

PROTOTIPE VIDEO EDITOR DENGAN MENGGUNAKAN *DIRECT X* DAN *DIRECT SHOW*

Djoni Haryadi Setiabudi

Fakultas Teknologi Industri, Jurusan Teknik Informatika, Universitas Kristen Petra
e-mail : djonih@petra.ac.id

Ibnu Gunawan

Fakultas Teknologi Industri, Jurusan Teknik Informatika, Universitas Kristen Petra
e-mail : ibnu@petra.ac.id

Teguh Mujianto

Alumnus Fakultas Teknologi Industri Jurusan Teknik Informatika, Universitas Kristen Petra

ABSTRAK: Perkembangan teknologi memberi kesempatan masyarakat untuk mengabadikan saat – saat yang penting menggunakan video. Pembentukan video digital yang baik membutuhkan proses editing yang baik pula. Untuk melakukan proses editing video digital dibutuhkan program editor. Berdasarkan permasalahan diatas maka pada penelitian ini dibuat prototipe editor sederhana untuk video digital.

Pembuatan aplikasi memakai teknik pemrograman di bidang multimedia, khususnya *video*. Perencanaan dalam pembuatan aplikasi tersebut dimulai dengan pembentukan kompresi dan dekompresi *file video*, kemudian dilanjutkan dengan proses *editing* terhadap *file video digital*. Selain itu program juga dilengkapi dengan fasilitas untuk mendukung proses *editing*. Aplikasi tersebut dibuat dengan Microsoft Visual C++ dengan menggunakan teknologi DirectX, khususnya DirectShow. Aplikasi tersebut menyediakan fasilitas – fasilitas dasar yang membantu proses *editing* pada *video digital*. Dengan aplikasi tersebut dihasilkan *file video* baru dalam format AVI yang terbentuk dari proses *editing* yang dilakukan.

Hasil pengujian menunjukkan kemampuan aplikasi melakukan proses pemotongan dan penyisipan file video dalam format AVI, MPEG, MPG dan DAT. Proses pemotongan dan penyisipan dilakukan dengan urutan yang statis. Selain itu aplikasi juga menyediakan efek antara masing – masing clip. Dan yang terakhir melakukan proses penyimpanan satu file dalam format AVI yang terbentuk dari proses editing dalam aplikasi.

Kata kunci: Multimedia, Video Editor, Direct-X, Direct-Show, AVI, MPG.

ABSTRACT: *Technology development had given people the chance to capture their memorable moments in video format. A high quality digital video is a result of a good editing process. Which in turn, arise the new need of an editor application. In accordance to the problem, here the process of making a simple application for video editing needs.*

The application development use the programming techniques often applied in multimedia applications, especially video. First part of the application will begin with the video file compression and decompression, then we'll step into the editing part of the digital video file. Furthermore, the application also equipped with the facilities needed for the editing processes. The application made with Microsoft Visual C++ with DirectX technology, particularly DirectShow. The application provides basic facilities that will help the editing process of a digital video file. The application will produce an AVI format file after the editing process is finished.

Through the testing process of this application shows the ability of this application to do the 'cut' and 'insert' of video files in AVI, MPEG, MPG and DAT formats. The 'cut' and 'insert' process only can be done in static order. Further, the application also provide the effects facility for transition process in each clip. Lastly, the process of saving the new edited video file in AVI format from the application.

Keywords: Multimedia, Video Editor, Direct-X, Direct-Show, AVI, MPG.

1. PENDAHULUAN

Dalam pembentukan suatu video diperlukan proses editing yang membantu pengguna untuk memilih bagian-bagian terbaik dan benar-benar dibutuhkan dalam video tersebut. Proses editing biasanya dilakukan dengan memotong - motong bagian yang tidak dibutuhkan. Proses editing ini tidak dapat dilakukan secara manual melainkan dengan software pendukung yang umumnya disebut Video Editor.

Program video editor sangat dibutuhkan oleh pengolahan video, karena dalam satu rekaman video terdapat beberapa bagian yang tidak memiliki kualitas yang baik atau sesuai dengan tujuan video itu dibuat, bagian - bagian tersebut yang biasanya dihilangkan agar tidak mengurangi kualitas video secara keseluruhan. Saat ini banyak program video editor yang beredar dipasaran, dengan berbagai macam interface dan fitur yang disediakan untuk membantu pengguna program tersebut, mulai dari yang sederhana sampai yang menggunakan proses - proses yang rumit dan sulit dimengerti oleh pengguna yang masih belum berpengalaman dalam bidang editing.

Selain program video editor dibutuhkan pula program encoder. Program encoder adalah suatu program yang bertujuan untuk mengubah format file video menjadi format yang diinginkan seperti AVI, MPEG1, MPEG2, MPEG4, SVCD dan masih banyak yang lainnya. Pada umumnya pengolahan video dilakukan secara langsung dalam format AVI atau dalam format MPEG dengan melakukan proses encoder terlebih dahulu, kemudian setelah proses pengeditan video tersebut selesai dilakukan barulah video akan disiapkan untuk disimpan ke dalam format file yang dipilih sesuai dengan kebutuhan penggunaannya.

