

APLIKASI PENCARIAN DOKUMEN *IMAGE* BERDASARKAN METADATA *FILE* DILENGKAPI FASILITAS PEMESANAN CETAK *ONLINE* PADA STUDIO FOTO *DIGITAL*

Fatikha Floressya A ²⁾, Soetam Rizky Wicaksono ¹⁾

¹⁾ Program Studi Sistem Informasi, STIKOM Surabaya

²⁾ Program Studi Sistem Informasi, STIKOM Surabaya, email: soetam@stikom.edu

Abstract: Due to the vast amount of image files, a digital photo studio requires a suitable search method. Photo searching process needs more than file name. Thus, they must apply better searching method. The method that we used in this research is utilizing EXIF, an image's metadata. EXIF it self will be the primary data in photo searching searches. Searches will be performed recursively and the program execution processes are broken down into different threads which can run concurrently (otherwise known as multithreading). This threading method fasten the searching process for abundant amount of photo files.

Keywords: Searching, Metadata, EXIF, Multithreading, Recursive

Pada saat ini, produk kamera *digital* telah membanjiri pasar elektronika Indonesia. Kamera *digital* meniadakan beban biaya film, sehingga pengoperasiannya dirasakan lebih murah dibandingkan kamera film. Namun, harga peralatan kamera *digital* masih relatif tinggi, sehingga tidak semua orang dapat membeli kamera *digital* (Chandra, 2003). Sebagai gantinya banyak bermunculan studio foto *digital* yang menawarkan pelayanan dalam bentuk pemotretan dengan kamera *digital* yang dilakukan di studio foto tersebut. Adapun pemotretan dengan kamera *digital* menghasilkan *file* yang dapat disimpan ataupun dihapus, untuk kemudian dilakukan pemotretan ulang bila hasilnya kurang memuaskan, sebelum pelanggan memutuskan untuk mencetaknya.

Djonie Studio, yang beralamat di Jl. Letjen. Panjaitan 91 Bondowoso, mencoba untuk tetap kompetitif di bisnis studio foto dengan meningkatkan pelayanannya terhadap pelanggan. Untuk itu, diperlukan adanya penerapan unsur sistem informasi dalam mencapai tujuan tersebut. Dan untuk memperluas usahanya, juga diperlukan layanan *online* via *website*. Di sinilah peran teknologi informasi itu tampak. Melalui *website*, pelanggan tersebut tidak harus datang pada tempat studio foto itu berada untuk mendapatkan pelayanan. Sistem informasi studio foto *digital* ini mencakup pelayanan *offline* yang berupa manajemen studio foto *digital* dan pencarian gambar

(*image searching*) serta pelayanan *online* pemesanan cetak gambar secara *online* (*online printing*).

Pada fasilitas pencarian gambar, diperlukan sebuah metode yang dapat mempercepat proses pencarian. Hal ini dikarenakan banyaknya *file* foto yang dimiliki sebuah studio foto. Pencarian akan semakin menyulitkan bila ada pelanggan yang meminta *file*-nya dicetak kembali setelah sekian lama pemotretan dilakukan. Selain itu, pada studi kasus kali ini masalah dalam pencarian juga muncul karena *file* yang disimpan tidak terorganisir dengan baik, butuh waktu lama untuk menelusuri *folder* demi *folder*, yang juga tidak terorganisir, untuk menemukan *file* yang dimaksud.

Dari masalah inilah muncul sebuah metode untuk melakukan pencarian *file* gambar bukan berdasarkan nama *file*, atau kata kunci tertentu, melainkan melalui metadata *file* gambar itu sendiri. *Metadata* yang dimaksud di sini adalah *Exchangeable image file format (Exif)*. Telah banyak *software* yang mampu membaca, mengganti nama, dan mengekspor *metadata Exif*. Tetapi masih belum ada yang mampu menangani pencarian berdasarkan *metadata* secara rekursif, hal ini menimbulkan lambatnya proses pencarian dokumen *image*. Karena itu, perlu dibuat aplikasi yang dapat menangani pencarian secara rekursif berdasarkan *metadata Exif*. Untuk mempercepat proses pencarian ini digunakan konsep *multithreading* yang berguna

untuk memecah proses eksekusi program kedalam thread-thread yang berbeda yang dapat berjalan secara bersamaan.

