

## Emulasi Aplikasi Pemantauan Ruangan Melalui *Handphone* Menggunakan *Webcam*

Iwan Handoyo Putro, Petrus Santoso, Stephanie Imelda Pella

Fakultas Teknologi Industri, Jurusan Teknik Elektro, Universitas Kristen Petra

Jl. Siwalankerto 121-131, Surabaya 60236

Email: [iwanhp@petra.ac.id](mailto:iwanhp@petra.ac.id), [petrus@petra.ac.id](mailto:petrus@petra.ac.id)

### Abstrak

Makalah ini menjelaskan tentang implementasi teknologi GPRS dan Java 2 Micro Edition untuk aplikasi pemantauan ruangan melalui *handphone* menggunakan *webcam*. *Webcam* yang terhubung dengan *server* melakukan proses *capture* ruangan pada periode waktu tertentu. *Handphone* yang telah mendukung teknologi Java dan GPRS dipakai untuk mengakses hasil *capture* gambar yang telah disimpan. Pengujian dilakukan dengan menggunakan emulator Nokia 3300 dalam sebuah *local area network*. Waktu transfer rata-rata untuk gambar berukuran 36 KB pada adalah 0,9 detik. Dari hasil pengujian, disimpulkan bahwa sistem ini cocok jika dipakai untuk melakukan monitoring dengan *frame rate* yang rendah.

**Kata kunci** : pemantauan ruangan, *handphone*, *webcam*.

### Abstract

This paper describes implementation of GPRS and Java 2 Micro Edition (J2ME) technology in room monitoring application via mobile phone and webcam. The webcam, which is connected to a server, captures picture periodically. A mobile phone, which has GPRS feature and supports J2ME technology, is used to access the picture. Experiments were done using Nokia 3300 emulator in a local area network. The transfer rate time for a 36 KB picture is 0.9 second. The experiment's result shows that the system is only suitable for low frame rate room monitoring.

**Keywords**: room monitoring, mobile phone, webcam.

### Pendahuluan

Sistem pemantauan ruangan yang umum dipakai menggunakan kamera yang dihubungkan dengan televisi atau komputer untuk menampilkan hasil *capture* kamera. Sistem ini mempunyai kelemahan hasil tangkapan kamera hanya dapat diakses dari tempat yang relatif dekat. Kelemahan yang lain adalah kamera yang ada letaknya relatif tetap, tidak bisa bergerak ataupun berputar.

Untuk mengatasi masalah ini dibuat sebuah aplikasi yang mengakses hasil *capture* kamera melalui *handphone*. *Webcam* dihubungkan ke komputer *server*, *handphone* mengakses IP address komputer *server* untuk mendapatkan hasil *capture webcam*. *Webcam* diletakkan di atas sebuah robot mobil untuk memungkinkan *webcam* bergerak di sekeliling ruangan.

### Java 2 Micro Edition

Java 2 Micro Edition (J2ME) merupakan bagian dari bahasa pemrograman Java yang dirancang untuk menjalankan program-program java pada perangkat bergerak dengan jumlah memori yang kecil seperti

*handphone* dan *Personal Digital Assistant* (PDA). Kedudukan J2ME dalam arsitektur bahasa pemrograman Java dapat dilihat pada Gambar 1:



Sumber : <http://www.e-zest.net/j2me.html>

Gambar 1. J2ME dalam Bahasa Pemrograman Java

J2ME terdiri dari beberapa komponen yaitu:

- Java Virtual Machine  
Komponen ini untuk menjalankan program-program Java pada emulator atau *handheld device*.
- Java API (*Application Programming Interface*)  
Komponen ini merupakan kumpulan *library* untuk menjalankan dan mengembangkan java pada *handheld device*.

**Catatan**: Diskusi untuk makalah ini diterima sebelum tanggal 1 Desember 2005. Diskusi yang layak muat akan diterbitkan pada Jurnal Teknik Elektro volume 6, nomor 1, Maret 2006.

- *Tools* lain untuk pengembangan Berupa emulator *handphone* atau PDA yang digunakan untuk simulasi program dikomputer.