Dan saat ini umumnya kedua program tersebut masih dibuat dalam dua program terpisah, sehingga dalam proses pengolahan video pengguna diharuskan untuk menggunakan dua

program terpisah untuk mendapatkan hasil yang diinginkan. Oleh karena itu dibutuhkan satu perangkat lunak yang mendukung kedua fasilitas dari kedua program diatas untuk memudahkan proses pengeditan suatu file video. Selain itu program ini secara tidak langsung juga harus dapat berfungsi sebagai *player* yang digunakan untuk melihat hasil editing user.

Dengan latar belakang diatas, pada penelitian ini dicoba dibuat satu prototipe suatu program video editor sederhana yang dilengkapi dengan proses editing video, encoder dan player.

2. DIRECTSHOW PROGRAMMING [5]

Microsoft DirectShow digunakan untuk melakukan pemrograman yang mendukung file audio dan video. DirectShow mendukung berbagai macam format termasuk file Advanced Streaming Format (ASF), Motion Picture Experts Group (MPEG), Audio-Video Interleaved (AVI), MPEG Audio Layer-3 (MP3), and WAV. Dalam penggunaannya DirectShow juga mendeteksi dan menggunakan akselerasi video dan audio pada perangkat keras komputer tersebut jika tersedia, bila tidak terdapat kemampuan akselerasi oleh perangkat keras DirectShow juga mendukung sistem tanpa akselerasi perangkat keras.

DirectShow berbasis pada Component Object Model (COM). DirectShow menyediakan tiga fitur utama yaitu: Media-stream playback, Capture, Multimedia streaming.

Microsoft DirectShow Editing Service

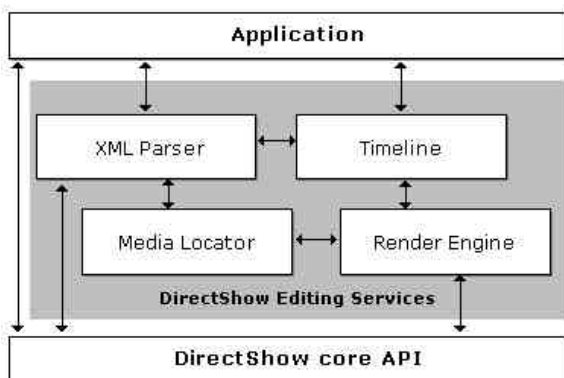
Microsoft DirectShow Editing Service (DES) adalah pemrograman dalam directX yang mengatur proses khusus untuk video editing. Microsoft DES dibangun diatas inti arsitektur dasar DirectShow yang menyediakan sekumpulan *interface* yang didesain untuk pemrograman video editing. Ada beberapa *feature* yang diberikan Microsoft DES melalui DirectShow yaitu :

- Suatu model garis waktu yang mengorganisir track video dan audio ke dalam layer yang berhubungan, hal itu akan memudahkan untuk mendapatkan hasil akhir.
- Kemampuan untuk menampilkan hasil dari suatu proyek video.
- Mendukung efek untuk video dan audio, juga transition antara dua track video seperti fade dan wipe.
- Konversi frame rate dan audio sampling rate secara otomatis.
- Mengubah ukuran dan memotong video.

Dalam pengoperasiannya Microsoft DES terdiri beberapa bagian yang mendukung cara kerja Microsoft DES dalam DirectShow. Bagian-bagian tersebut adalah :

- Timeline
- XML Parser
- Render Engine
- Media Locator

Semua bagian tersebut bekerja seperti pada gambar 1.



Sumber : MSDN Microsoft DirectShow

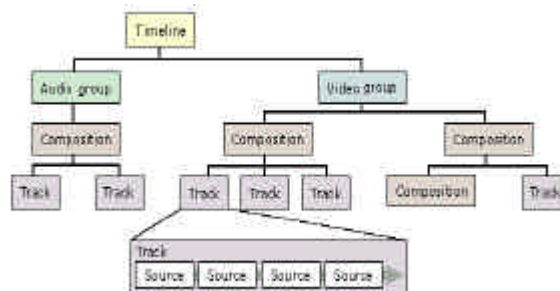
Gambar 1. Arsitektur Microsoft DES

Dalam proses video editing Microsoft DES hanya dibutuhkan dua bagian dari keseluruhan Microsoft DES yaitu timeline dan render engine.

Timeline merupakan dasar dari proyek video editing yang semua obyek yang digunakan dalam proyek tersebut. Timeline menggambarkan pembuatan sebuah video sebagai kumpulan dari source clip, transition, dan efek yang diatur dalam satu track yang berurutan.

Render engine mengubah timeline menjadi suatu bentuk yang dapat dirender sebagai media bergerak. Umumnya render engine membentuk sebuah filter graph.

Sebuah pembuatan video editing diawali dengan bahan baku mentah yang dapat berupa video clips, file – file suara, dan file gambar statis. Kemudian akan dilakukan proses pengambilan bagian – bagian yang dibutuhkan dari masing – masing file bahan baku tersebut, dan diatur dalam urutan tertentu. Dalam video editing cara termudah adalah dengan menggabungkan potongan – potongan video berurutan secara langsung. Setiap potongan diikuti oleh potongan yang lainnya. Pengaturan seperti ini menggambarkan sistem kerja dari track dimana dalam satu proyek video editing sederhana minimal terdapat satu video track yang selanjutnya dapat berkembang menjadi multiple track. Misalnya ada beberapa audio yang dicampur menjadi satu. Sekumpulan source satu tipe media (audio atau video) yang sejenis adalah sebuah track. Track juga dapat menggunakan transition – transition baik untuk audio dan video seperti fade dan wipe untuk berpindah dari satu track ke track yang lain. Kumpulan dari track akan membentuk compositing. DES menunjukkan kerja komponen – komponen diatas menggunakan sebuah bagan pada gambar 2.



