

Data yang dikirim oleh pengirim ke penerima akan melalui beberapa tahap proses salah satunya adalah proses koreksi kesalahan data. Salah satu metode dalam koreksi kesalahan adalah metode deteksi dan koreksi kesalahan dengan Hamming Code. Metode Hamming Code dapat digunakan untuk membuat sandi baru dan juga untuk mengoreksi data yang dikirim. Dalam perancangan aplikasi ini, penulis membuat aplikasi yang dapat membuat sandi baru dan juga untuk koreksi data.

**Ladangan Teori**

Algoritma Hamming Code dikemukakan oleh Richard W. Hamming pada tahun 1940-an. Metode koreksi kesalahan dengan menggunakan Hamming Code merupakan suatu metode pendeteksian kesalahan dengan menambahkan data dengan suatu kode, biasanya disebut paritas. Algoritma Hamming Code merupakan salah satu algoritma pendeteksi error (error detection) yang mampu untuk mendeteksi beberapa error, namun hanya mampu mengoreksi satu error (single error correction).[3] Data yang dikirimkan berupa data asli dan data yang mengandung parity.

Jumlah parity tergantung dari panjangnya data yang akan dikirimkan. Jika suatu data memiliki nilai dari perpangkatan 2 (misal  $2^y$ ), maka untuk menentukan jumlah parity nya dapat dilakukan dengan cara:

$$\text{parity} = (y + 1)$$

Jika data yang dimiliki adalah 4, maka jumlah parity ada 3, dan jika data ada 16, maka jumlah parity ada 5. Dengan kata lain, jika pengirim mengirimkan 4 bit, maka data yang akan dikirim menjadi 7 bit, dengan 3 bit merupakan bit parity, begitu juga jika pengirim akan mengirimkan data sebanyak 16 bit, maka banyak data

yang dikirim sebenarnya adalah 21 dengan 5 bit tambahan merupakan bit parity. Untuk menentukan letak parity, maka dapat ditemukan dengan menggunakan persamaan

$$P_i = 2^{i-1}$$

Sehingga posisi parity dapat dilihat pada tabel dibawah.

Tabel 1 Posisi Parity

Parity	Posisi
P <sub>1</sub>	1
P <sub>2</sub>	2
P <sub>3</sub>	4
P <sub>4</sub>	8
P <sub>5</sub>	16
P <sub>6</sub>	32

Proses deteksi kesalahan dengan menggunakan *Hamming Code*, yaitu:

- Menentukan jumlah data yang akan dideteksi/dikirim
- Menentukan letak parity (di posisi 1,2,4, dst)
- Menentukan nilai dari setiap parity
- Menggabungkan nilai bit parity dengan bit data yang ada

Proses koreksi error dengan menggunakan *Hamming Code*, yaitu:

- Hitung jumlah bit data yang dikirimkan
- Tandai posisi setiap parity pada data bit tersebut
- Lakukan operasi XOR pada setiap parity
- Jika nilai output dari parity tidak sama dengan input, maka terjadi error
- Mengkonversikan hasil output parity ke bilangan desimal sehingga akan diketemukan letak data yang salah

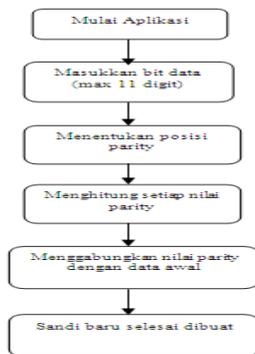
**Perancangan Simulasi**

Perancangan aplikasi simulasi deteksi dan koreksi menggunakan metode *Hamming Code* terbagi dalam dua kategori. Kategori pertama adalah pembuatan sandi baru pada saat data

akan dikirimkan. Pembuatan sandi baru dapat dilakukan dengan beberapa tahapan, yaitu:

- a. Memasukkan bit data maksimal 11 digit
- b. Menentukan posisi parity 1, parity 2, parity 3, dan parity 4 pada data yang telah dimasukkan
- c. Menentukan nilai dari setiap parity
- d. Menggabungkan data parity dengan data awal
- e. Pembuatan sandi baru selesai

Untuk lebih jelasnya, perancangan simulasi untuk pembuatan sandi baru dapat dilihat pada diagram dibawah.



