

BAB III

ANALISA DAN PERANCANGAN

III.1. Analisis Masalah

Perkembangan game dari skala kecil maupun besar sangat bervariasi yang dapat dimainkan oleh siapa saja tanpa memandang umur, dari anak-anak hingga orang dewasa. Game berkembang begitu pesat dengan jenis platform yang beragam mulai *console*, *mobile*, *PC* dan lain sebagainya, serta dapat dimainkan secara *online* maupun *offline*.

Salah satu permainan (*game*) komputer yang cukup menarik adalah *game warplane*, dimana dalam permainan ini pemain berusaha untuk menghancurkan para musuh-musuhnya dan begitu juga sebaliknya agar salah satu hancur dan bisa menentukan nilai terbesar.

Dalam hal ini membahas tentang bagaimana algoritma *greedy* bekerja pada *game warplane*. Dalam permainan *game warplane*, pemilihan tempat sangat penting diperhatikan. Salah satu tolak ukur keberhasilan pemain dalam bermain *game warplane* yaitu posisi amunisi terhadap pesawat musuh, dan *score* yang didapat mengatasi optimasi secara efektif dan optimum untuk mendapatkan skor maksimal.

Permainan *warplane* merupakan *genre game* minimalis dengan *sub-genre* dari *Real-Time Strategy* yang dirancang untuk mengasah kemampuan logika. Permainan ini sangat sederhana, yakni pemain mencoba menghentikan pesawat musuh yang melihat dari atas meriam. Dalam *warplane* akan mendapatkan nilai

setelah amunisi tepat mengenai pesawat musuh yang sedang melintas. Pemain dapat menggeser pesawat pemain untuk penempatan sasaran yang lebih tepat mendapatkan nilai maksimal.

III.1.1. Penerapan Algoritma Greedy

Sebelum melakukan perancangan terhadap sistem, penulis terlebih dahulu melakukan analisa tentang sistem yang akan dirancang. Dalam analisa ini, penulis melakukan analisa mengenai fasilitas apa yang disediakan dalam sistem yang akan dirancang dan langkah-langkah pembuatan *game warplane*.

Algoritma greedy merupakan metode yang paling populer untuk memecahkan persoalan optimasi. Sesuai dengan nama dari algoritma, algoritma greedy bersifat rakus dan tamak. “*Take what you can get now!*” adalah prinsip utama dari algoritma greedy dengan cara membentuk solusi langkah perlangkah (*step by step*). Pada setiap langkahnya, algoritma *greedy* mengambil keputusan berdasarkan langkah tercepat (minimasi dan maksimasi) tanpa mengambil konsekuensi langkah dan pilihan di depan (langkah selanjutnya).

Penerapan algoritma pada perancangan aplikasi *game warplane* ini penulis lakukan dengan mengumpulkan materi-materi yang berhubungan dengan *game* pesawat perang, tampilan 3D, dan lain sebagainya. Kemudian penulis hanya menggunakan materi-materi yang paling dekat dengan materi animasi pesawat dan *game flash*.

Pada metode algoritma *greedy* hanya satu rangkaian keputusan yang dapat dihasilkan oleh karena itu mengapa algoritma digunakan untuk menentukan

bagaimana penyajian informasi yang sesuai dengan aplikasi *game warplane*. Oleh karena itu digunakan algoritma *greedy*, untuk mengetahui informasi-informasi tersebut dalam setiap tampilan yang langsung pada pokok pembahasan yang diperlukan.

Algoritma *greedy* digunakan untuk mengetahui langkah-langkah pemecahan masalah dan membentuk solusi langkah per langkah, sesuai dengan arti tersebut, algoritma ini mengambil pilihan yang terbaik yang dapat diperoleh pada saat itu tanpa memperhatikan konsekuensi ke depan. Penulis membahas aplikasi algoritma untuk menentukan bagaimana menyajikan informasi yang merupakan intisari dari aplikasi *game warplane*. Elemen-elemen algoritma *greedy* pada permainan *warplane* adalah sebagai berikut :

1. Himpunan kandidat, himpunan titik-titik pada lintasan pesawat musuh.

Himpunan Laser dikatakan sama dengan Himpunan Pesawat jika hanya elemen Laser merupakan elemen Pesawat dan sebaliknya setiap elemen Pesawat merupakan elemen Laser. Sebaliknya, A tidak sama dengan Pesawat.