Sumber : MSDN Microsoft DirectShow

Gambar 2. Struktur Timeline

Timeline minimal harus berisi paling sedikit satu group. Pada umumnya suatu pembentukkan video menggunakan satu video group dan satu audio group. Sedangkan keberadaan composition

adalah pilihan. Dari bagan diatas dapat dipahami beberapa hal yaitu:

Bagian dari video maupun audio yang diambil disebut dengan source

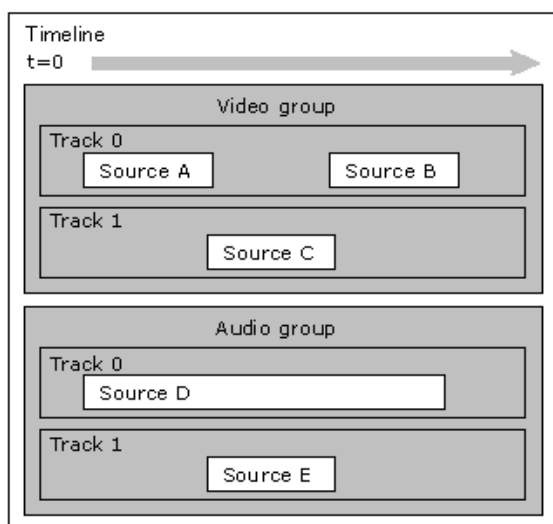
Sekumpulan source dengan tipe media yang sama (video atau audio) disebut track

Sejumlah track akan membentuk satu composition yang dibuat sebagai gabungan dari semua track yang terkandung didalamnya. Satu composition dapat mengandung composition yang lain yang akan membentuk susunan track yang lebih kompleks.

Tingkat tertinggi dari kumpulan composition dan track adalah group.

Satu atau sejumlah group akan membentuk satu timeline yang menjadi akar dalam bagan diatas.

Proses render dalam dalam timeline sangat dibutuhkan untuk membentuk gabungan dari semua track dan composition yang terkandung didalamnya. Untuk memahami proses rendering pada timeline, bayangkan bahwa timeline adalah sebuah urutan waktu. Proses rendering akan dilakukan pada track yang terdapat dalam timeline sesuai dengan urutan waktu tertentu.

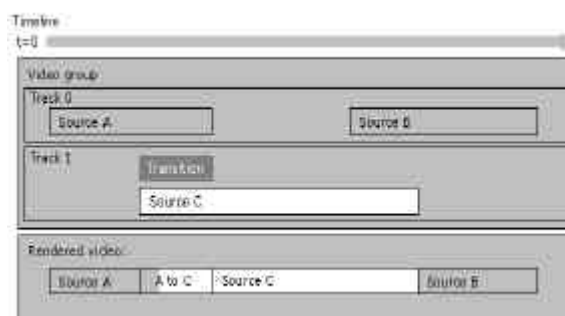


Gambar 3. Proses Rendering Suatu File

Pada gambar 3 ditunjukkan video group track 1 memiliki prioritas yang lebih tinggi dibandingkan track 0. Ketika pada track 1 terdapat source maka proses render akan mengarahkan semua yang

ada di track 0 dengan source pada track 1 dan sebaliknya ketika track 1 kosong maka proses render akan mengambil source pada track 0. Jadi proses render diatas akan dimulai dengan source A pada track 0 yang kemudian dipotong oleh source C pada track 1, dan terjadi penumpukan pada track 0 oleh source C dimulai dari akhir source A dan awal source B. Setelah source C selesai maka di lanjutkan oleh source B pada track 0. Tetapi pada Audio group terjadi proses render yang berbeda semua track di campur bersama-sama. Audio dimulai dengan source D kemudian beralih ke campuran antara source D dan source E dan diselesaikan oleh source D kembali.

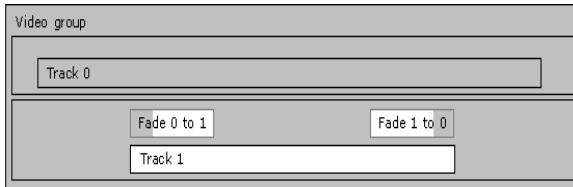
Seperti dijelaskan diatas bahwa antara track dapat juga diisi dengan transition-transition baik yang digunakan oleh video maupun audio. Transition adalah suatu jalan untuk berpindah dari satu video track ke track lainnya, menggunakan efek visual seperti fade atau wipe. Dengan adanya transition pada perpindahan dari satu track ke yang lainnya akan membuat video menjadi lebih baik, karena perpindahan dari satu potongan video ke potongan yang lain akan dibuat tidak jelas dengan adanya transition. Lihat Gambar 4.



Sumber : MSDN Microsoft DirectShow

Gambar 4. Proses Rendering File Dengan Transition

Pada transition juga terdapat arah tertentu untuk transition tersebut yang akan mengatur dari arah mana transition itu akan dilakukan. Seperti ditunjukkan pada gambar 5.



