Implementasi Plugin Impor Kamera dan Plugin Play Blast Untuk Autodesk Maya Berbasis Python

Novia Nurrohmah[1], Wawan Gunawan[2]
Divisi Produksi, Artis Layout 3D, PT Studio SHOH Entertainment[1]
Jurusan Teknik Informatika, Fakultas Ilmu Komputer, Universitas Mercu Buana [2]
Plaza 89, Jl. H. R. Rasuna Said No.6, Kuningan, Karet Kuningan, Setiabudi, Jakarta, 12940[1]
Jl. Raya Meruya Selatan, Kembangan, Jakarta, 11650[2]
E-mail : novia.nur@studioshoh.com[1], wawan.gunawan@mercubuana.ac.id[2],

Abstract – A 3D animation is creation through processes that can be done from computer work. These processes involving many team and takes a lot of time. To reduce time, plugin can be a solution. Plugin that can automate previsualization artist works with importing camera plugin and play blast plugin. As for importing camera plugin, instead of looking manually where the camera file is store, it is more convenient to import camera with one click. Also for the play blast plugin. Usually, camera file is imported from any folder that artist need to locate it manually and it takes a lot of time. And play blast video file as default, it stored in (C:\) Maya Document folder, that can be a problem later when (C:\) directory is full because play blast file. With plugin, it can save artist time to look for camera file, also it can prevent (C:\) directory from running out of space because of play blast video file.
Key words : Animation, three dimensions, plugin, previsualization

Abstrak* – Sebuah animasi tiga dimensi, merupakan sebuah karya yang dihasilkan melalui proses yang dikerjakan menggunakan komputer. Proses ini melibatkan banyak tim dan menghabiskan banyak waktu. Untuk memaksimalkan waktu, *plugin* dapat menjadi solusi. *Plugin* yang dapat mengautomatisasi pekerjaan layout artis, yaitu *plugin* impor kamera dan *plugin play blast.* Untuk *plugin* impor kamera misalnya, di bandingkan dengan mencari secara manual lokasi dimana file kamera disimpan, akan lebih mudah untuk melakukan impor kamera dengan satu kali klik. Begitu pula dengan *play blast.* Biasanya, file kamera diimpor dari folder mana saja dimana artis harus mencari file secara manual sehingga menghabiskan banyak waktu. Serta file video *play blast* secara default akan disimpan di (C:\) folder Maya Document, yang nantinya dapat menyebabkan masalah ketika direktori (C:\) penuh karena file *play blast*. Dengan menggunakan *plugin*, dapat menghemat waktu artis untuk mencari file kamera, serta dapat mencegah direktori (C:\) kehabisan ruang karena file video *play blast.
***Kata kunci* : Animasi, tiga dimensi, *plugin*, previsualisasi.**

I. PENDAHULUAN

Film atau seri animasi dihasilkan dari proses yang hampir kesemuanya di selesaikan dengan menggunakan komputer. Keseluruhan proses ini melibatkan banyak divisi dan memerlukan banyak waktu. Sedikit saja kesalahan pada satu proses, dapat berpengaruh pada keseluruhan proses. Oleh karena itu, untuk mengurangi kesalahan yang terjadi *plugin* merupakan jawaban. Pada kasus ini adalah *plugin* Autodesk Maya untuk divisi previsualisasi (layout).

Divisi previsualisasi merupakan divisi yang melakukan proses impor kamera. Untuk melakukan impor kamera ke dalam ruang kerja Maya, artis previsualisasi dapat menyimpan file kamera di folder manapun, ini tidak menjadi masalah karena obyek yang dihasilkan dari proses impor akan menjadi obyek independen di dalam ruang kerja Maya. Namun, akan berbeda jika proyek yang dikerjakan membutuhkan kamera hasil referensi, yang mana folder penyimpanan kamera menjadi hal penting. Karena, obyek hasil dari referensi akan menjadi obyek yang tersambung dengan file induknya, dimana setiap kali file kerja artis dibuka, obyek referensi ini akan mencari dimana file induknya berada. Apabila file Maya kamera (induk) rusak atau dipindahkan ke folder lain, maka obyek yang ada di dalam ruang kerja Maya akan menjadi *"missing reference object"* dan obyek kamera yang berada di dalam ruang kerja Maya tidak dapat digunakan.

Untuk mencegah hal demikian terjadi, *plugin* impor kamera dapat menjadi sebuah solusi. Pertama, *plugin* ini mengautomatisasi pekerjaan artis untuk melakukan impor atau referensi kamera. Cukup dengan menekan pada tombol *plugin* untuk melakukan impor atau referensi. Selain itu, poin penting dari *plugin* kamera ini adalah file Maya kamera (file induk) akan disimpan pada folder yang ditentukan. Dengan begini, waktu yang sebelumnya digunakan untuk mencari dimana file kamera disimpan dapat digunakan untuk melakukan pekerjaan lain.

Video pra-tinjau atau file *play blast* secara default akan disimpan pada folder dokumen Maya (C:/Document/maya/…/movies). Alamat folder ini sulit untuk diingat. File *play blast* milik artis dapat berjumlah ratusan, nantinya hal ini dapat menjadi masalah apabila artis tidak menyadari bahwa direktori (C:\) kehabisan ruang. Apabila direktori (C:\) kehabisan ruang, hal ini dapat menurukan performansi komputer, dimana performansi komputer merupakan hal yang sangat penting di dalam industri animasi. Hal ini lah yang akan

menjadikan *plugin play blast* merupakan solusi, karena *plugin* akan menyimpan file hasil *play blast* ke dalam direktori (D:\) yang dapat mengurangi resiko direktori (C:\) kehabisan ruang karena serta file hasil *play blast* menjadi lebih mudah ditemukan.