J2ME dibagi menjadi dua bagian yaitu J2ME *configuration* dan J2ME *Profiles*. J2ME *Configuration* mendefinisikan lingkungan kerja J2ME *runtime* dengan menyediakan *library* standar untuk mengimplementasikan fitur standar dari sebuah *handheld device*. Dua kategori dari J2ME *Configuration* adalah CDC (*Connected Device Configuration*) dan CLDC (*Connected Limited Device Configuration*). Perbedaan perangkat yang didukung oleh CDC dan CLDC diperlihatkan pada Gambar 2.



Sumber: [plex.coe.psu.ac.th/java/j2me/intro.html](http://plex.coe.psu.ac.th/java/j2me/intro.html)

Gambar 2. Perangkat yang didukung CDC dan CLDC

Perbedaan antara CLDC dan CDC dapat dilihat pada tabel 1.

Tabel 1. Perbandingan CDC dan CLDC

	CLDC	CDC
Implementasi J2SE	Subset	Seluruh feature
Java Virtual Machine	KVM	CVM
Memori	160 – 512 KB	Minimal 2 MB
Prosesor	16 dan 32 bit	32 bit

CLDC digunakan pada perangkat-perangkat dengan ukuran memori kecil seperti *Handphone* dan PDA. Karena kecilnya memori perangkat tempat CLDC bekerja maka beberapa fitur Java 2 yang dianggap kurang penting untuk diimplementasikan pada *handheld device* dibuang.

J2ME profile menyediakan Java *libraries* untuk implementasi tambahan yang spesifik untuk setiap *handheld device*. Lima kategori dari J2ME profile

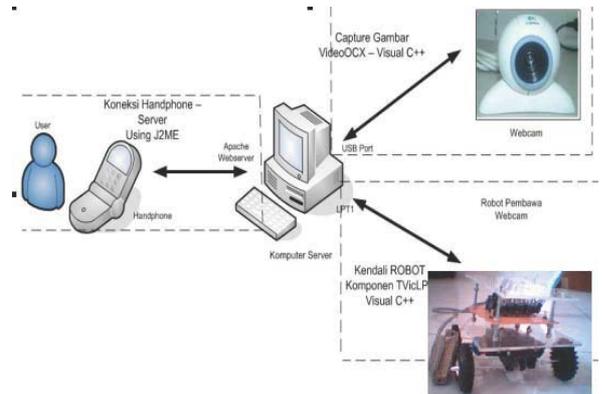
adalah : Mobile Information Device Profile (MIDP), Foundation Profile, Personal Profile, RMI Profile, Personal Digital Asistant Profile. MIDP dan Personal Digital Assistant Profile digunakan pada perangkat CLDC sedangkan Foundation Profile, Personal Profile, RMI Profile digunakan pada perangkat CDC.

### Perencanaan Sistem

Perencanaan sistem terdiri dari dua bagian yaitu pengambilan gambar *webcam* oleh komputer *server* dan pembentukan koneksi antara *handphone* dan komputer *server*. Blok diagram sistem dapat dilihat pada gambar 3.

Sistem terdiri dari:

- Komputer server yang mengendalikan webcam dan robot mobil
- Webcam dan robot mobil
- Handphone yang terhubung ke jaringan GPRS



Gambar 3. Blok Diagram Sistem

Garis besar kerja sistem adalah sebagai berikut:

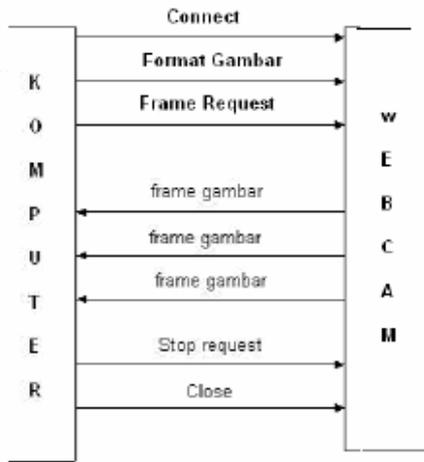
- Handphone akan menghubungi server melalui jaringan GPRS. Ada dua hal yang bisa dilakukan oleh handphone, meminta server untuk mengaktifkan webcam dan meminta sever untuk menggerakkan robot mobil
- Webcam menerima perintah dari server untuk melakukan capture kondisi ruangan dan hasilnya disimpan ke dalam server untuk kemudian dikirimkan ke handphone
- Robot mobil menerima perintah dari server dan kemudian bergerak sesuai dengan perintah yang diterima untuk mengatur posisi webcam.