Gambar 5. Pengaturan Arah Transition

Dalam sejumlah track, track pertama mempunyai prioritas yang terendah, dan setiap track berikutnya mempunyai satu tingkat yang prioritas lebih tinggi. Track dapat disamakan sebagai sebuah layer, dimana jika pada proses rendering ditemukan dua layer pada posisi yang sama, maka layer dengan prioritas tertinggi akan ditampilkan dan layer yang lain akan disembunyikan. Untuk melakukan proses editing pada video anda harus berpegangan pada beberapa konsep waktu, seperti :

Setiap klip memiliki durasi waktu.

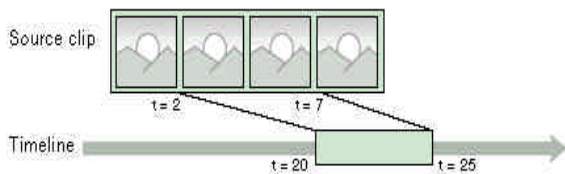
Klip, transition, dan efek tampil pada waktu dalam suatu proyek.

Video mempunyai frame rate dalam satuan frame per detik (fps).

DES memiliki beberapa metode untuk berkoordinasi dengan waktu atau frame rate. Ada beberapa ukuran waktu yang digunakan dalam DES yaitu Time Values dan *Frame Rate*. Pada *time values* terdapat beberapa parameter yang menunjukkan waktu, tiga parameter tersebut adalah :

Timeline Time, yaitu waktu yang berhubungan dengan permulaan waktu dalam timeline jadi timeline time juga dapat menjadi "project time".

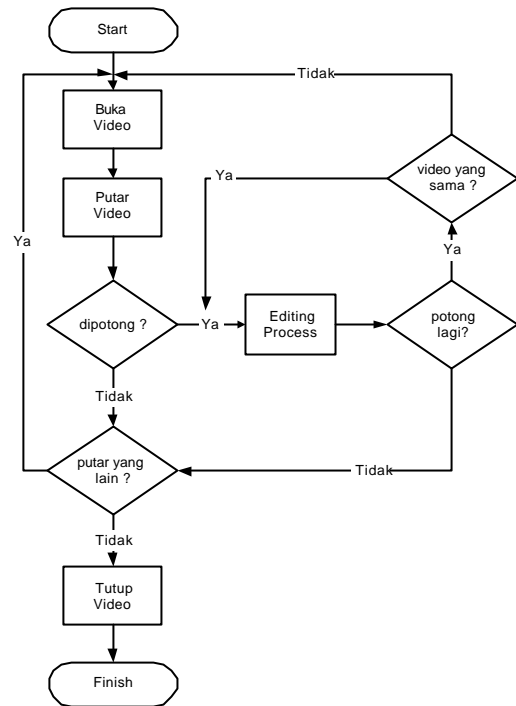
Media Time, adalah satuan waktu yang digunakan pada file source.



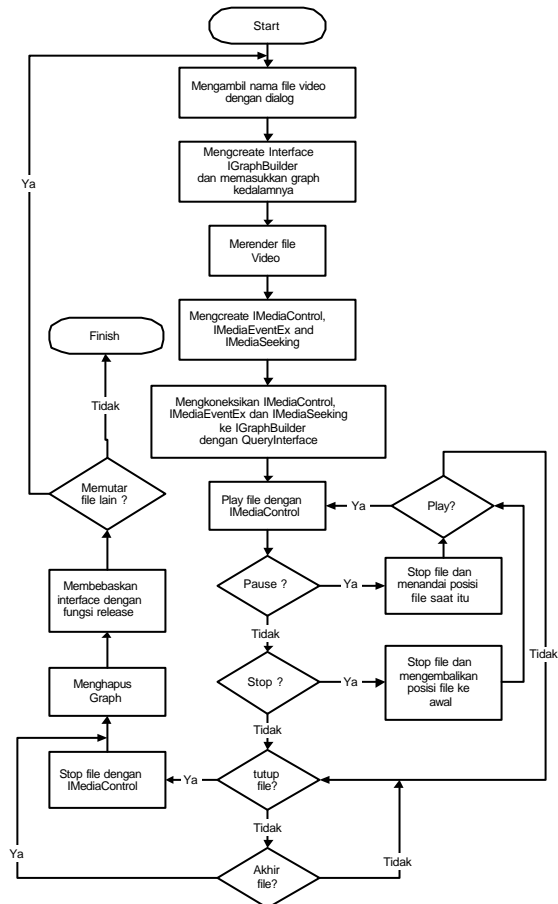
Gambar 6. Perbedaan Timeline Time dan Media Time

Parent Time, adalah satuan waktu yang berhubungan dengan sebuah obyek dalam timeline. Lihat gambar 6