Dalam industri animasi, *plugin* Maya yang tersedia biasanya selain untuk tim previsualisasi. Misalnya penelitian untuk tim *modelling*, menghasilkan *plugin* python untuk bekerja sama dengan Maya dalam mencari fitur *modelling* yang dibutuhkan oleh artis *modelling.* Dengan adanya *plugin* tersebut, artis *modelling* tidak perlu mencari fitur yang dibutuhkan melalui menu bar *modelling*[1]. Atau penelitian untuk tim *rendering*, yang membuat plugin untuk melakukan *rendering* dengan kontrol jarak jauh dan menggunakan teknologi Dropbox untuk menyimpan hasil *render* di server Dropbox[2]. Ada pula penelitian untuk tim animasi yang menghasilkan *plugin* untuk membuat *walk cycle* karakter dengan mengkombinasikan *key frame* dan prosedur animasi dengan tujuan untuk mebuat *walk cycle* realistis secara otomatis[3]. Dan juga *plugin* untuk membuat partikel yang dapat berotasi secara otomatis di sekitar karakter yang bergerak[4]. Selain itu ada pula *plugin* untuk membuat kontroler berdasarkan sketsa untuk *blend-shape* animasi wajah dimana kontroler di konstruksikan dari *stroke* yang terkonversi pada koordinat vertex terdekat[5].

Pada paper ini, kami mencoba untuk membuat *plugin* yang dapat digunakan oleh tim previsualisasi. *Plugin* yang dapat mengimpor dan mereferensi kamera dari spesifik folder, serta *plugin* yang menyimpan file hasil *play blast* di direktori (D:\) secara otomatis.

Tabel 1. Jurnal Terkait

Jurnal	Hasil Penelitian	Saran
Creation of 3D Modeling Layout by Maya Scripting oleh Md.Aslam Hossen	Penelitian menghasilkan *plugin* python untuk Maya yang dapat mencari dan menampilkan fitur menu *Modelling* yang dibutuhkan oleh artis *modelling*. Sehingga artis tidak perlu mencari fitur secara manual melalui menu bar.	Penelitian ini membutktikan bahwa *plugin* dapat dibuat khusus untuk sebuah tim. Penelitian dapat dikembangkan dengan melakukan penelitian pada tim lain, selain tim *modelling*. Sehingga masing-masing tim dapat memiliki *plugin* sesuai dengan kebutuhan tim.

II. PENELITIAN TERKAIT

Tiga dimensi (3D) merupakan salah satu teknik populer yang digunakan oleh banyak animator[6]. Konsep dan metode yang digunakan pada teknologi yang terbilang baru ini adalah dengan menciptakan dunia virtual di dalam komputer[7]. Pada dunia virtual tadi, figur 3D diberi kontroler dengan menggunakan tulang rangka virtual yang kemudian oleh animator di gerakkan dengan *key frame*[8]. Kemudian, dunia virtual tadi akan diberikan efek pencahayaan dan efek visual. Setelah seluruh proses diatas selesai, maka dunia virtual siap untuk di-*render*, dimana gambar yang dihasilkan oleh perangkat lunak menjadi sebuah adegan utuh dari produk 3D[9].

Proses-proses yang terjadi di dalam animasi sampai menghasilkan suatu produk masuk dalam sebuah *pipeline* produksi. *Pipeline* memiliki konsep untuk mencari cara bagaimana teknologi dan perangkat keras dapat bekerja secara selaras[10]. Sebuah *pipeline* harus dibuat sesuai dengan kebutuhan studio. Kebutuhan studio satu dan yang lain bisa saja berbeda sesuai dengan jenis studio nya, apakah studio servis, atau produksi mandiri. Jika *pipeline* yang dibuat peruntukannya adalah studio dengan produk sendiri, maka keseluruhan proses animasi harus masuk kedalam *pipeline*, termasuk proses previsualisasi. Proses previsualisasi dikerjakan sebelum animasi dan efek visual[11]. Previsualisasi merupakan jembatan penghubung antara dua dimensi dan tiga dimensi. Karena pada proses ini, gambar dan teks (disebut *storyboard*) dua dimensi divisualisasikan ke dalam dunia virtual tiga dimensi[2]. Previsualisasi digital di zaman sekarang ini membuat perencanaan menjadi lebih matang dan menghasilkan keluaran visual dengan kualitas tinggi[12]. Proses ini merupakan proses yang rumit karena harus melibatkan kemampuan teknik dan artistik dari artis.

Dalam perihal perangkat lunak seperti apa yang seharusnya digunakan untuk produksi tiga dimensi, Zhang dan Huang (2015), dan Collado (2016) menyetujui hal yang sama, bahwa Autodesk Maya merupakan perangkat lunak yang sangat berguna untuk produksi animasi. Pada penelitiannya, Collado (2016) menyimpulkan bahwa Autodesk Maya termasuk salah satu perangkat lunak yang rumit untuk pemula, namun perangkat lunak ini juga sangat serbaguna[4]. Sedangkan Zhang dan Huang (2015) menyebutkan bahwa teknologi 3D semakin matang, pada pengembangan teknologi serta aplikasinya. Banyak peneliti yang mencoba untuk melakukan percobaan penelitian virtual, mereka mengembangkan *plugin* Maya untuk menghasilkan adegan tiga dimensi secara otomatis

berdasarkan C++ API di Maya[13]. Selain itu, mengenai *plugin* Autodesk Maya, Wood (2014) menyebutkan bahwa pada tahun-tahun sekarang, banyak studio yang membuat skrip untuk mengautomatisasi pekerjaan yang berulang atau menambahkan fungsi spesifik untuk proyek yang dikerjakan studio[14]. Membuat *plugin* atau *tool* untuk orang yang sudah berpengalaman merupakan pekerjaan yang menantang karena pengetahuan mereka yang paling banyak mengenai seperti apa *tool* yang mereka perlukan[15].

III. METODE

Python merupakan salah satu bahasa pemrograman yang populer. Hal ini dikarenakan, python dapat digunakan untuk mengembangkan berbagai macam hal. Salah satunya adalah untuk membuat *plugin* Autodesk Maya[16]. Pembuatan kode skrip python di Maya, kode nya dapat digunakan untuk berbagai *tool* di dalam Maya [1].