### Perencanaan Pengambilan Gambar *Webcam* oleh Komputer Server

Pengambilan gambar *webcam* oleh komputer *server* menggunakan bahasa pemrograman Visual C++ dengan komponen VideoOCX [4]. Program mengakses hasil capture *webcam* menggunakan driver

bawaan *webcam* dan secara berkala menyimpannya dalam *file* gambar dengan format *Bitmap*(Bmp). Setiap *file bitmap* tersebut kemudian dikonversi ke format *Portable Network Graphic* (PNG) dan disalin ke direktori *server* sehingga dapat diakses oleh *handphone*.

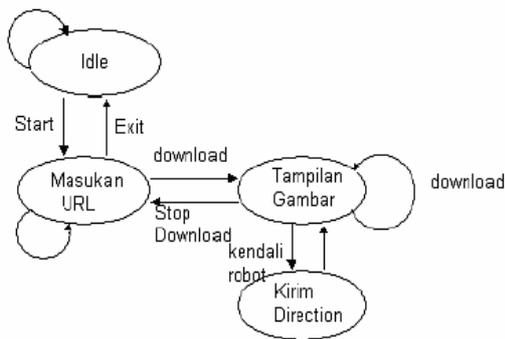
*Webcam* terhubung dengan komputer server melalui USB Port. Protokol pengaksesan *webcam* diperlihatkan pada gambar 4.



Gambar 4. Protokol Hubungan *Webcam* dengan Komputer Server

**Perencanaan User Interface**

Pengguna berinteraksi dengan *handphone* melalui 3 *interaction points* yaitu tempat memasukan URL komputer *server*, tempat menampilkan gambar dan *command* untuk mengirimkan kendali robot.

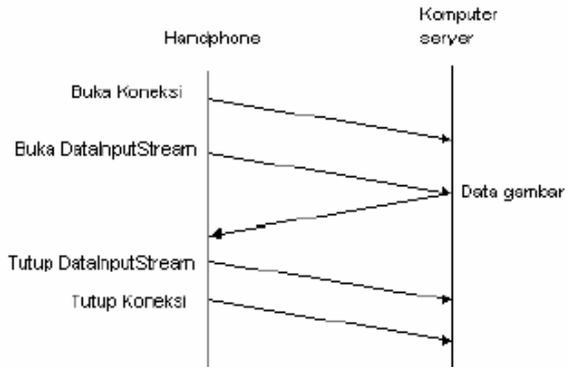


Gambar 5. State Diagram Interaksi User dengan *Handphone*

Gambar 5 mendeskripsikan protokol interaksi pengguna dengan *handphone*. Pengguna memasukan URL yang digunakan untuk mengakses komputer *server*. Kemudian, *handphone* menampilkan gambar hasil download. Pada saat itu, pengguna dapat mengirimkan perintah untuk mengendalikan robot.

**Perencanaan Koneksi Jaringan untuk *Download* Gambar**

Koneksi jaringan untuk *download* gambar menggunakan kelas *Connection*. Pembacaan gambar yang ada pada komputer *server* digunakan Data Input Stream dari paket *java.io*. Protokol proses ini diperlihatkan pada gambar 6.



Gambar 6. *Time Sequence* Diagram Koneksi Jaringan untuk *Download*

*Handphone* membuka koneksi dengan komputer menggunakan URL yang dimasukan user. Untuk mengambil data gambar digunakan sebuah *Stream*. Setelah selesai membaca dari *input stream*, *handphone* menutup *stream*. Setelah proses *download* selesai, *handphone* menutup koneksi ke komputer *server*. Data gambar yang dibaca dari *stream* berbentuk *byte array*, sebelum ditampilkan maka data harus diubah menjadi gambar terlebih dahulu.