Gambar 2. Diagram pembuatan sandi baru

Untuk proses deteksi dan koreksi kesalahan dengan menggunakan *Hamming Code* dilakukan dengan beberapa tahapan, yaitu:

- a. Memasukkan bit data maksimal 15 digit
- b. Menentukan posisi parity 1, parity 2, parity 3, dan parity 4 pada data yang telah dimasukkan
- c. Menentukan nilai dari setiap parity
- d. Menentukan data yang benar dan yang salah dari data yang telah diinputkan
- e. Menentukan letak data yang salah
- f. Melakukan koreksi data berdasarkan data yang diinputkan

Untuk lebih jelasnya, perancangan simulasi untuk deteksi dan koreksi kesalahan dapat dilihat pada diagram dibawah.

Berikut adalah potongan code untuk proses pembuatan sandi baru dan proses deteksi dan koreksi kesalahan menggunakan *Hamming Code*.

```

public class Cek extends ActionBarActivity {
    Button button;
    EditText data1;
    EditText data2;

    TextView txt;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.cek_satu);

        final EditText ed=(EditText)findViewById(R.id.ed);
        final EditText ed1=(EditText)findViewById(R.id.ed1);
        final EditText ed2=(EditText)findViewById(R.id.ed2);
        final EditText ed3=(EditText)findViewById(R.id.ed3);
        final EditText ed4=(EditText)findViewById(R.id.ed4);
        final EditText ed5=(EditText)findViewById(R.id.ed5);
        final EditText ed6=(EditText)findViewById(R.id.ed6);
        final EditText ed7=(EditText)findViewById(R.id.ed7);
        final EditText ed8=(EditText)findViewById(R.id.ed8);
        final EditText ed9=(EditText)findViewById(R.id.ed9);
        final EditText ed10=(EditText)findViewById(R.id.ed10);
        final EditText ed11=(EditText)findViewById(R.id.ed11);
    }
}
    
```