Dengan pembuatan aplikasi pencarian gambar ini, diharapkan proses pencarian gambar pada studio foto menjadi lebih cepat. Kelebihan lain dari aplikasi ini adalah pengguna dapat mencari gambar tidak hanya berdasarkan nama *file*, atau kata kunci tertentu, melainkan dengan *metadata Exif* gambar itu sendiri. Selain itu, pelayanan *online printing* diharapkan dapat memperluas dan mempermudah pelayanan cetak foto dari Studio Foto Djonie, karena pelanggan tidak perlu mendatangi studio foto jika ingin mencetak fotonya.

LANDASAN TEORI

Exif adalah sebuah standar yang menjadi jembatan antara kamera dan *printer*. Standar ini memberikan keterkaitan antara informasi (atau atribut) pada saat pengambilan foto yang kemudian bisa dibaca, digunakan dan diolah oleh *printer* untuk menghasilkan foto yang sesuai dengan saat pengambilan.

Exif merupakan singkatan dari *Exchangeable image file format*, yang diperkenalkan sebagai standar terbuka oleh *Japan Electronics Industries Development Association (JEIDA)* pada tahun 1995. Pada saat itu versi 1.0 hanya berorientasi pada produsen kamera. Kemudian dikembangkan terus sehingga pada tahun 1998 keluar versi 2.0 dan tidak lama kemudian keluar versi 2.1 yang banyak dianut oleh para produsen kamera *digital* dan *printer*. Bulan Februari tahun 2002 keluar versi *Exif 2.2* yang ditahbiskan oleh *Japan Electronics and Information Technology Industries Association (JEITA)*. Standar ini sudah berhasil menjadi jembatan antara kamera dan *printer* dan sering disebut dengan *Exif Print*.

Para produsen yang mendukung *Exif* antara lain: Canon, Nikon, Olympus, FujiFilm, Minolta, Pentax, Kodak, HP, Sony, Konica, Kyocera, Casio, Toshiba, Panasonic, Sanyo, Ricoh, Sharp, JVC, Mitsubishi, dan juga Epson (Kertajaya, 2003).

Spesifikasi *Exif* mengidentifikasi metode *record* data gambar dalam *file* serta hal-hal sebagai berikut (JEITA, 2002):

1. Struktur dari data *file* gambar.
2. *Tag* yang digunakan oleh secara standar
3. Definisi dan manajemen versi dari format.

Pada Tabel 1 disajikan beberapa tag *Exif*. Komputer bisa melakukan operasi secara *paralel*, seperti *compile* program, mencetak *file* dan menerima *e-mail* melalui jaringan. Bahasa pemrograman biasanya mendukung bagian kecil dari struktur kontrol yang memungkinkan *programmer* melakukan satu aksi sekali waktu, lanjut ke aksi berikutnya hanya jika aksi sebelumnya selesai. *NET Framework Class Library* membuat operasi *paralel* primitif tersedia untuk *programmer*. *Programmer* bisa membuat sebuah aplikasi memiliki *thread* di mana setiap

Tabel 1 Tag *Exif*

Nama Tag Exif	Keterangan
Aperture	Diafragma dari lensa.
Artist	Merekam nama dari pemilik kamera, fotografer atau pembuat gambar.
Copyright	Informasi Copyright.
Date time digitized	Tanggal dan waktu ketika gambar disimpan sebagai data digital.
Date time last modified	Tanggal dan waktu ketika gambar terakhir dimodifikasi
Date time Original	Tanggal dan waktu ketika gambar asli diambil dengan kamera.
Description	Deskripsi gambar
Maker	Pembuat DSC, scanner, video digitizer atau alat perekam gambar lainnya.
Model	Nama model atau nomor model dari peralatan yang membuat gambar
Flash Mode	Status lampu cahaya ketika foto dijepret.
Focal Length	Focal length sebenarnya dari lensa, dalam satuan mm.
Height	Jumlah baris dari data gambar.
ISO	Mengindikasikan kecepatan ISO
Light Source	Jenis sumber cahaya yang dimanfaatkan.
Orientation	Orientasi gambar dalam bentuk baris dan kolom.
Resolution X	Jumlah pixel tiap <i>ResolutionUnit</i> pada <i>ImageWidth</i> .
Resolution Y	Jumlah pixel tiap <i>ResolutionUnit</i> pada <i>ImageLength</i> .
Software	Merekam nama dan versi dari software atau firmware dari kamera atau image input device yang digunakan untuk mengambil gambar.
Subject Distance	Jarak ke subjek foto, dalam satuan meter.
Title	Judul gambar
Width	Jumlah kolom dari data gambar.

thread di desain untuk menjalankan sebagian dari program secara bersamaan dengan *thread* yang lain. Kemampuan ini disebut *multithreading* (Deitel, 2002).