**Perencanaan Pengiriman Perintah Kepada Robot Mobil**

Robot mobil yang dipakai untuk membawa kamera adalah robot sederhana yang digerakkan dengan dua buah motor DC. Gambar 7 menunjukkan bentuk robot mobil yang dipakai.



Gambar 7. Robot Mobil

Robot mobil dikendalikan oleh komputer server melalui port LPT1. Perintah kendali terhadap robot mobil yang diterima dari handphone disimpan dalam sebuah file teks dan kemudian dikirimkan kepada robot mobil. Perintah yang bisa dikerjakan oleh robot mobil adalah: maju, mundur, belok kiri, belok kanan. Daftar bit-bit yang dikirimkan dan perintah yang direspon bisa dilihat pada tabel 2 berikut.

Tabel 2. Perintah Pengaturan Kendali Robot Mobil oleh Komputer Server

Bit	Motor 1	Motor 2	Respon Robot
1010	Forward	Forward	Maju
0101	Reverse	Reverse	Mundur
1001	Forward	Reverse	Belok Kanan
0110	Reverse	Forward	Belok Kiri

Untuk setiap perintah dalam tabel di atas, setelah 0.5 detik komputer server akan mengirimkan bit 0000, sebagai respon robot mobil akan menghentikan putaran motor. Cara ini dipakai memperhatikan kondisi jaringan yang kadang-kadang responnya lambat, sehingga kadang-kadang perintah yang dikirim terlambat sampai. Dengan cara ini, gerakan robot mobil bisa lebih diprediksi.

### Implementasi Sistem

Dalam implementasinya ada dua program yang dihasilkan. Program pertama diletakkan dalam komputer server, dipakai untuk melayani permintaan dari handphone, mengirim perintah ke robot mobil dan mengirim perintah ke webcam. Program kedua diletakkan dalam handphone, berfungsi sebagai client dan merupakan interface terhadap user.

Program server diimplementasikan dengan menggunakan Microsoft Visual C++ dengan bantuan komponen VideoOCX [4]. Program client diimplementasikan dengan menggunakan Java 2 Micro Edition [6].

### Program Server

Ada 3 bagian penting dalam program server, yaitu mengimplementasikan fungsi sebagai server, mengendalikan webcam untuk melakukan capture gambar dan mengendalikan robot mobil untuk bergerak sesuai perintah. Bagian yang dibahas di sini adalah proses capture gambar dan pengendalian robot mobil.

Untuk melakukan proses capture gambar, perlu dilakukan inisialisasi terhadap webcam lebih dahulu. Berikut adalah kode yang dipakai untuk melakukan inisialisasi terhadap webcam.

```
m_webcam.Init(); //membuka hubungan
dengan driver webcam
m_webcam.Start(); //inisiasi proses
capture
```

```
// Siapkan memory untuk menangani
gambar
m_Image =
m_webcam.GetColorImageHandle();
// memanggil fungsi CaptureThread
AfxBeginThread(CaptureThread, this);
```

Program kemudian akan melakukan capture setiap selang waktu tertentu, hasil capture kemudian akan disimpan dalam format bitmap. Berikut adalah kode yang dipakai.

```
CCaptureDlg *dlg; // referensi ke
parent dialog
dlg = (CCaptureDlg *)pParam; //
pointer ke parent dialog
while (dlg->m_Running) // looping
{
//capture gambar
if (dlg-
>m_webcam.Capture(dlg->m_Image))
{
//menampilkan gambar pada
interface
dlg-
>m_webcam.Show(dlg-
>m_Image);
// menyimpan gambar dalam
format bmp
dlg-
>m_webcam.SaveBMP(dlg-
>m_Image, "gambar.bmp"); }
else Sleep(5);
dlg->hapusframe(); // hapus
frame dari memori Sleep(5);
//selang waktu pengambilan
0.005s
} return(0);
```