Notasi : $A=B \iff A \subseteq B \text{ dan } B \subseteq A$

2. Fungsi seleksi, memilih titik yang sesuai pada senjata pesawat. Titik yang dipilih adalah titik yang memiliki AOE (area yang memungkinkan dilakukan serangan) paling besar.

Setiap elemen Laser dipetakan secara unik ke suatu elemen Musuh; kumpulan dari pemetaan-pemetaan ini disebut fungsi atau pemetaan dari Laser ke Musuh.

$f : A \rightarrow B$

3. Fungsi kelayakan, untuk mengecek apakah jumlah amunisi dapat digunakan lagi atau tidak. Permainan akan berakhir bila jumlah amunisi telah habis digunakan.

$$\frac{p \vee q}{\sim p} \\ \hline \therefore q$$

4. Fungsi objektif, untuk memaksimalkan skor dengan memperoleh AOE sebesar mungkin. Jumlah skor tertinggi didapat dari jumlah amunisi yang lebih banyak mengenai pesawat musuh yang melintas.

Proposisi berlaku untuk $n=1$ karena $n=1/2(1)(1+1)$, sehingga $P(1)$ benar, yang mana $P(n+1)$. Sehingga, $P(n+1)$ benar bilamana $P(n)$ benar. Menurut prinsip induksi, P berlaku untuk setiap n .

III.1.2. Analisis Sistem

Analisis sistem dapat didefinisikan sebagai “Penguraian dari suatu sistem informasi yang utuh ke dalam bagian- bagian komponennya dengan maksud untuk mengidentifikasikan dan mengevaluasi permasalahan-permasalahan, kesempatan dan hambatan yang terjadi dan kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan perbaikannya” Di dalam tahap analisis sistem terdapat langkah-langkah dasar yang harus dilakukan oleh analisis sistem sebagai berikut :

1. *Identify*, yaitu mengidentifikasi masalah.
2. *Understand*, yaitu memahami kerja dari sistem yang ada.
3. *Analyze*, yaitu menganalisis sistem.
4. *Report*, yaitu membuat laporan hasil analisis.

III.1.3. Analisis Perancangan Game

Game ini dibuat untuk diimplementasikan pada PC dengan sistem operasi *Windows*. *Game* ini bersifat single player, tugas utama pemain dalam game ini adalah menembak pesawat musuh yang mulai masuk ke kota. Berikut rincian game yang akan dibuat :

1. *Game* petualangan.
2. Sistem permainan singleplayer.
3. Tokoh yang ada dalam game ini adalah pesawat tempur pemain yang harus menembak pesawat tempur musuh hingga meledak.
4. Senjata yang dipakai adalah tembakan laser.
5. Bahasa yang di gunakan adalah Bahasa Inggris.
6. Peralatan input yang digunakan adalah keyboard.
7. Tombol yang digunakan dalam permainan.
 - a. Pemain akan menggunakan keyboard untuk tombol panah kiri , kanan, atas dan bawah untuk menggerakkan pesawat.
 - b. Tombol Spasi pada keyboard untuk mengeluarkan tembakan peluru yang keluar dari pesawat.
8. Jika pesawat pemain tertembak, Pesawat pemain akan memiliki daya tahan, daya tahan / nyawa akan berkurang jika kita terkena pesawat musuh dan tertembak pesawat musuh.

III.2. Strategi Pemecahan Permasalahan

Sebelum melakukan perancangan terhadap sistem, penulis terlebih dahulu melakukan analisa tentang sistem yang akan dirancang. Dalam analisa ini, penulis melakukan analisa mengenai fasilitas apa yang disediakan dalam sistem yang akan dirancang dan langkah-langkah pembuatan game pesawat tersebut.