Gambar 3 Potongan source code untuk membuat sandi baru

### Hasil Dan Pembahasan

#### Pembuatan sandi baru

- Masuk ke aplikasi



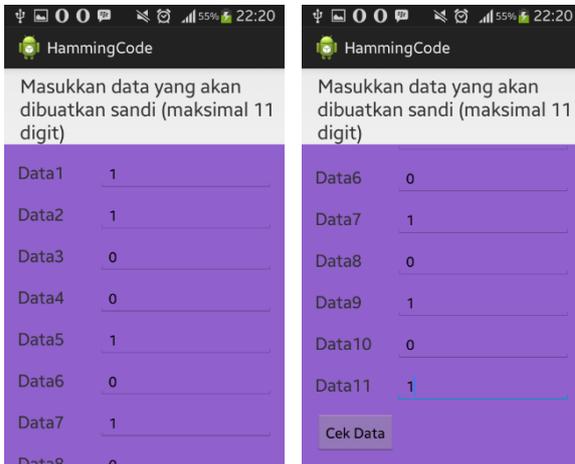
Gambar 4 Halaman awal aplikasi

- Pilih menu Buat Sandi Baru



Gambar 5 Halaman pemilihan menu

- Masukkan maksimal 11 data bit untuk pembuatan sandi baru



Gambar 6 Halaman pembuatan sandi baru dengan memasukkan data maksimal 11 bit

- Penentuan posisi parity sebagai data tambahan



Gambar 7 Penentuan parity

- Penentuan sandi baru untuk menjadi data



Gambar 8 Pembuatan sandi baru

### Deteksi dan koreksi kesalahan

1. Masuk ke aplikasi



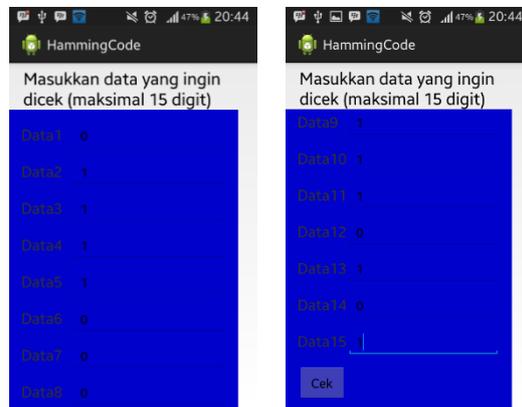
Gambar 9 Halaman awal aplikasi

2. Pilih menu Koreksi Kesalahan



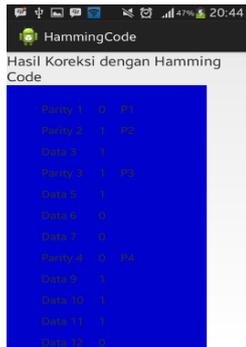
Gambar 10 Halaman pemilihan menu

3. Masukkan maksimal 15 data bit untuk dicek data



Gambar 11 Halaman koreksi kesalahan dengan memasukkan maksimal 15 bit data

4. Penentuan parity pada setiap data yang telah dimasukkan



Gambar 12 Halaman penentuan parity dari data awal

5. Penentuan jumlah parity dan posisi kesalahan pada data



Gambar 13 Halaman hasil deteksi dan koreksi kesalahan

## Kesimpulan

1. Data yang dikirim oleh pengirim akan berbeda dengan data yang diterima oleh penerima dikarenakan pada penerima sudah mendapatkan sandi baru.
2. Pembuatan sandi baru menambah panjang bit data dari pengirim, sehingga semakin banyak data yang dikirim, maka penambahan data bit untuk sandi barupun akan semakin banyak.
3. Pada deteksi dan koreksi kesalahan, bisa mendeteksi berdasarkan posisi data yang salah dan melakukan perbaikan pada data tersebut.

## Daftar Pustaka

1. Fiedler James, "Hamming Code.pdf"
2. Forouzan, Behrouz, "Data Coomunication and Networking 4th edition.pdf".
3. Hutauruk, Sindak, "Perancangan simulasi koreksi kesalahan data dengan metode FEC pada komputer berbasis Visual Basic.pdf".
4. Purnomo, Galih, "Metode Hamming".
5. Tanenbaum, "Computer\_network\_4th\_edit ion.pdf".
6. Xie, Yao, "Lecture 15: Hamming codes and Viterbi algorithm.pdf".

**APLIKASI PENGAWASAN KOMPUTER BERBASIS SERVER-CLIENT**

Muhammad Masykur, Irman Nurichsan  
Prodi Teknik Elektro  
Universitas Garut

**Abstrak**

Penelitian ini bertujuan merancang dan membuat aplikasi yang mampu mengawasi proses perjalanan aplikasi dalam setiap komputer. Manfaat sistem ini adalah membantu pengawas untuk mengontrol para pengguna komputer yang terhubung ke dalam jaringan dalam mengawasi perjalanan aplikasi pada jam-jam tertentu. Pengembangan aplikasi dimulai dengan mengidentifikasi kebutuhan sistem. Kebutuhan sistem digunakan untuk desain sistem yang dirancang. Selanjutnya pembuatan aplikasi dibangun menggunakan Visual Basic 6.0. Terakhir, dilakukan uji coba dengan variasi skenario yang bertujuan untuk mengetahui skenario keberhasilan dan hasil yang ditampilkan. Proyek akhir ini menghasilkan Aplikasi Pengawasan Komputer Berbasis Jaringan (Menggunakan Visual basic 6.0). Fungsi utama aplikasi ini pada Remote Process yang mampu menghentikan aplikasi yang sedang berjalan, serta fungsi Remote File yang mampu memonitor data-data yang ada dalam komputer client.

**Kata Kunci:** jaringan, pengawasan, Visual Basic

**Pendahuluan**

Dalam penggunaan komputer *non personal* terdapat suatu aturan penggunaan komputer yang merupakan otoritas setiap instansi atau organisasi yang membawahi jaringan komputer itu sendiri sehingga aturan penggunaan komputer bukan merupakan suatu ketetapan yang berlaku umum dan general, namun ada satu benang merah diantara semua peraturan tersebut yaitu tidak merugikan pihak instansi atau organisasi tersebut dalam penggunaan fasilitas komputer yang telah disediakan seperti penukaran hardware milik instansi dengan milik pribadi tanpa sepengetahuan pihak instansi atau main game/*browsing* internet pada waktu jam kerja/belajar.

Namun pada kenyataannya aturan saja tidak dapat menangani permasalahan yang ada. Suatu aturan tidak dapat menjamin semua orang

akan patuh terhadap aturan tersebut, karena tidak semua orang mempunyai rasa taat dan patuh terhadap peraturan yang ada. Selain itu, aturan saja tidak dapat menindaklanjuti para pelanggar aturan, sehingga diperlukan suatu unsur lain yang dapat dijadikan sebagai pendukung proses pengawasan terhadap penggunaan fasilitas komputer *non personal* berdasarkan peraturan yang berlaku yang disebut dengan pengawas. Dalam hal ini seorang pengawas bertugas melakukan pengawasan terhadap semua penggunaan fasilitas komputer apakah sesuai dengan aturan yang ada di instansi tersebut.