Untuk melakukan pengontrolan robot mobil, program server akan membaca file yang dikirim oleh program client. Isi file akan disesuaikan dengan perintah bit yang diperlukan kemudian dikirim melalui LPT1 kepada robot mobil. Berikut potongan kode yang dipakai.

```
if (a == "kiri"){
m_Dlg->myDlg->m_port.SetDataPort(6);
for (npin = 9; npin >=2; npin--) {
if(m_Dlg->myDlg->m_port.GetPin(npin)
== 1)
c = c + "1";
else
c = c + "0";
}
m_Dlg->myDlg-
>m_data.SetWindowText(c);
Sleep(1000); // 11 detik perputaran
motor
m_Dlg->myDlg->m_port.SetDataPort(0);
c = "";
m_Dlg->myDlg-
>m_data.SetWindowText(c);
}
```

### Program Client

Dalam program Client bagian yang paling penting adalah proses melakukan download gambar dan

menampilkannya ke layar handphone. Untuk melakukan download, diimplementasikan Class Download yang akan mengambil URL dari interface.

```
public Download(String url, Connection MIDlet)
{
    this.url = url;
    this.MIDlet = MIDlet;
}
```

Pembangunan koneksi menggunakan *interface ContentConnection* yang menyediakan *streamconnection*. Proses *download* gambar dilakukan oleh sebuah fungsi *getImage()*.

```
private void getImage(String url)
throws IOException
```

Untuk menampung data gambar yang diakses dari komputer *server* digunakan *DataInputStream*. Untuk menyimpan data yang dibaca dari *InputStream* dibuat variable *imageData* yang bertipe byte array.

```
ContentConnection connection =
ContentConnection
Connector.open(url);
DataInputStream iStrm = connection.
openDataInputStream();
byte imageData[];
```

Pembacaan data dari *InputStream* meliputi dua kondisi. Apabila panjang koneksi memungkinkan untuk menampung semua data maka data yang diakses langsung disimpan kedalam *imageData*

```
if (length != -1)
{
    imageData = new byte[length]; //
    baca data dan simpan ke array
    iStrm.readFully(imageData);
}
```

Jika data gambar melebihi panjang koneksi maka *InputStream* akan disimpan terlebih dahulu dalam sebuah variable bertipe *ByteArrayOutputStream*. Setelah seluruh *InputStream* selesai dibaca baru disimpan kedalam *imageData*.

```
bStrm = new ByteArrayOutputStream();
int ch;
while ((ch = iStrm.read()) != -1)
    bStrm.write(ch);
imageData = bStrm.toByteArray();
bStrm.close();
```

Kemudian diciptakan sebuah dari *bytearray* menggunakan fungsi *createImage* sehingga dapat ditampilkan pada *fmgambar*.

```
Image im = null;
im = Image.createImage(imageData, 0,
imageData.length);
```

Fungsi *getImage()* akan diakhiri dengan menutup koneksi apabila seluruh data gambar sudah berhasil diakses.

```
finally
{
    if (connection != null)
        connection.close();
    if (iStrm != null)
        iStrm.close();
    if (bStrm != null);
        bStrm.close();
}
```

Untuk menampilkan image ke layar handphone digunakan rutin berikut:

```
ImageItem ii = new
ImageItem(null, im,

ImageItem.LAYOUT_DEFAULT,
null);
// Jika pada form sudah ada
gambar gantikan
if (fmgambar.size() != 0)
    fmgambar.set(0, ii);
else // jika belum tambahkan
pada form
    fmgambar.append(ii);
display.setCurrent(fmgambar)
;
```

## Pengujian Sistem

Pengujian sistem dilakukan untuk mengetahui tingkat keberhasilan Aplikasi Pemantauan Ruang melalui Handphone menggunakan *Webcam*. Pengujian yang dilakukan adalah: pengujian *download* gambar dari komputer *server*, pengujian pengiriman perintah pengendalian robot ke komputer *server*, serta pengujian pengiriman bit kontrol robot oleh komputer *server*.

Spesifikasi komputer yang dipakai adalah:

Komputer 1: P4 1.8 GHz, RAM 128 MB, Windows 2000 Profesional.

Komputer 2: P III 800 MHz, RAM 128 MB, Windows 98.

Jaringan komputer yang dipakai adalah jaringan komputer Laboratorium yang terhubung ke Internet. IP address yang dipergunakan adalah 192.168.12.104 dan 192.168.12.105.