III.3. Perancangan

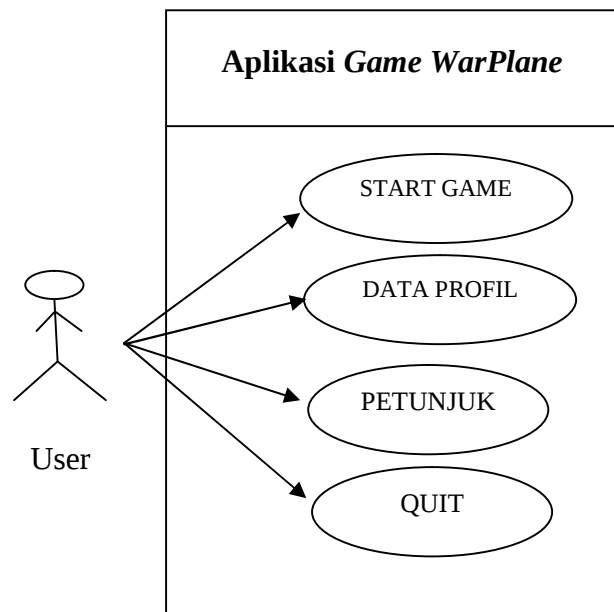
Perancangan aplikasi merupakan perancangan yang dilakukan untuk merancang sebuah aplikasi dengan menggunakan salah satu bahasa pemrograman, dalam kasus ini penulis merancang sebuah aplikasi *game warplane* dengan menggunakan *3D max* dan memakai bahasa action script 2.0 pada *Adobe Flash Professional CS 6*. Secara umum gambaran sistemnya adalah game ini dibuat untuk di implementasikan pada laptop dengan sistem operasi windows 7 dan XP. Game ini bersifat *singleplayer*, tugas utama pemain adalah menembak pesawat musuh.

Struktur data yang digunakan penulis dalam perancangan perangkat lunak adalah *Unified Modelling Language (UML)*. UML adalah bahasa spesifikasi standar untuk mendokumentasikan, menspesifikasikan dan membangun sistem perangkat lunak. UML yang digunakan meliputi perancangan *Diagram Use Case*, *Activity Diagram* dan *Squence Diagram*.

III.3.1. Rancangan Use Case Diagram

Use case diagram ini menggambarkan *simulasi* yang akan dibuat untuk sebuah aplikasi *game warplane* tersebut. Sedangkan pengguna atau *user* melihat

aplikasi tersebut dengan melalui tombol. Sehingga pengguna dapat menjalankan program aplikasi pembelajaran. Berikut rancangan *use case diagram* terdapat pada gambar III.1.

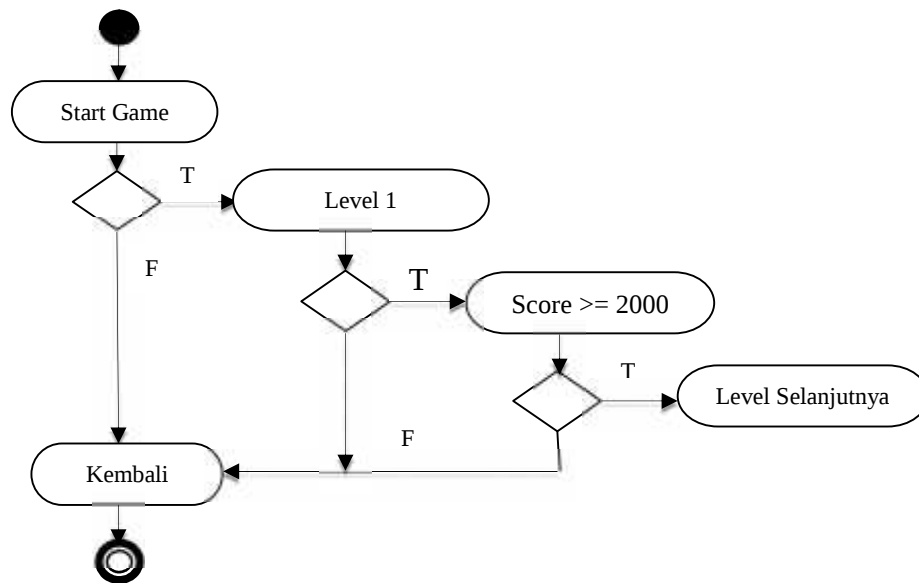


Gambar III.1. Rancangan *use case diagram* menu utama

Pada gambar perancangan *use case diagram* diatas, menu utama terdiri dari tiga menu, yaitu : *start game*, *game credit*, dan *quit*.

III.3.2. Activity Diagram

Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. Berikut merupakan *Activity diagram* perancangan aplikasi *game warplane* pada Gambar III.2.