3. DESAIN PROTOTYPE



Gambar 7. Flowchart General atau Keseluruhan



Gambar 8. Flowchart Proses Video Player

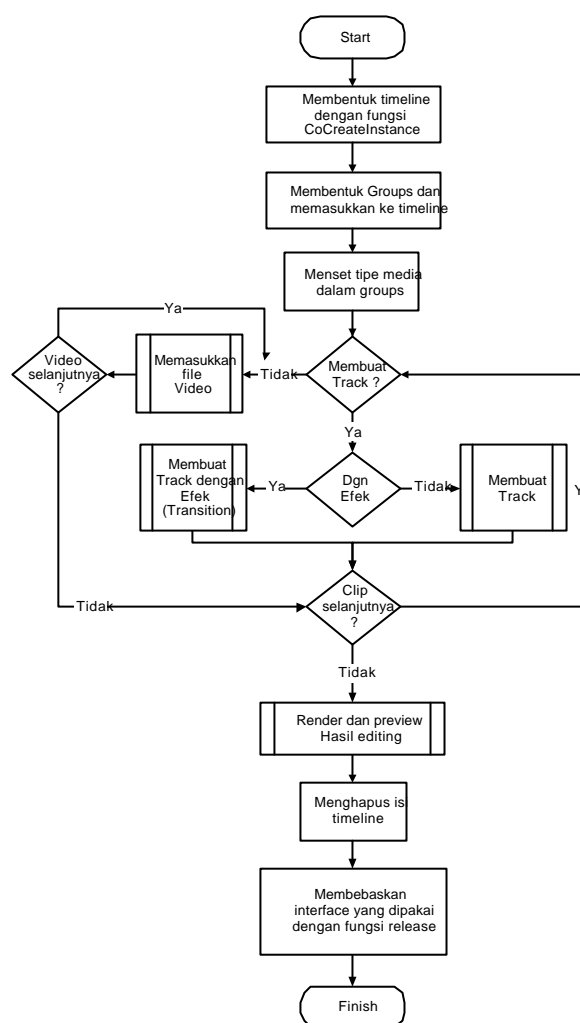
Pada flowchart gambar 7 digambarkan cara kerja program secara umum, dalam program ini user diberi kesempatan untuk memanfaatkan program *player* tanpa terpengaruh dengan program *editing*. Pada flowchart gambar 8 dijelaskan cara kerja program *player* secara detail tanpa menggunakan program *editing* sama sekali. Pada Gambar 7 terdapat flowchart yang dimulai dengan membuka *file video* dalam berbagai macam format seperti AVI, MPEG, dan lainnya. Kemudian *file video* itu akan diputar sampai user memutuskan apakah *file* tersebut akan mengalami proses *editing* atau tidak. Jika tidak mengalami proses *editing* maka akan ditanyakan apakah user akan memutar *file video* yang lainnya. Jika mengalami proses *editing* maka akan memasuki *editing process* yang dijelaskan pada Gambar 9.

Pada gambar 9 (*editing process*) terdapat beberapa tahap yang harus dilalui oleh user untuk mendapatkan potongan *video* yang pada akhirnya akan mengalami proses penyatuan. Tahap yang pertama adalah penentuan awal dan akhir dari potongan *video* yang akan digunakan. Penentuan awal dan akhir dari potongan *video* ini dilakukan bersama-sama dengan memutar *file video* tersebut pada *player*.

Kemudian pada tahap selanjutnya potongan *video* tersebut disimpan sementara untuk menunggu potongan-potongan *video* yang berikutnya. Selain itu user juga dapat menyisipkan efek-efek yang dibutuhkan user dalam perpindahan dari satu potongan *video* ke potongan yang lainnya. Efek-efek ini dianggap sebagai satu obyek *independent* yang berada diantara dua potongan *video*.

Setelah itu user dapat mengurutkan potongan-potongan *video* dan efek yang telah disimpan diatas untuk membentuk *video* yang diinginkan, kemudian user akan ditanya apakah akan melakukan proses pemotongan kembali pada *file video* yang sama. Apabila pada *file video* yang sama maka user akan menjalankan kembali tahap-tahap diatas. Jika tidak pada *file video* yang sama maka user akan

membuka *file video* yang lain dan masuk ke Gambar 7 yang kemudian kembali ke proses *editing* dan menjalani tahap-tahap pemotongan seperti sebelumnya.



Gambar 9 Flowchart Editing Process

Setelah user selesai melakukan pemotongan-pemotongan *file video* yang dibutuhkan user menyelesaikan proses *editing* ini dengan melakukan proses penggabungan potongan-potongan *video* dan efek-efek yang digunakan untuk membentuk satu *file video* baru yang merupakan hasil dari proses *editing* yang dilakukan.

4. IMPLEMENTASI

Prototipe dibuat dengan menggunakan Microsoft Visual C++ 6.00, harapan dapat diperoleh kestabilan dalam program video editing ini. Selain itu dengan

menggunakan visual C++ dapat dibentuk program yang memiliki interoperability yang tinggi sehingga memudahkan pemakai dalam menggunakan di platform mana saja selama berbasis windows. Dalam prototipe ini terdapat dua program dasar yang disatukan untuk menunjang program yang lainnya, yaitu : Video Player dan Video Editor.

Program video player berfungsi untuk memutar file video dalam berbagai macam format seperti AVI, MPEG, DAT dan yang lainnya. Atau file – file audio seperti MP3 dan MIDI. Dengan program ini dapat dilihat bagian mana dari suatu file video yang akan mengalami proses editing, selain itu program ini juga berfungsi untuk melihat hasil proses editing yang dilakukan oleh user baik dari satu file multimedia atau dari beberapa file multimedia.