Plugin untuk perangkat lunak dalam industri animasi 3D, khususnya untuk Maya, biasanya ditujukan untuk tim animator. Seperti *plugin* untuk membuat pergerakan berjalan[3], manipulator ekspresi wajah[5], dan masih banyak lagi. Sedangkan untuk tim previsualisasi, sangat jarang ditemukan *plugin* yang dapat membantu pekerjaan mereka. Oleh karena itu, kami memutuskan untuk membuat *plugin* impor kamera dan *plugin play blast* untuk tim previsualisasi.

A. *Plugin* Impor Kamera

Dalam industri animasi, jaringan komputer yang digunakan berbasis pada jaringan klien-server. Klien adalah komputer milik para artis dan server adalah komputer utama yang menyimpan seluruh aset animasi dan hasil kerja para artis. Selama ini, file Maya kamera tidak berada pada folder tertentu, sehingga membutuhkan waktu untuk mencari dimana file Maya kamera disimpan. Karena itu, kami memutuskan untuk menyimpan file Maya kamera di server, dan secara khusus ada di dalam folder proyek.

Plugin impor kamera, merupakan *plugin* yang membantu tim previsualisasi untuk mengimpor file kamera yang berwujud secara otomatis. Berwujud maksudnya, file kamera harus berekstensi Maya ASCII (.ma) dan berada pada folder yang telah ditentukan. Dengan begini, *plugin* akan berfungsi dengan baik.

Pertama, kami membuat file kamera Maya yang berperan sebagai kamera dari proyek yang dikerjakan. Maya sebgai perangkat lunak untuk animasi memiliki fitur untuk membuat kamera nya sendiri. Oleh karena itu, untuk penelitian ini kami menggunakan fitur kamera yang dimiliki Maya. Setelah kamera dibuat, kemudian kamera ini disimpan menjadi tiga file Maya kamera dengan nama yang berbeda, CameraX.ma, CameraY.ma, dan CameraZ.ma.

Langkah selanjutnya, kami membuat lingkungan kerja buatan untuk melakukan simulasi. Kami membuat direktori (X:\) yang berperan sebagai jaringan server. Direktori (X:\) ini sebenarnya hanyalah folder biasa yang berada di direktori lokal komputer yang dinamai Drive X. Folder Drive X ini akan di bagikan alamat direktori nya yang selanjutnya dipetakan menjadi drive jaringan. Dalam direktori (X:\) kami membuat tiga buah folder bernama Project X, Project Y, dan Project Z dimana di dalam folder-folder tersebut terdapat file Maya kamera yang telah dibuat sebelumnya. CameraX.ma di dalam folder Project X, CameraY.ma di dalam folder Project Y, dan CameraZ.ma di dalam folder Project Z.

Setelah lingkungan kerja buatan selesai dibuat, langkah selanjutnya adalah pengkodean *plugin* impor kamera. *Plugin* harus dapat melakukan impor kamera dari file kamera yang berada di jaringan server, oleh karena itu *plugin* harus terkoneksi dengan direktori skrip. Dengan python, kamu menggunakan sintaks ***sys.path.append*** untuk mengkoneksikan *plugin* dengan direktori (X :\SC\src\) yang kami jadikan sebagai root dari skrip. Sintaks diatas ditulis di dalam skrip mainUI. Untuk jendela impor, setiap proyek memiliki jendela nya masing-masing. Sehingga ada tiga jendela yang merepresentasikan masing-masing proyek. Setiap jendela proyek dapat di panggil dari jendela main UI. Jadi, untuk *plugin* impor kamera, kami memiliki empat file skrip, mainUI, proX, proY, dan proZ dengan ekstensi masing-masing (.py).

Agar *plugin* dapat berfungsi dengan baik, file hasil kompilasi dari tiga skrip yang merepresentasikan folder proyek, di simpan pada folder root. Jika file kompilasi tidak disimpan di dalam folder root, maka jendela impor kamera hanya dapat muncul sekali.

B. *Plugin Play Blast*

Untuk melihat hasil dari previsualisasi yang telah dibuat adalah dengan melakukan *play blast*. *Play blast* merupakan metode untuk merubah file kerja Maya menjadi video. Dengan begini, pergerakan figur, pergerakan kamera, warna tekstur dari figur atau objek lain yang berada di dalam ruang kerja Maya, dapat dilihat dan dilakukan pengecekan.

Biasanya, secara default file *play blast* akan tersimpan pada dokumen Maya yang berada di direktori (C:\). Direktori (C:\) (pada sistem operasi Windows) merupakan direktori dimana system berjalan. Jumlah dari file *playblast* bisa mencapai ratusan dengan ukuran per satu file nya dapat mencapai puluhan megabite. Hal ini dapat menyebabkan direktori (C:\) kehabisan ruang. Apabila hal ini terjadi, maka performansi komputer dapat menurun. Dimana, dalam industri animasi performansi komputer merupakan hal yang sangat krusial. Apabila performansi

menurun, maka waktu yang diperlukan untuk mengerjakan sebuah projek bisa jadi menjadi lebih lama. Maka dari itu, untuk menghindari direktori (C:\) kehabisan ruang, kami memutuskan untuk membuat *plugin play blast* yang dapat menyimpan file ke dalam direktori (D:\) secara otomatis.

Untuk menjalankan *plugin*, tidak seperti *plugin* impor kamera, *plugin* ini tidak membutuhkan lingkungan buatan, dan juga tidak perlu terkoneksi dengan folder root. *Plugin* ini hanya membutuhkan alamat direktori folder untuk menyimpan file *play blast* yang kami tentukan pada direktori (D:\MyProject). Artis tidak perlu membuat folder MyProject karena secara *plugin* akan membuat folder secara otomatis.