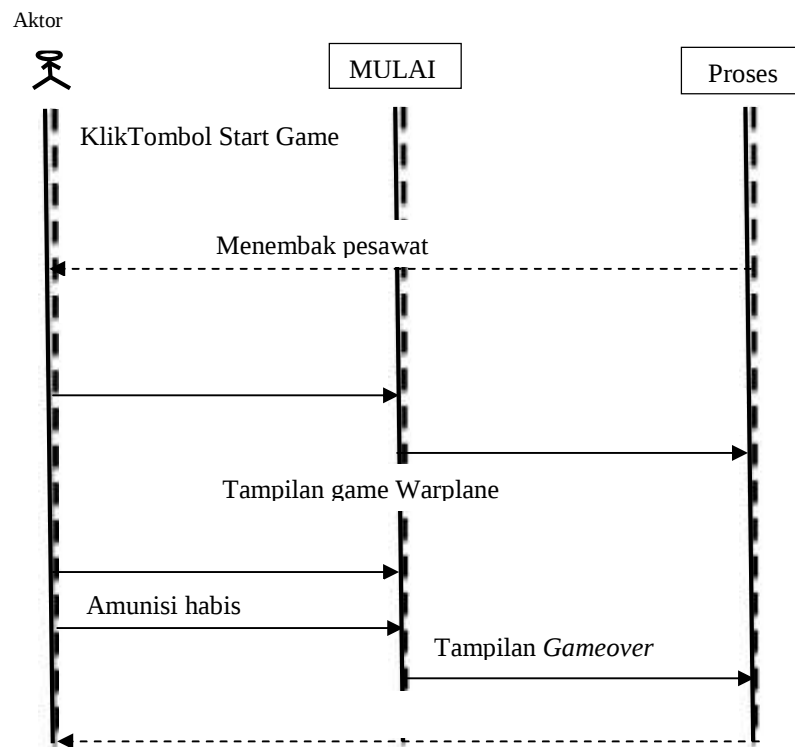


Gambar III.2. Rancangan *activity diagram* aplikasi *game warplane*

III.3.3. Sequence Diagram

Sequence diagram adalah suatu diagram yang menampilkan interaksi-interaksi antar objek atau sistem yang disusun pada sebuah urutan atau rangkaian waktu. Digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian untuk menggambarkan output tertentu. Rancangan *sequence diagram* aplikasi *game warplane* dapat dilihat sebagai berikut :

Adapun perancangan *sequence diagram* tombol dapat dilihat pada gambar III.3.



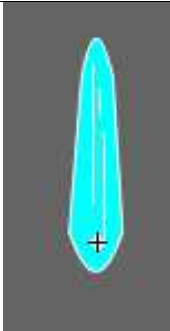
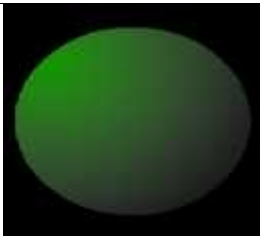
Gambar III.3. Rancangan *Sequence Diagram Start Game*

Gambar rancangan *sequence diagram start game* merupakan halaman menjalankan permainan *warplane*.

III.4. *Storyboard*

Storyboard pada aplikasi *game warplane* ini menjelaskan objek-objek apa yang dapat ditemukan oleh pengguna (*user*) di aplikasi tersebut.

Tabel III.1. *Storyboard*

No.	Gambar Objek-objek	Keterangan
1		Pesawat Pemain
2		Pesawat Musuh
3		Laser pesawat
4		Perisai pesawat

III.5. Rancangan Dan Pembuatan Antar Muka

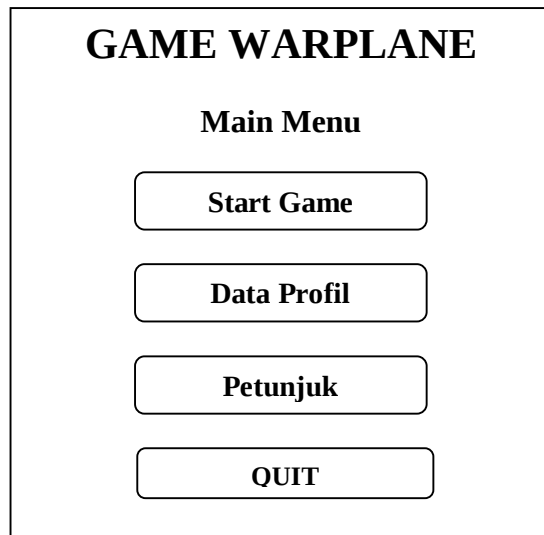
Dalam pembuatan antar muka game pesawat tersebut ini ada tiga tahapan, tahap pertama adalah antar muka menu utama, kedua adalah antar muka untuk menu petunjuk permainan, ketiga adalah antar muka permainan game.