Program video editing adalah program yang digunakan untuk mengedit satu file multimedia. Proses editing tersebut meliputi berbagai macam proses lainnya seperti memotong file video, menggabungkan potongan-potongan file tersebut, memberikan efek terhadap penggabungan dari dua potongan file. Dengan program ini diharapkan user dapat memperbaiki file multimedia yang membutuhkan sentuhan akhir sebelum siap digunakan.

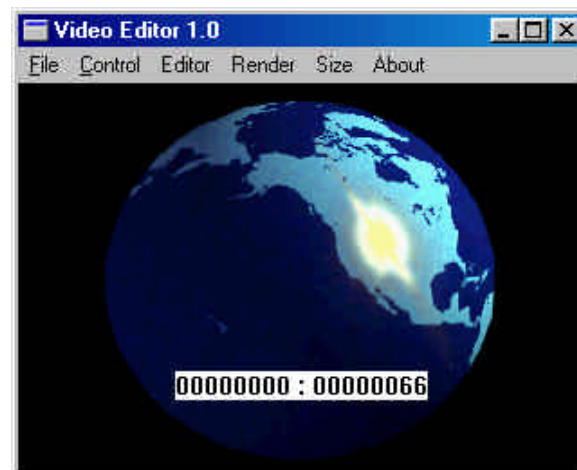
Kedua program diatas saling berkaitan satu dan yang lainnya sehingga dalam penggunaannya sebaiknya digunakan bersama-sama untuk mendapatkan kemudahan dan hasil yang maksimal dalam pembentukan suatu video.

5. PENGUJIAN

5.1 Program Video Player

Video player adalah video yang digunakan untuk memutar file multimedia baik video maupun audio, pada video player dapat digunakan beberapa menu yaitu : File, Control, Size, seperti pada Gambar 10. File video yang diinginkan akan terbuka pada layar hitam yang berada pada program utama. Ukuran

layar akan berubah sesuai dengan ukuran panjang dan lebar dari video yang sesungguhnya (100%). Dan file video tersebut dapat langsung diputar atau dijalankan.



Gambar 10 Tampilan program Video Player

Ketika *file video* dijalankan program dapat langsung mengontrol jalannya *file video* tersebut dengan menu *Control* untuk menghentikan dan menjalankan kembali *file* tersebut. Selain itu program juga dapat dirubah ukuran layarnya dengan menggunakan menu *Size*.

5.2 Program Video Editor

Dalam penggunaan program *video editor* diperlukan proses – proses yang berasal dari program *video player*. Proses *playing* diperlukan oleh program *video editor* untuk menentukan bagian mana dari satu *file video* yang diinginkan untuk dipotong dan dimasukkan ke dalam *file video* yang baru.

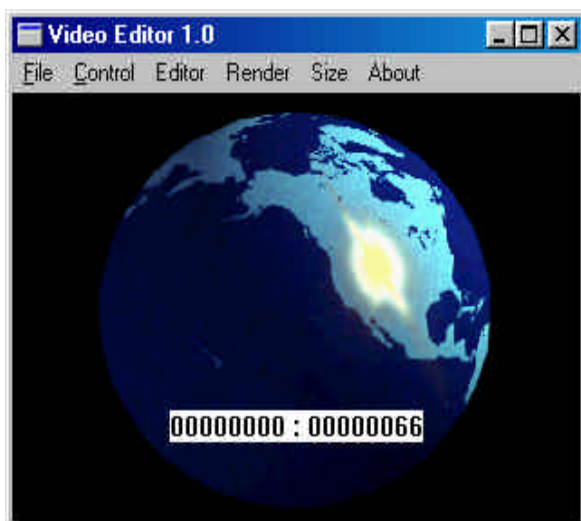
Untuk memulai menggunakan program *video editor* dapat dimulai dengan memilih sub menu *New File* yang berada pada menu *File*. Setelah memilih menu tersebut program dapat langsung melakukan pemotongan pada *file-file video* yang lain dengan menggunakan menu *Editor*. Pada program *video editor* ini menunjang tiga macam proses yaitu: *Insert Track*, *Insert Video*, *Insert Track with Wipe Effect*.

Untuk melakukan ketiga macam proses tersebut dapat dilakukan seperti penjelasan dibawah ini.

5.2.1 Memasukkan potongan Video (Insert track)

Untuk memasukkan *track* atau potongan *file video* dapat dilakukan dengan memilih sub menu *Insert Track*, setelah itu akan muncul *dialog box* kembali seperti pada sub menu *Open File* untuk menentukan *file* mana yang akan dipotong oleh program. Kemudian untuk membukanya dapat dilakukan dengan menekan tombol *Open*.

Setelah itu *file video* tersebut akan terbuka seperti saat kita menggunakan program *video player*. Hanya saja *file* tersebut tidak langsung dijalankan seperti pada program *video player* tetapi dalam posisi berhenti (*pause*). Untuk memberi kesempatan pada *user* menentukan bagian awal potongan yang sama dengan bagian awal *file*.