Dengan menggunakan teknologi jaringan komputer dapat memberikan alternatif lain terhadap sistem pengawasan penggunaan fasilitas komputer. Salah satu cara yang ditawarkan ialah dengan mengawasi

komputer-komputer (komputer karyawan, misalnya) yang terhubung pada suatu jaringan oleh suatu aplikasi yang telah ditanam di dalam komputer-komputer tersebut, kemudian mengirimkan hasil pengawasannya kepada suatu komputer (komputer staf pengawas, misalnya) yang di dalamnya sudah tertanam suatu aplikasi pusat yang dapat menampung hasil pengawasan tersebut lalu menampilkannya.

### **Ladangan Teori**

**Pengertian Jaringan Komputer**  
 Jaringan komputer adalah sebuah kumpulan komputer dan peralatan lainnya yang terhubung. Informasi dan data bergerak melalui kabel-kabel sehingga memungkinkan pengguna jaringan komputer dapat saling bertukar dokumen dan data. Tiap komputer yang terhubung dengan jaringan disebut titik atau lebih sering disebut dengan node. Sebuah jaringan komputer dapat memiliki dua, puluhan, ribuan atau bahkan jutaan node. Sebuah jaringan biasanya terdiri dari dua atau lebih komputer yang saling berhubungan diantara satu dengan yang lain, dan saling berbagi sumber daya misalnya CD-ROM, Printer, pertukaran file, atau memungkinkan untuk saling berkomunikasi secara elektronik. Komputer yang terhubung tersebut, dimungkinkan berhubungan dengan media kabel, saluran telepon, gelombang radio, satelit, atau sinar infra merah.

### **Protokol**

Protokol adalah aturan-aturan main yang mengatur komunikasi diantara beberapa komputer di dalam sebuah jaringan, aturan itu termasuk di

dalamnya petunjuk yang berlaku bagi cara-cara atau metode mengakses sebuah jaringan, topologi fisik, tipe-tipe kabel dan kecepatan transfer data.

### **FDDI**

*Fiber Distributed Data Interface* (FDDI) adalah sebuah Protokol jaringan yang menghubungkan antara dua atau lebih jaringan bahkan pada jarak yang jauh Metode aksesnya yang digunakan oleh FDDI adalah model token. FDDI menggunakan dua buah topologi ring secara fisik. Proses transmisi biasanya menggunakan satu buah ring, namun jika ada masalah ditemukan akan secara otomatis menggunakan ring yang kedua.

### **ATM**

ATM adalah singkatan dari *Asynchronous Transfer Mode* (ATM) yaitu sebuah protokol jaringan yang mentransmisikan pada kecepatan 155 Mbps atau lebih. ATM mentransmisikan data kedalam satu paket dimana pada protokol yang lain mentransfer pada besar-kecilnya paket. ATM mendukung variasi media seperti video, CD-audio, dan gambar. ATM bekerja pada model topologi Bintang dengan menggunakan Kabel fiber optic ataupun kabel *twisted pair*. ATM pada umumnya digunakan untuk menghubungkan dua atau lebih LAN. Dia juga banyak dipakai oleh *Internet Service Providers* (ISP) untuk meningkatkan kecepatan akses Internet untuk klien mereka.

### **Perancangan Simulasi**

Analisa Requirement akan dijelaskan mengenai kebutuhan-kebutuhan yang harus dipenuhi oleh APK yang terdiri dari :

- Perspektif Produk; Pada bagian ini akan dijelaskan mengenai APK beserta ruang lingkup penggunaannya.

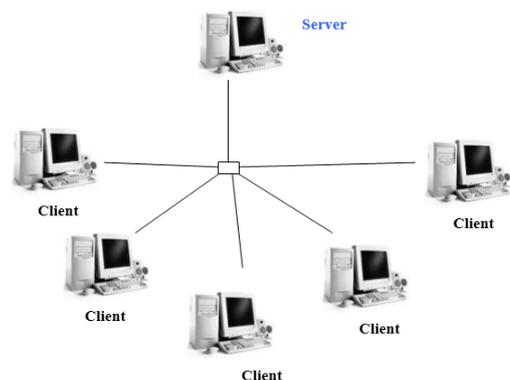
- Analisa Karakteristik Pengguna Sistem; Pada bagian ini akan dijelaskan mengenai karakteristik pengguna yang dapat menggunakan APK.
- Analisa Kebutuhan Sistem; Pada bagian ini akan dijelaskan mengenai spesifikasi kebutuhan sistem.
- Capability; Pada bagian ini berisi mengenai kemampuan yang dimiliki oleh APK.
- Mekanisme Fungsi; Pada bagian ini akan dijelaskan mengenai mekanisme yang dilakukan oleh setiap fungsi-fungsi APK.