IV. HASIL DAN PEMBAHASAN

A. Hasil

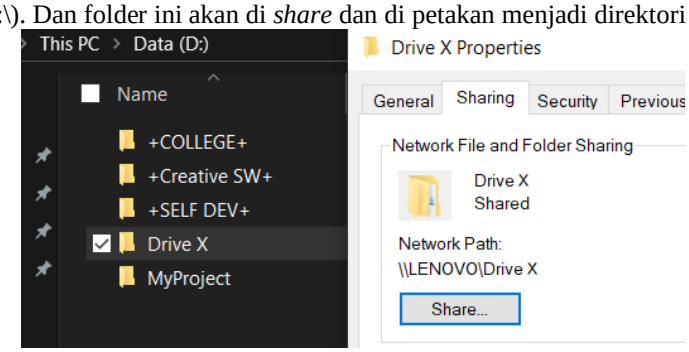
Penelitian dimulai dengan melakukan wawancara kepada artis previsualisasi untuk mengumpulkan data kebutuhan pengguna. Data kebutuhan pengguna ini dijadikan sebagai referensi bagi kami untuk membuat *plugin* sesuai dengan apa yang dibtuhukan oleh artis/pengguna.

Tabel 2. *User Requirement*

User Requirement	Operational Needs	Performance Parameter
Pilihan Proyek	Untuk memilih proyek yang akan dikerjakan	Akan muncul jendela antarmuka setelah *plugin* dijalankan agar pengguna dapat memilih proyek yang akan dikerjakan
Kamera Importir	Untuk memilih fungsi impor pada file kamera sesuai dengan proyek yang telah dipilih	File kamera menjadi objek kamera yang terimpor di dalam ruang kerja Maya
Kamera *Referencer*	Untuk memilih fungsi *reference* pada file kamera sesuai dengan proyek yang telah dipilih	File kamera menjadi objek kamera yang tereferensi di dalam ruang kerja Maya
Lokal *play blast*	Untuk menyimpan file *play blast* bukan pada folder dokumen Maya	Setelah melakukan *playblast* menggunakan *plugin*, pengguna akan memiliki file *playblast* yang tersimpan pada direktori (D:\MyProject)

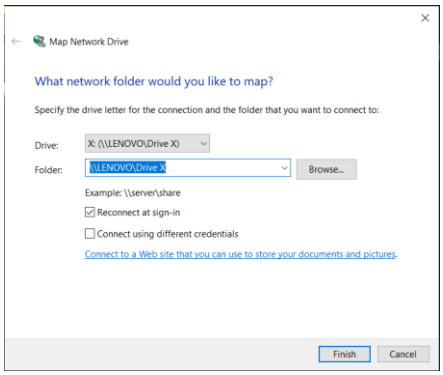
Perbedaan antara file hasil impor dan hasil referensi adalah file hasil impor menjadi file independen yang berarti apabila file kamera yang ada di jaringan server rusak atau terhapus, objek kamera di dalam ruang kerja Maya tetap dapat digunakan. Namun, tidak dengan file hasil referensi, karena file hasil referensi menjadi objek referen yang harus tersambung dengan file kamera Maya.

Setelah mengumpulkan data, kami membuat lingkungan kerja untuk simulasi. Kami membuat folder bernama Drive X di direktori (D:\). Dan folder ini akan di *share* dan di petakan menjadi direktori jaringan.



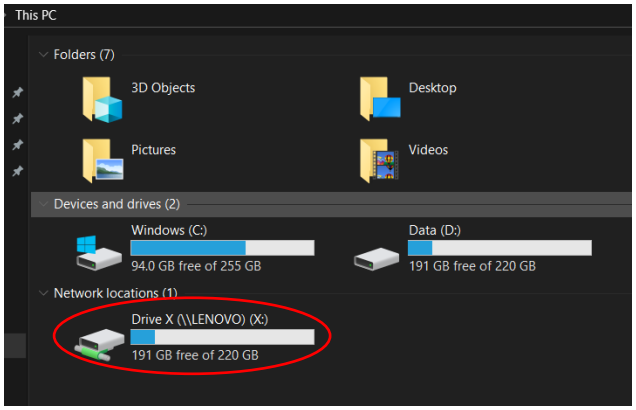
Gambar 1. Folder Drive X pada direktori (D:\)

Parameter yang membuktikan bahwa folder Drive X sudah ter*share* adalah dibawah “Network Path :” akan terlihat alamat direktori dari folder. Alamat direktori ini di salin pada jendela Map Network Drive untuk membuat folder Drive X menjadi direktori (X:\) di dalam komputer.



Gambar 2. Pemetaan direktori (X:/)

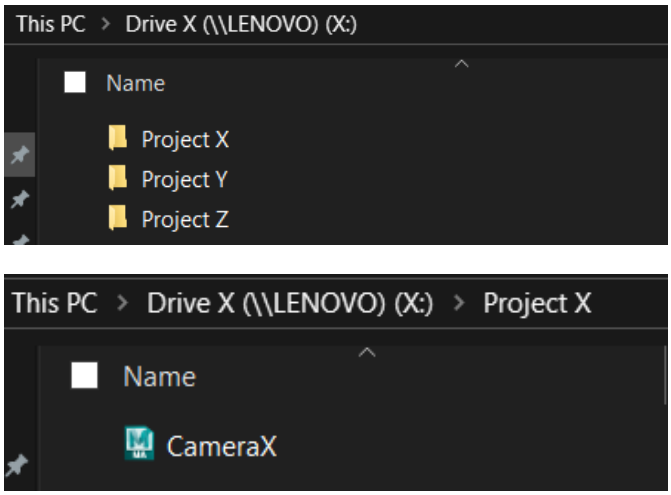
Setelah tombol *Finish* di klik dan tidak muncul masalah, kami mempunyai direktori (X:\) yang terlihat pada Gambar 3.