Aplikasi *Visual Basic* bisa menjalankan tugas yang berbeda secara simultan menggunakan *multithreading*, yaitu proses di mana masing-masing tugas dieksekusi pada *thread* yang terpisah. *Multithreading* meningkatkan performa dan responsivitas aplikasi. Aplikasi yang menggunakan *multithreading* lebih responsif terhadap *input* dari *user* karena *user interfacenya* tetap aktif ketika tugas

intensif prosesor dieksekusi pada *thread* yang terpisah (MSDN, 2005). Cara yang paling *reliable* untuk menciptakan aplikasi *multithreading* adalah menggunakan komponen *BackgroundWorker*. *Namespace* dari *class BackgroundWorker* adalah *System.ComponentModel*.

Class BackgroundWorker mengatur *thread* terpisah yang ditujukan untuk memproses metode yang telah dispesifikasikan. Untuk memulai operasi di *background*, buatlah *BackgroundWorker* dan perhatikan *event* yang melaporkan *progress* dari operasi dan berilah sinyal ketika operasi selesai. Objek *BackgroundWorker* bisa dibuat secara *programming*, atau dengan melakukan proses *drag* objek ini ke *form* dari *tab Components* dari *Toolbox*. Jika *BackgroundWorker* dibuat pada *FormsDesigner*, maka akan muncul pada *Component Tray*, dan propertinya akan ditampilkan pada jendela *Properties*.

Untuk mengatur *background operation*, tambahkan *event handler* untuk *event DoWork*. Panggil operasi yang *time-consuming* pada *event handler* ini. Untuk memulai operasi, panggil *RunWorkerAsync*. Untuk menerima pemberitahuan tentang *update* dari *progress*, *handle event ProgressChanged*. Untuk menerima pemberitahuan tentang kapan operasi selesai, *handle event RunWorkerCompleted*.

Method yang meng-*handle event ProgressChanged* dan *RunWorkerCompleted* bisa mengakses *user interface* dari aplikasi, karena *event* tadi dibangkitkan pada *thread* yang memanggil *method RunWorkerAsync*. bagaimanapun *event handler DoWork* tidak dapat memanipulasi objek pada *user interface* karena berjalan pada *background thread*.

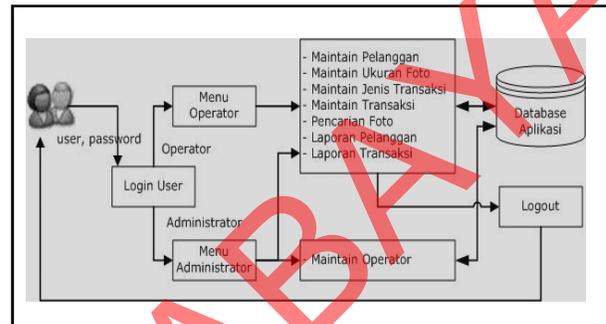
PEMBAHASAN

Aplikasi yang dibangun untuk mencakup pelayanan *offline* yang berupa manajemen studio foto *digital* dan pencarian gambar (*image searching*) serta pelayanan *online* pemesanan cetak gambar secara *online* (*online printing*). Adapun perangkat lunak yang dibuat nanti tidak terintegrasi antara aplikasi *web online printing* dan aplikasi yang dipakai oleh manajemen studio foto.

Aplikasi manajemen studio foto digital mampu menangani pengolahan data pelanggan, pencarian gambar, transaksi pencetakan foto, serta pelaporan hasil transaksi selama periode tertentu. Sedangkan, untuk aplikasi pencetakan foto *online* mampu menangani registrasi pelanggan, pembuatan album foto, penyimpanan foto, pencarian foto, serta transaksi pemesanan pencetakan foto secara *online*.