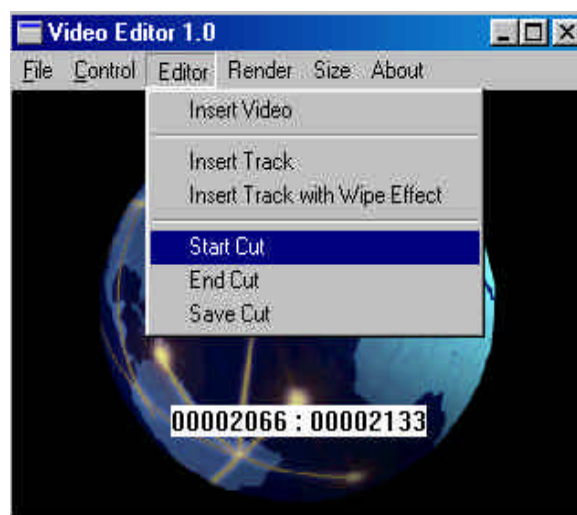
Gambar 11. File video yang terbuka dalam keadaan Pause

Kemudian kita dapat menjalankan *file* tersebut dengan menggunakan sub menu *play* atau *pause* pada menu *Control* sampai pada saat dimana kita akan memotong *file* tersebut. Seperti terlihat pada gambar 11.

Saat kita ingin memulai pemotongan *file video* tersebut dapat dilakukan dengan menghentikan *file video* tersebut

melalui sub menu *pause* atau dapat juga dilakukan dengan menekan tombol 'P' pada *keyboard*. Dan *file video* tersebut akan dihentikan oleh program.

Kemudian untuk menandai bagian awal dari potongan yang diinginkan pada *file* tersebut dapat dilakukan dengan menekan sub menu *Start Cut* pada menu *Editor*. Seperti terlihat pada gambar 12.



Gambar 12. Menentukan bagian awal pemotongan file

Untuk menentukan awal pemotongan suatu *file* harus dilakukan dalam keadaan *file video* berhenti, jadi sebelum memilih sub menu *Start Cut* seperti pada gambar 12, program harus menghentikan *file video* yang sedang dijalankan dengan cara diatas.

Setelah menandai bagian awal dari pemotongan yang dikehendaki, kemudian dapat dilanjutkan dengan kembali menjalankan *file video* dengan sub menu *pause* pada menu *Control* atau dengan menekan tombol 'P' pada *keyboard*. Setelah itu *file video* tersebut akan dijalankan kembali. Kemudian jika telah sampai pada bagian akhir dari potongan *video* tersebut dapat ditandai kembali dengan sub menu *End Cut* seperti pada gambar 12 diatas. Namun sebelum memilih sub menu *End Cut*, seperti pada menu *Start Cut* *file video* harus kembali dihentikan, yang kemudian diikuti dengan pemilihan sub menu *End Cut*.

Setelah bagian akhir dari pemotongan ditentukan jika bagian yang akan

dipotong tersebut telah benar dan sesuai dengan kehendak user dapat disimpan dengan memilih sub menu *Save Cut* pada menu *Editor*. Namun jika pemotongan tersebut mengalami kesalahan *user* dapat mengulangi penentuan bagian awal dan akhir dari potongan *video* tersebut dengan memutar ulang *file video* tersebut, dan mengulangi langkah-langkah diatas kembali. Untuk memutar ulang *file video* dapat dilakukan dengan memilih sub menu *stop* pada menu *Control* dan dilanjutkan dengan memilih sub menu *play* pada menu *Control* juga.

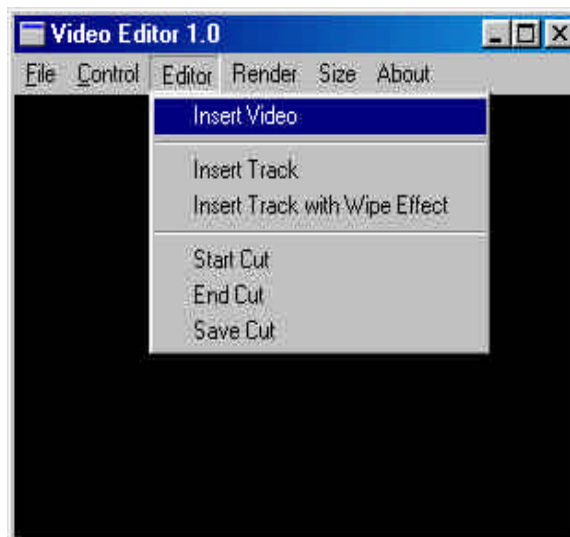
Setelah proses penentuan awal dan akhir potongan *file video* selesai dapat diakhiri dengan menyimpan potongan tersebut dengan cara memilih sub menu *Save Cut*. Kemudian jika ada bagian lain dalam *file video* yang sama yang ingin dipotong kembali *user* dapat menjalankan *file video* tersebut sampai pada bagian tersebut dimana *user* ingin melakukan pemotongan dan dapat dilakukan penandaan awal dan akhir pemotongan dengan mengulangi langkah-langkah diatas dan kemudian diakhiri dengan menyimpan potongan tersebut ke dalam program. Untuk memasukkan potongan lain pada *file video* yang sama tidak perlu memilih sub menu *Insert Track* pada menu *Editor* lagi, namun jika potongan yang diinginkan berada pada *file video* yang lain atau berbeda, diperlukan untuk memilih sub menu *Insert Track* lagi untuk menentukan file baru yang akan dilakukan proses pemotongan.