Gambar 3. Direktori (X:/) selesai dibuat

1) *Plugin* Impor Kamera

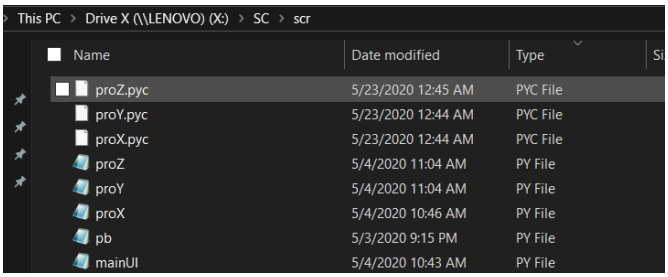
Sebelum memulai pengkodean, kami membuat tiga folder yang berperan sebagai folder proyek bernama Project X, Project Y, dan Project Z pada direktori (X:\). Dan di dalam setiap folder terdapat file Maya kamera. File kamera di dalam folder-folder ini lah yang akan di panggil dari *plugin*, apakah untuk diimpor atau untuk di *reference*.



Gambar 4. Folder Proyek dan File Maya Kamera di dalamnya

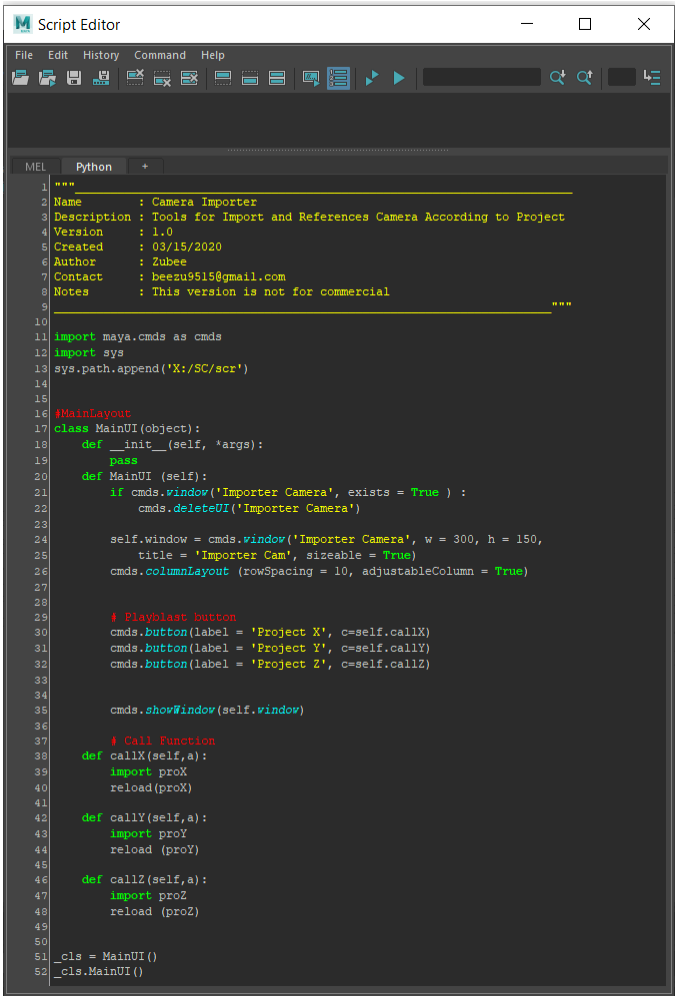
Setelah lingkungan virtual sudah selesai dibuat, selanjutnya kami mulai ke tahap pengkodean *plugin*. Kami menulis kode dengan perangkat lunak *Sublime Text*. Kode pertama yang dibuat adalah untuk jendela awal yang akan muncul ketika kode di jalankan. Kode ini kami namai "mainUi.py". Pada kode jendela ini, kami memberikan perhatian khusus karena kami harus memastikan setiap folder proyek terkoneksi dengan di dalam kode ini. Untuk melakukan koneksi, kami membutuhkan alamat folder *root*. Alamat folder *root* ini akan di letakkan pada sintaks *sys.path.append*.

Untuk kode jendela setiap proyek, kami menyimpan file kode yang sudah terkompilasi ke dalam folder *root* dari *plugin* yang kami deklarasikan di kode mainUI.py. File kode yang sudah terkompilasi dari python untuk Maya akan memiliki ekstensi *.pyc*. Alasan kenapa kami hanya menggunakan file kode yang sudah terkompilasi adalah karena kode dari setiap proyek ini tidak dijalankan secara langsung di dalam Maya, melainkan akan di panggil dari jendela mainUI.

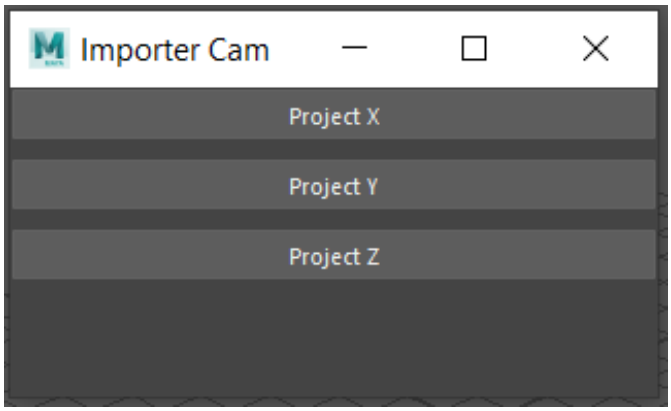


Gambar 5. Folder Root

Setelah kami menyelesaikan kode mainUI.py, kami menyalin kode pada tab Phython di jendela *Script Editor* untuk menjalankan kode.