Aplikasi manajemen ini digunakan oleh 2 jenis *user*, yaitu *Operator* dan *Administrator*. *Operator* dalam aplikasi ini bisa berstatus sebagai karyawan yang bekerja pada studio foto, sedangkan *Administrator* merupakan pemilik studio foto. *User Operator* mempunyai hak akses terhadap menu mengolah data ukuran foto, data

pelanggan, data jenis transaksi, pencarian foto, mencetak laporan pelanggan, dan mencetak laporan transaksi pencetakan foto. *Administrator* mempunyai hak akses untuk menambah *user Operator* yang diambil dari karyawannya beserta seluruh hak akses yang dimiliki *user Operator*. Pada Gambar 1 ditampilkan struktur aplikasi manajemen studio foto digital secara garis besar.



Gambar 1 Struktur Aplikasi

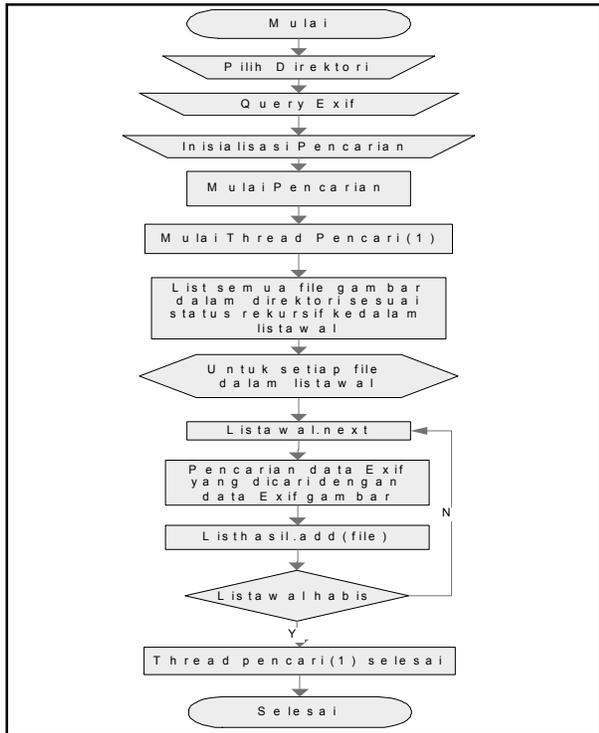
Seorang *user* harus melakukan proses *login* terlebih dahulu dengan memasukkan *username* dan *password* yang dimilikinya untuk menentukan *level user* yang dimiliki. *User Administrator* berhak mengakses seluruh menu yang menjadi milik *User Operator* dan tidak berlaku sebaliknya. Seluruh proses pengolahan data mengakses *Database Aplikasi*. Untuk mengakhiri penggunaan aplikasi, *user* diminta untuk melakukan proses *Logout* terlebih dahulu.

Proses pencarian gambar berdasarkan *metadata Exif* selanjutnya disebut *Exiffinder*. Proses pencarian dimulai dari adanya masukan jenis *query* oleh *user*. *User* bisa mengisikan salah satu atau beberapa *tag* dari data *Exif*, tidak perlu semuanya. Setelah itu, *user* memilih data yang dimaksudkan. Kemudian, dihasilkan sebuah *query* masukan yang sudah berisi nilai dari masing-masing *tag Exif* yang dipilih dan diisi datanya. Proses pencarian gambar dan menampilkan hasil pencarian dilakukan dengan konsep *multithreading*.