5.2.2 Memasukkan *file video* (*Insert video*)

Setelah proses pemotongan *file video* dengan menggunakan sub menu *Insert Track* pada menu *Editor*, program dapat memasukkan satu *file video* secara utuh ke dalam program dengan menggunakan sub menu *Insert Video*.

Dengan memilih sub menu *Insert Video* user tidak perlu lagi menentukan bagian awal dan akhir pemotongan yang dilakukan dengan sub menu-sub menu *Start Cut*, *End Cut* dan *Save Cut*, karena proses penentuan bagian awal dan akhir

dari *file* tersebut dilakukan oleh program yang langsung diatur secara otomatis sesuai bagian awal dan akhir dari *file video* tersebut. Setelah *file* terbuka, *file* akan disimpan ke dalam program sebagai bagian dari *file editing* yang sedang dibuat. Lihat Gambar 13



Gambar 13. Sub menu *Insert Video*

5.2.3 Memasukkan Efek (*Transition*)

Transition adalah penambahan efek di antara satu *track* dan *track* yang lainnya. Proses ini dapat dilakukan dengan menggunakan sub menu *Insert Track with Wipe Effect* pada menu *Editor*. Efek disini memiliki durasi yang tetap yaitu empat detik. Untuk memasukkan efek diantara kedua *track* harus dilakukan pada saat memasukkan *track* yang kedua. Misal antara *track A* dan *track B* akan dimasukkan efek *wipe*. Maka urutan yang harus dilakukan adalah memasukkan *track A* dengan sub menu *Insert Track*, dan kemudian diikuti dengan memasukkan *track B* dengan menggunakan sub menu *Insert Track with Wipe Effect*. Untuk melihat hasil yang telah dibuat dapat dilakukan dengan menggunakan sub menu *Pre-viewing* dalam menu *Render*. Hasil render dapat dilihat pada Gambar 14.



Gambar 14. Hasil render dengan efek wipe

6. KESIMPULAN

Berdasarkan pengujian yang dilakukan terhadap aplikasi *video player* yang terbentuk, maka dapat ditarik kesimpulan sebagai berikut :

1. Aplikasi *video player* dapat memutar *file video* dalam *format* tertentu yaitu AVI, MPG, MPEG dan DAT.
2. Aplikasi dapat melakukan pemotongan pada *file-file video* dalam *format* tertentu yaitu AVI, MPG, MPEG, dan DAT
3. Aplikasi *video editor* hanya dapat memasukkan potongan-potongan *video* dengan urutan yang tetap. Setelah satu potongan *video* di simpan atau dimasukkan ke dalam program maka urutannya tidak dapat dirubah lagi.
4. Kemampuan aplikasi *video editor* untuk memasukkan *file video* secara utuh dapat pula dijadikan sebagai alat untuk menggabungkan beberapa *file video*.
5. Aplikasi *video editor* dapat langsung melakukan proses *saving* tanpa melakukan proses *preview* terlebih dahulu.
6. Aplikasi *video editor* hanya dapat membentuk satu *format file output* yaitu dalam *format* AVI.
7. Dalam pemrograman menggunakan DirectX, khususnya DirectShow menggunakan *filter-filter* yang saling dihubungkan satu dan yang lainnya oleh program untuk melakukan proses

tertentu. *Filter-filter* tersebut dapat ditemukan pada *interface-interface* COM yang digunakan dalam program, dalam menghubungkan *interface-interface* tersebut dibutuhkan perhatian ekstra terhadap tiap *interface*, karena masing-masing *interface* yang ada saling tergantung. Karena ketergantungan setiap *interface* ketika selesai digunakan dan dalam proses *release* masing-masing *interface* harus dilakukan dengan hati-hati supaya tidak mempengaruhi *interface* yang lain. Proses *release* pada masing-masing *interface* tidak dapat dihilangkan karena apabila kita tidak menggunakan proses tersebut maka penggunaan memori tidak dapat dikembalikan dan digunakan kembali oleh program.

DAFTAR PUSTAKA

1. Changgui Shi. "A Fast MPEG Video Encryption Algorithm" (online), http://www.acm.org/sigs/sigmm/MM98/electronic_proceedings/shi/ (2002, Sep.10). 1998.
2. Charles, P., *Programming Windows*. Washington: Microsoft Press. 1998.
3. David, J.K., Scot, W., & George, S., *Programming Visual C++*, Washington: Microsoft Press. 1998.
4. Microsoft. "MSDN Microsoft Visual C++" (online), <http://msdn.microsoft.com/visualc/> (Jan,25), 2003.
5. Microsoft. "MSDN Microsoft DirectShow" (online) <http://www.microsoft.com/Developer/PRODINFO/directx/dxm/help/ds/default.htm> (2003, Jan. 25), 1999.
6. OpenDML Subcommittee. *OpenDML AVI File Format Extensions*, Matrox Electronic System Ltd. 1997.
7. SFU Computer Science Course. "Video Compression" (online) <http://www.cs.sfu.ca/undergrad/CourseMaterials/C MPT479/material/notes/Chap4/Chap>

4.2/Chap4.2.html (2002, September 20), 1996.

8. Woobin Lee. "MPEG Compression Algorithm" (online), <http://icsl.ee.washington.edu/~woobin/papers/General/node2.html> (2002, Nov. 8), 1995.