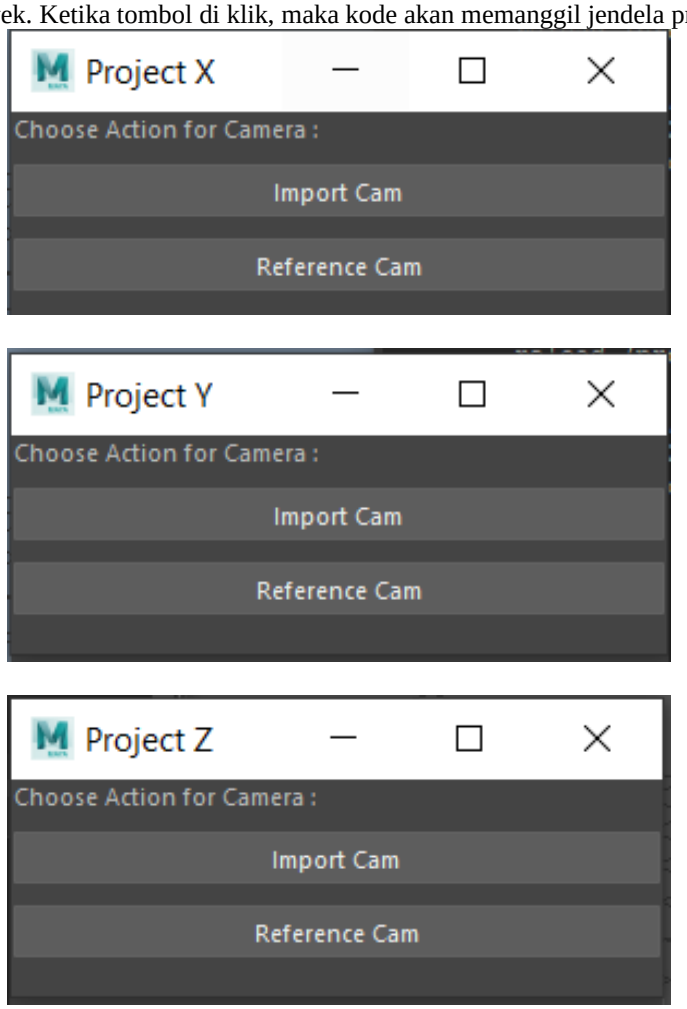


Gambar 6. Jendela *Script Editor* dengan kode mainUI



Gambar 7. Antarmuka jendela Impor Kamera

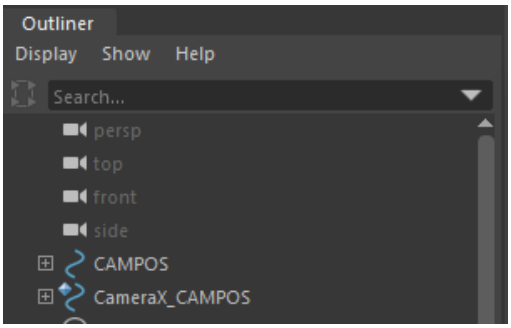
Jendela antar muka impor kamera di desain dengan menggunakan tombol, setiap tombol merepresentasikan folder proyek yang ada di direktori jaringan server. Sehingga tombol ini terhubung dengan folder proyek. Ketika tombol di klik, maka kode akan memanggil jendela proyek.



Gambar 8. Antarmuka Jendela Proyek

Gambar 8 memperlihatkan jendela untuk setiap proyek. Pada jendela proyek, pengguna dapat memilih antara untuk mengimpor atau me*reference* kamera. Untuk jendela antar muka setiap proyek, kami menggunakan desain yang sama, satu-satu nya yang berbeda adalah nama dari jendela antar mukanya.

Untuk memastikan apakah fungsi impor dan *reference* berjalan seperti semestinya, kami mencoba untuk melakukan impor dan *reference* dengan jendela Project X. Kami mencoba kedua fungsi tersebut dalam satu ruang kerja.



Gambar 9. Kamera yang telah diimpor dan di*reference*

Gambar 9 memperlihatkan file kamera yang berhasil diimpor dan dereference pada tab Outliner, yang berarti menunjukkan bahwa fungsi impor dan *reference* berjalan dengan baik. Obyek "CAMPOS" menunjukkan camera yang terimpor. Sedangkan obyek "CameraX_CAMPOS" menunjukkan kamera yang di *reference*.

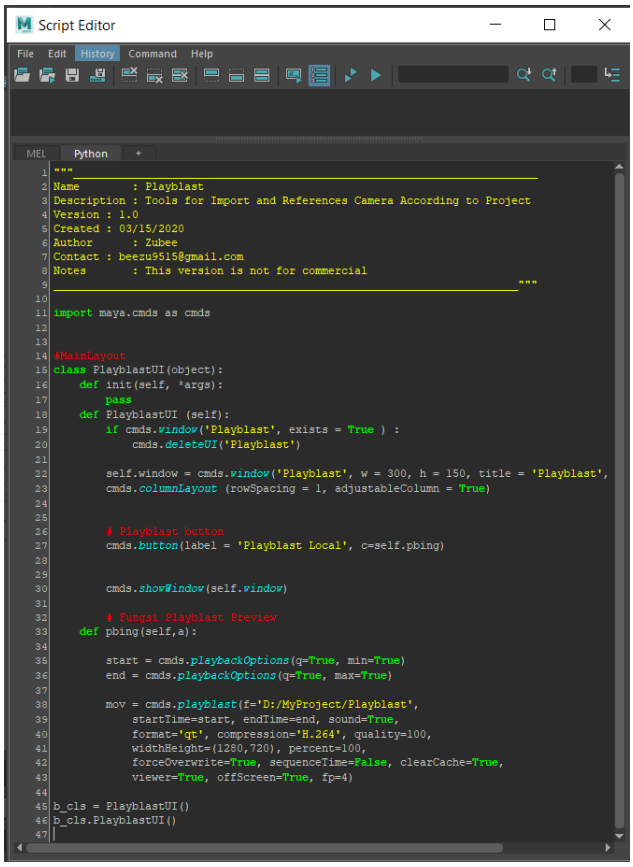
TABEL 3 . VALIDASI *PLUGIN* IMPOR KAMERA

Parameter	Tanpa Plugin	Dengan Plugin
Waktu untuk mencari file kamera	1 menit	10 detik
Jumlah Klik Mouse	15-20	3-5
Fitur Impor Kamera	Manual	Otomatis
Fitur Reference Kamera	Manual	Otomatis
Folder Penyimpanan File Kamera	Dimana saja	Folder yang sudah ditentukan