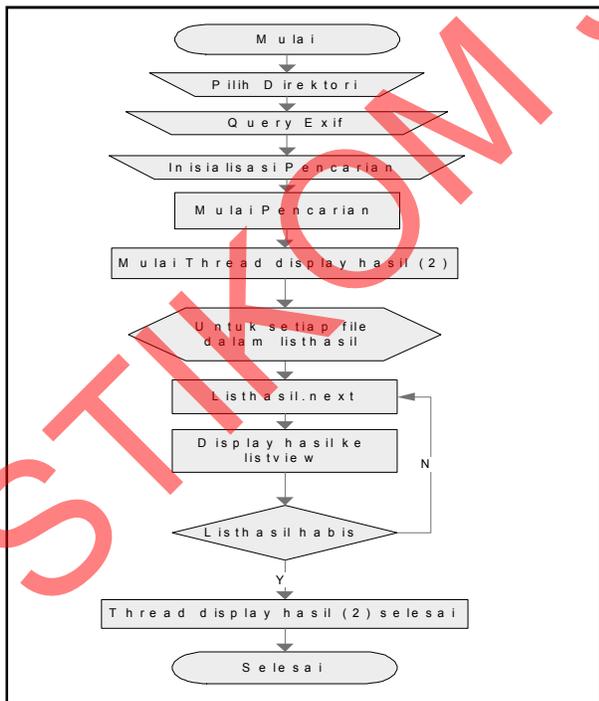
Adapun detail dari proses pencarian digambarkan pada *flowchart* di Gambar 2 dan 3. Gambar 2 merupakan proses pencarian pada *thread* pencari dan Gambar 3 pada *thread display* hasil. *Thread* pencari dan *display* hasil mengimplementasikan konsep *multithreading*.

Proses pencarian dimulai dengan melakukan *query Exif* yang dipilih oleh *user*. Setelah *user* memilih *metadata Exif* mana yang akan dicari, maka dilakukan pengesetan nilai dari *metadata Exif* yang diinginkan *user*. Lalu, untuk menjalankan proses pencarian dan menampilkan hasil dibuatlah 2 *thread* yang menangani 2 proses tersebut. *Thread* pertama menangani proses pencarian berdasarkan *metadata Exif* yang telah diset sebelumnya untuk dicocokkan dengan *metadata Exif* dari *file* foto dan *thread*

kedua menangani proses menampilkan hasil pencarian.



Gambar 2 Proses Pencarian pada Thread Pencari



Gambar 3 Proses Pencarian pada Thread Display Hasil

SIMPULAN

Kesimpulan yang dapat diambil dari pembuatan aplikasi Sistem Informasi Manajemen Studio Foto Digital adalah sebagai berikut:

1. Aplikasi *desktop* manajemen studio foto digital yang dibuat mampu menangani transaksi pencetakan foto di studio dan juga mampu melakukan pencarian pas foto digital berdasar *metadata Exif*.
2. Aplikasi *web* pemesanan cetak foto *online* menyediakan layanan bagi anggota untuk menyimpan foto pada album *virtual* di *website* dan juga memesan pencetakan foto *online*, sehingga tidak perlu datang ke studio foto untuk mencetak fotonya.
3. Berdasarkan hasil uji coba dapat disimpulkan bahwa aplikasi dapat menerapkan konsep *multithreading* dalam proses pencarian foto berdasar *metadata Exif*.

Adapun saran-saran yang dapat digunakan untuk mengembangkan aplikasi yang telah dibuat adalah sebagai berikut:

1. Pencarian foto bisa dikembangkan tidak hanya berdasar *metadata Exif* yang dimiliki oleh gambar atau foto tetapi juga memanfaatkan unsur *CBIR* seperti bentuk dan tekstur gambar.
2. Aplikasi *web* layanan pemesanan cetak foto *online* bisa dikembangkan dengan menggunakan sistem *SMS Broadcast*.
3. Aplikasi *desktop* untuk pencarian foto bisa memanfaatkan teknologi *web service* untuk fungsi pencariannya, sehingga bisa diakses oleh banyak aplikasi.

RUJUKAN

- Chandra, I. 2003. *Utility Kamera Digital*. Jakarta: PT. Elex Media Komputindo.
- Deitel, HM, Deitel PJ, and Nieto, TR. 2002. *Visual Basic .NET How to Program Second Edition*. New Jersey: Prentice-Hall International, Inc., Upper Saddle River.
- Kertajaya, H. 2003. *Exif dan PIM Mempermudah Pencetakan?*. Jakarta: Fotomedia PT Gramedia.
- Lowe, D. 2006. *ASP.NET 2.0 Everyday Apps for Dummies*, New Jersey: Wiley Publishing Inc.
- Microsoft Software Developer Network (MSDN). 2007. *ASP.NET AJAX Overview*. URL: <http://msdn.microsoft.com>, diakses pada 16 Januari 2007.