## Pengujian Download Gambar dari Komputer Server

Sampai saat ini pengujian pengaksesan gambar hasil *capture webcam* dilakukan dari emulator *handphone*. Pengaksesan gambar dari *handphone* baru berhasil dilakukan untuk gambar dengan ukuran *file* lebih kecil dari 1 kB. Hal diperkirakan disebabkan oleh keadaan jaringan provider selular serta bandwidth jaringan tempat *server* berada tidak memungkinkan untuk dilakukan secara real.

Pada pengujian pengaksesan gambar melalui *handphone*, gambar yang digunakan adalah gambar PNG yang sudah dibuat bukan gambar hasil *capture webcam*. Pengujian pengaksesan gambar diperlihatkan pada gambar 9.



Gambar 9. Pengujian Pengaksesan Gambar dengan *Handphone*

Pada pengujian *download* gambar hasil *capture webcam* penulis menggunakan emulator bawaan J2ME serta emulator Nokia 3300 yang dapat *download* dari <http://www.devx.com/Nokia/RSSFeed/21160>. Pengujian dilakukan dari *local server* (emulator dan *webcam* berada pada satu komputer) maupun melalui koneksi internet.

Pada gambar 10 ditunjukkan pengujian pengambilan gambar *webcam* selama selang waktu tertentu, menggunakan emulator J2ME dengan resolusi *webcam* 128x96 pixel. Pengujian dilakukan melalui *local area network* dengan spesifikasi *server* dan *client* pada komputer 1.



Gambar 10. Pengambilan Gambar *Webcam* dalam Selang Waktu Tertentu

Pengujian selang waktu yang diperlukan untuk mengakses sebuah frame sejak diambil *webcam* sampai ditampilkan di emulator yang dilakukan melalui *local area network*. Pengukuran selang waktu dilakukan dengan mengarahkan *webcam* pada dua jenis keadaan yang kontras, misalnya pada mulanya *webcam* ditutupi benda tertentu sehingga hasil pengambilan gambarnya hanya berupa tampilan berwarna hitam, kemudian benda diangkat dengan cepat sehingga *webcam* kembali mengambil gambar

situasi ruangan. Dengan demikian mempermudah membedakan dua frame yang berurutan.

Hasil pengujian waktu transfer gambar dengan resolusi 128 x 96 pixel diperlihatkan pada tabel 2.

Tabel 3. Waktu Transfer Gambar melalui Local Area Network

No	Percobaan	Waktu Transfer (Detik)	Ukuran File (KB)
1	1.112	38	
2	0.899	38	
3	1.098	38	
4	1.123	38	
5	1.121	38	
6	0.768	37	
7	0.756	37	
8	0.756	37	
9	0.89	38	
10	0.892	38	

Tabel 3 menunjukkan selang waktu untuk transfer gambar berkisar antara 0.756-1.1 detik dengan waktu rata-rata 0,942 detik. Data ini menunjukkan faktor delay jaringan tidak terlalu berpengaruh terhadap tampilnya gambar di emulator *handphone*.

## Kesimpulan

Berdasarkan perancangan dan pengujian sistem dapat diambil kesimpulan sebagai berikut:

- *Download* gambar hasil *capture webcam* berhasil dilakukan dengan emulator melalui jaringan.
- Pemilihan ukuran resolusi gambar *webcam* bergantung pada tiga faktor yaitu ukuran pixel gambar yang didukung *handphone*, ukuran file gambar yang ditransferkan serta kualitas gambar yang diinginkan.
- Sistem cocok dipakai jika monitoring dilakukan dengan frame rate yang rendah.

## Daftar Pustaka

- [1] Morrison, Michael. *Sams Teach Yourself Wireless Java with J2ME in 21 Days*. New York Sams, 2003.
- [2] Keogh, James. *J2ME: The Complete Reference*. New York: Osborne/New York, 2001.
- [3] Williams, Mickey, David Bennet. *Visual C++ 6 Unleashed*. Indiana: Sams, 2000.
- [4] VideoOCX Component, URL: <http://www.videoocx.de>
- [5] Nokia 3300 Emulator, URL: <http://www.devx.com/Nokia/RSSFeed/21160>
- [6] Java 2 Platform Micro Edition, URL: <http://java.sun.com/j2me/index.jsp>