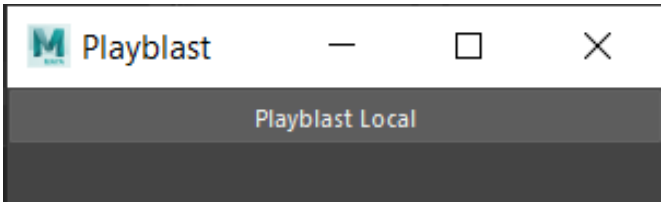
Manual pada tabel diatas maksudnya, untuk melakukan impor dan *reference* file kamera artis harus menggunakan fitur yang ada pada menu File Maya, kemudian mencari file kamera yang tersimpan di dalam komputer pada jendela impor/*reference* dari Maya. Otomatis berarti, dengan *plugin* artis hanya perlu melakukan klik pada tombol jendela antarmuka *plugin.*

2) *Plugin Play Blast*

Pada *plugin play blast*, tidak seperti *plugin* impor kamera, kami tidak membutuhkan lingkungan buatan. *Plugin* ini dapat berdiri sendiri. Kami hanya perlu menjalankan kode *plugin* secara langsung di dalam Maya. Setelah kode selesai dibuat, kami menyalin kode ke tab Python di jendela *Script Editor.*

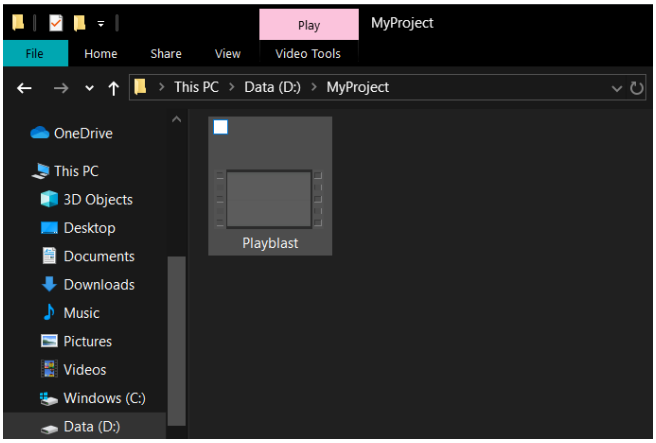


Gambar 10. Jendela *Script Editor* dengan kode *plugin play blast.*



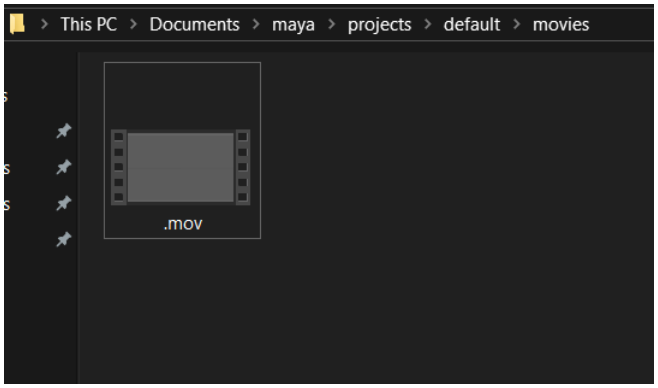
Gambar 11. Jendela Antarmuka *Play Blast*

Untuk antar muka *play blast*, kami juga memilih untuk mendesain jendelanya dengan button. Ketika tombol “Playblast Local” di klik, maka proses *play blast* akan dilakukan sesuai dengan durasi yang sudah ditentukan di ruang kerja Maya. File hasil dari proses *play blast* ini akan tersimpan di direktori (D:\MyProject) dan bernama Playblast.mov. Gambar 12 menunjukkan hasil dari proses *play blast.*



Gambar 12. Penyimpanan File Hasil *Play Blast* dengan *Plugin*

Autodesk Maya sebenarnya memiliki fitur *play blast* nya sendiri. Namun ketika fitur ini digunakan, file hasil *play blast* akan tersimpan di (C:/Document/maya/…/movies). Alamat penyimpanan default Maya ini cukup panjang dan dapat dikatakan agak sulit untuk diingat.



Gambar 13. Penyimpanan File Hasil *Play Blast* dengan Fitur Maya

Gambar 12 dan 13 menunjukkan perbedaan tentang sebenrapa banyak folder yang harus artis buka sebelum artis menemukan file hasil *play blast.* Untuk menunjukkan perbedaan yang lebih jelasnya, dapat dilihat pada Tabel 3.

TABEL 4. VALIDASI *PLUGIN PLAY BLAST*

Parameter	Tanpa Plugin	Dengan Plugin
Pengaturan Movie / Video *Play Blast*	Manual	Otomatis
Jumlah Klik Mouse	15-20	2
Jumlah Klak Mouse	1	0
Waktu untuk menemukan File Movie / Video *Play Blast*	30 – 50 detik	10 detik
Direktori Penyimpanan File Movie / Video *Play Blast*	(C:/)	(D:/)

B. Pembahasan

Plugin yang kami buat, keduanya memberikan kesempatan untuk artis previsualisasi mengerjakan pekerjaan mereka lebih otomatis. Pertama, dibandingkan dengan melakukan impor kamera secara manual, dimana artis harus mencari folder dimana file kamera disimpan, mereka dapat melewati tahap pencarian dengan melakakan klik pada tombol *plugin.* Dan juga dengan disimpannya file kamera pada folder yang telah ditentukan, dapat menghemat waktu artis previsualisasi untuk mencari file kamera. Waktu yang mereka gunakan sebelumnya untuk mencari file kamera dapat digunakan untuk mengerjakan pekerjaan lain.

