['name'];
move_uploaded_file($_FILES['filesuara']['tmp_name'],
$direktori);
move_uploaded_file($_FILES['filegambar']['tmp_name'],
$direktori2);
move_uploaded_file($_FILES['filevideo']['tmp_name'],
$direktori3);
mysql_query("INSERT INTO berita(judul,
                                id_kategori,
                                isi_berita,
                                id_user,
                                jam,
                                tanggal,
                                hari,
                                gambar,
                                suara,
                                video)
VALUES('$judul',
'$kategori',
'$isi',
'$username',
'$jam_sekarang',
'$tgl_sekarang',
'$hari_ini',
'$gambar',
'$suara',
'$video')");
}
?>
```

```

<?php
header("Content-type: text/plain");
require("../config/koneksi.php");
include "../config/library.php";

//retrieve data

//$title = trim($_GET["username"]);
$password = md5(trim($_GET["password"]));
$username = trim($_GET["username"]);
$name = trim($_GET["nama"]);
$nomor = trim($_GET["nomor"]);

//connect to the DB
//$db = connect();

//set the query to check the credentials
//$query = "SELECT username, password FROM users WHERE
username='$username' AND password='$password'";

//execute it
//$result = mysql_query($query, $db);

//if(mysql_num_rows($result) == 0)
//{
//    // wrong credentials
//    echo "Wrong username and/or password";
//    die();
//}
//else
{
//right credentials
//retrieve the current date
//$date = date("Y-m-d H:i:s");
$text = addslashes($text);
//set the query
$query = "INSERT INTO user(id_user,
                        password, nama_lengkap, nomor)
VALUES('$username',
      '$password', '$nama', '$nomor')";

//execute it
$result = mysql_query($query);
if(!$result)
{
    echo "";
}
else
{
    //echo "Post sent correctly";
    echo "Sign Up berhasil..";
}
}
//mysql_close($db);
?>

```

```

<?php
header("Content-type: text/plain");
include "config/koneksi.php";
include "config/fungsi_indotgl.php";
include "config/library.php";
$tgl = tgl_indo($t[tanggal]);
$sterkini= mysql_query("SELECT * FROM berita,user
                        WHERE user.id_user=berita.id_user
                        ORDER BY id_berita DESC LIMIT 5");
while($t=mysql_fetch_array($sterkini)){

    $tgl = tgl_indo($t[tanggal]);
    echo "$t[hari], $tgl; ";
    echo "Judul : $t[judul]; ";
    echo "Ditulis Oleh : $t[nama_lengkap]; ";

    /*
    $kalimat=strtok(nl2br($t[isi_berita])," ");
    for ($i=1;$i<=50;$i++){
        echo ($kalimat);
        echo (" "); // Spasi antar kalimat
        $kalimat=strtok(" "); // Potong per kalimat
    /*
        echo $t[isi_berita];

        echo "<hr>";
    }
?>

```

```

a:link, a:visited {
    color:#007495;
    font-weight:bold;
    text-decoration:none;
}
a:hover {
    color:#FF9900;
    font-weight:bold;
}
input, textarea, select{
    background-color: #EEEEEE;
    font-size: 10px;
}
hr {
    border-bottom: 1px dashed #9d9d9d;
}
#menu{
    font-family:Tahoma;
    font-size:12px;
    color:#ffffff;
}
#menu a:link, #menu a:visited {
    color: #ffff00;
}
#menu a:hover {
    color:#ffffff;
}
#kiri{
    font-family:Tahoma;
    font-size:11px;
    color:#ffffff;
}
#kiri a:link, #kiri a:visited {
    color: #ffff00;
}
#kiri a:hover {
    color:#ffffff;
}
.isi {
    font-family:Tahoma;
    font-size:11px;
    padding-left:4px;
    color:#333333;
    text-align:justify;
    line-height: 18px;
}
.bullet{
    font-family:Tahoma;
    font-size:14px;
    padding-left:4px;
    color:#ffffff;
    text-align:justify;
    line-height: 18px;
}
.isi_kecil{
    font-family:Tahoma;
    font-size:11px;
    padding-left:4px;
    color:#FF6600;
}
.judul {
    font-family:Tahoma;
    font-size:12px;
    font-weight:bold;
}

```

```
        color:#265180;
        padding-left:4px;
    }
    .judul_head {
        font-family:Georgia;
        font-size:12px;
        font-weight:bold;
        background-color: #B7BCC2;
        color:#333333;
        padding-left:4px;
    }
    .kembali {
        font-family:Tahoma;
        font-size:11px;
        padding-left:4px;
        text-align:center;
    }
}
```