APK dirancang sebagai interface antara seorang pengawas pada suatu jaringan komputer dengan sistem manajemen yang dimiliki Windows sehingga seorang pengawas dapat mengetahui aplikasi yang dijalankan oleh pengguna pada komputer tertentu yang diawasi di dalam jaringan yang sama. APK juga memiliki fitur lain yaitu eksekusi *file*, penghapusan *file*, *download* dan *upload file*, dan akses kontrol program **Winamp** yang sedang dijalankan pada komputer *server*. Dengan demikian pengawas dapat mengetahui aplikasi apa saja yang dijalankan oleh user, *file-file* apa saja yang terdapat pada komputer pengguna tanpa melakukan *sharing* seperti halnya yang harus dilakukan dengan memakai fasilitas pada OS **Windows** jika akan berbagi file dalam jaringan. Fitur lainnya adalah kemampuan dalam mematikan aplikasi yang sedang dijalankan di komputer tertentu yang diawasi secara *remote* oleh *server*. APK dibagi kedalam dua executable aplikasi, yaitu **mServer** dan **mClient**.

- **mServer** adalah aplikasi yang dipasang (*di-install*) pada komputer yang ditujukan sebagai komputer untuk pusat

pengawasan yang terhubung pada suatu jaringan komputer. Komputer yang ditanami aplikasi ini disebut *Server*.

- **mClient** adalah aplikasi yang dipasang (*di-install*) pada komputer yang ditujukan sebagai komputer yang diawasi dan terhubung pada suatu jaringan komputer. Komputer yang ditanami aplikasi ini disebut *Client*.



Gambar 1. Gambaran mengenai sistem APK

### Hasil Dan Pembahasan

Ada beberapa tools yang digunakan untuk merancang Aplikasi Pengawasan Komputer yang terdapat pada komponen-komponen yang disediakan oleh Visual Basic itu sendiri. Untuk dapat menggunakan semua tools yang dibutuhkan dalam perancangan aplikasi ini, maka ada sebagian komponen yang harus diaktifkan terlebih dahulu, yaitu :

- Microsoft Common Dialog Control 6.0
- Microsoft Tabbed Dialog Control 6.0
- Microsoft Windows CommonControl 5.0 (SP2)
- Microsoft Windows CommonControl 6.0 (SP6)

Dalam pengembangan program Aplikasi Pengawasan Jaringan menggunakan bahasa pemrograman

Microsoft Visual Basic 6.0, yang dalam pengujian program terdiri dari pengujian koneksi jaringan, terminate aplikasi, remote file manager, pemakaian lebih dari dua perangkat, pengujian ketahanan program.

Identifikasi dan Rencana Pengujian Sebelum melakukan pengujian dilakukan identifikasi hal yang akan diuji dan rencana pengujiannya. Hal ini dilakukan supaya perangkat lunak yang dibuat dapat terukur berdasarkan input yang dimasukan dan output yang diharapkan.

### Kesimpulan

Program Aplikasi Pengawasan Komputer ini dapat mengawasi penggunaan aplikasi yang tidak diperkenankan pada waktu-waktu tertentu. Aplikasi ini juga bermanfaat untuk mengatur data-data yang tersedia pada komputer *client*.

Dengan dibuatnya aplikasi ini, dapat mempermudah para pengelola komputer yang terhubung kedalam jaringan untuk dapat me-monitor kegiatan-kegiatan penjalanan aplikasi yang terjadi dengan mudah. Aplikasi ini juga didukung fasilitas **Control Winamp** yang diperuntukan bagi para pengguna komputer *client* untuk dapat memutar lagu favorit dan mengontrol intensitas *volume* agar tidak mengganggu konsentrasi pada aktivitasnya masing-masing tetapi juga tidak membuat suasana menjadi jenuh.

### Daftar Pustaka

1. Dadang Gunawan, Filbert Hilman Juwono. *Pengolahan Sinyal Digital Dengan Pemograman Matlab*. Cetakan Pertama 2012. Yogyakarta.
2. Gunaidi Abdia Away. *The Shortcut Programing*. Cetakan Pertama Juni 2006. Bandung Informatika
3. Ir. Hartono Tanudjaja, M.Kom. *Pengolahan Sinyal Digital Dan Pemrosesan Sinyal*. Cetakan Pertama 2007. Yogyakarta
4. Fiedler James, "Hamming Code.pdf"