Kedua, untuk *plugin play blast*, dibandingkan dengan mencari file hasil *play blast* di folder dokumen Maya, akan lebih mudah untuk memukan file ini di direktori terdekat. Selama kami melakukan observasi ketika artis melakukan pekerjaan meraka, direktori yang paling sering dikunjungi adalah direktori (D:\). Atas observasi ini, kami memutuskan untuk menyimpan hasil *play blast* di direktori (D:\) sehingga akan lebih mempermudah untuk artis. Dan poin penting dari *plugin* ini adalah, dapat menurukan kemungkinan direktori (C:\) kehabisan ruang karena file hasil *play blast.*

V. KESIMPULAN

Pada peneilitan ini, dapat kami tunjukkan bahwa *plugin* untuk tim previsualisasi juga penting. *Plugin* ini difokuskan untuk membuat pekerjaan tim previsualisasi lebih mudah dan lebih cepat. Konsep dari *plugin* impor kamera adalah untuk melakukan impor kamera secara cepat dan mudah kedalam ruang kerja Autodesk Maya. Tidak perlu ada lagi proses pencarian dimana file kamera disimpan, ini dapat menghemat waktu artis untuk

bekerja. Sedangkan, konsep dari *plugin play blast* adalah bagaimana untuk mempermudah mencari file hasil *play blast* dan menghemat ruang dari direktori (C:\).

Kedua *plugin* ini tidak saling berhubungan. Maksudnya, kedua *plugin* dapat berjalan masing-masing tanpa ada ketergantungan. *Plugin* impor kamera dapat berjalan lancar tanpa *plugin play blast* dan juga sebaliknya. Namun, dapat kami katakan bahwa kedua *plugin* ini saling melengkapi. *Plugin* impor kamera sebagai awalan dan *plugin play blast* sebagai akhiran. Dapat kami katakana seperti tersebut diatas adalah karena artis akan melakukan impor kamera sebelum melakukan pekerjaan previsualisasi dan mereka membutuhkan file hasil *play blast* untuk melakukan pengecekan pada hasil kerja previsualisasi nya. Jadi *plugin* ini sangat membantu pekerjaan dari artis previsualisasi.

Namun, dengan keterbatasan pengetahuan yang kami miliki mengenai python untuk Maya, kami menyadari bahwa desain antarmuka, fungsi *plugin*, dan algoritma yang ada masih jauh dari sempurna. Misalnya saja seperti desain dari jendela antarmuka *plugin* masih terlalu monotone, dan jauh dari kata atraktif. Kemudian untuk fungsinya, dalam melakukan impor kamera, kedua fungsi impor dan *reference* dapat digunakan dalam satu antar muka yang sama, sehingga dapat menyebabkan terdapat dua obyek kamera di dalam satu ruang kerja Maya. Hal ini tidak di benarkan dalam dunia industri animasi. *Plugin* impor kamera dapat ditingkatkan dengan menambahkan fungsi yang dapat mencegah terdapat dua kamera dengan menampilkan peringatan atau apabila salah satu fungsi sudah digunakan, maka fungsi lain tidak dapat digunakan. Dan tentu saja algoritma dari *plugin* ini masih sangat kurang dalam banyak hal,

Ruang untuk meningkatkan *plugin* untuk tim previsualisasi akan selalu ada. Tidak harus seperti yang kami sebutkan diatas. Kami percaya akan banyak jalan untuk meningkatkan *plugin*. Di masa depan, kami berharap akan ada *plugin* yang khusus diperuntukkan bagi tim previsualisasi yang lebih menarik, dan memiliki lebih banyak fungsi.

VI. DAFTAR PUSTAKA

[1] S. By, “Creation of 3D Modeling Layout by Maya Scripting,” no. December, 2018.

[2] A. Bernik and D. Galetic, “Remote Rendering Control Using Python Scripts and Dropbox Technology,” *Teh. Glas.*, vol. 12, no. 2, pp. 62–67, 2018, doi: 10.31803/tg-20170919221607.

[3] M. Guindy and R. Elias, “Happy Feet Maya Plugin to Generate Customized Gait Cycles for 3D Characters,” pp. 372–377, 2017.

[4] G. E. G. Collado, “Modeling and Python Scripting in Maya,” 2016.

[5] O. Cetinaslan, V. Orvalho, and J. P. Lewis, “Sketch-Based Controllers for Blendshape Facial Animation,” 2015, doi: 10.2312/egsh.20151006.

[6] S. M. H. Asraf and S. Z. S. Idrus, “Hybrid Animation: Implementation of Three-Dimensional (3D) Animation,” *J. Phys. Conf. Ser.*, vol. 1529, p. 022094, 2020, doi: 10.1088/1742-6596/1529/2/022094.

[7] C. F. Xu and X. C. Zhou, “Three-dimensional Animation Technology in Showing the Application of the Art of Design,” *MATEC Web Conf.*, vol. 44, pp. 1–4, 2016, doi: 10.1051/matecconf/20164402044.

[8] R. Kushwaha, “Procedure of Animation in 3D Autodesk Maya : Tools & Techniques,” no. August, 2016, doi: 10.5121/ijcga.2015.5402.

[9] Y. Gao, “Research on Rendering Speed Based on Maya and Mental Ray,” no. Jimec, pp. 395–398, 2016, doi: 10.2991/jimec-16.2016.70.

[10] A. Evans, J. Agenjo, and J. Blat, “Designing a Multiplatform Pipeline for 3D Scenes.”

[11] H. K. Choi and S. H. Cho, “Development of Effective Pre-Visualization Authoring Tool Using Conversion Technologies-Based on Film Storyboard Application,” *Cluster Comput.*, vol. 17, no. 2, pp. 585–591, 2014, doi: 10.1007/s10586-013-0303-6.

[12] T. Muender, “Analysis of Previsualization Tasks for Animation , Film and Theater,” pp. 1–6, 2019.

[13] Q. Zhang and Z. Huang, “Implementation of Computer Graphics Algorithm Platform Based on Autodesk Maya Guangdong Province Science and Technology Project (2014A020212684),” no. Icmmcce, pp. 2758–2762, 2015.

[14] A. Wood, “Behind the Scenes : A Study of Autodesk Maya,” pp. 142–143, 2014, doi: 10.1177/1746847714546247.

[15] M. K. Berthelsen, B. N. Overgaard, and A. M. Thomsen, “Framing-Based Camera Tool for Artists in Game Development.”

[16] T. Vinothraj, M. Fareez, and A. . Sayeth Saabith, “Python Current Trend Applications- an Overview,” *Int. J. Adv. Eng. Res. Dev.*, vol. 2, no. 2, pp. 61–67, 2017.