Flowchart merupakan gambar atau bagan yang memperlihatkan urutan dan hubungan antar proses beserta pernyataannya. Gambaran ini dinyatakan dengan simbol dan dengan demikian setiap simbol menggambarkan proses tertentu. Hubungan antar proses digambarkan dengan garis penghubung.

Aplikasi *Flowchart* menggambarkan tahapan proses suatu sistem, termasuk sistem multimedia. Program *flowchart* menggambarkan urutan - urutan instruksi dari suatu program komputer. Berikut ini adalah flowchart sistem rancangan permainan game pesawat.

III.5.1. Tampilan Menu Utama

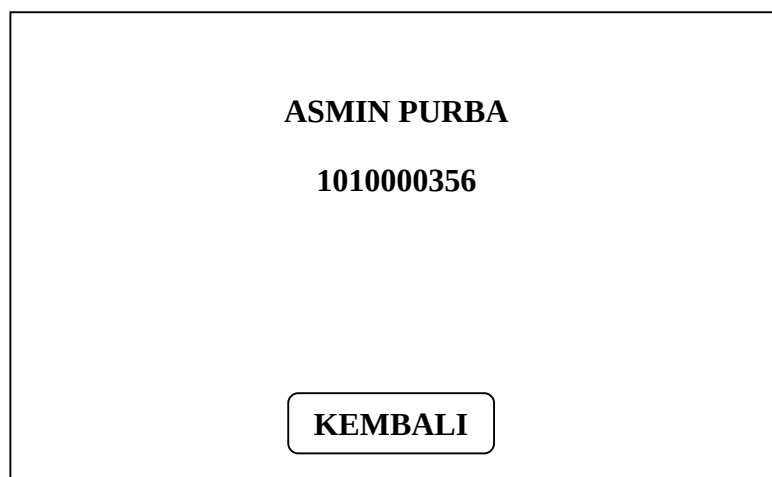
Tampilan Menu utama memiliki empat pilihan yaitu *Start Game*, *Data Profil*, *Petunjuk*, *Keluar*. Maka proses akan dilanjutkan ketampilan *Start Game*. Jika tombol *Data Profil* di pilih, maka akan di lanjutkan ke tampilan *Data Profil*. Jika tombol *Petunjuk* dipilih, maka proses akan dilanjutkan ke tampilan *Petunjuk*.



Gambar III.4 Tampilan Menu Utama

III.5.2. Rancangan Tampilan Data Profil

Pada tampilan menu *Data Profil* merupakan menu yang harus kita perhatikan secara seksama bagaimana mengatur hidup/mati suara musiknya game tersebut. Ketika kita membuka menu *Pengaturan* di tampilan tersebut akan memberi pilihan dengan kita tentang mengatur suara game pesawat tersebut.



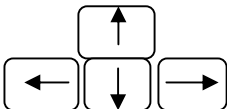
Gambar III.5 Tampilan Data Profil

III.5.3. Rancangan Tampilan Menu Petunjuk

Pada tampilan menu *Petunjuk* merupakan menu yang harus kita dibaca dan di perhatikan secara seksama bagaimana menggunakan atau menjalankan game tersebut. Ketika kita membuka menu *petunjuk* di tampilan tersebut akan memberi penjelasan tentang game pesawat tersebut.

PETUNJUK PERMAINAN

1. Pemain akan menggunakan keyboard untuk tombol panan kiri, kanan, atas bawah untuk menggerakan pesawat.



Keterangan tombol pada keyboard :

1. Tombol atas, di gunakan untuk menggerakan pesawat kearah atas
2. Tombol bawah, digunakan untuk menggerakan pesawat kearah bawah
3. Tombol kanan, digunakan untuk menggerakan pesawat kearah kanan
4. Tombol kiri, digunakan untuk menggerakan pesawat kearah kiri

2. Tombol speasi digunakan untuk menembak pesawat musuh Space

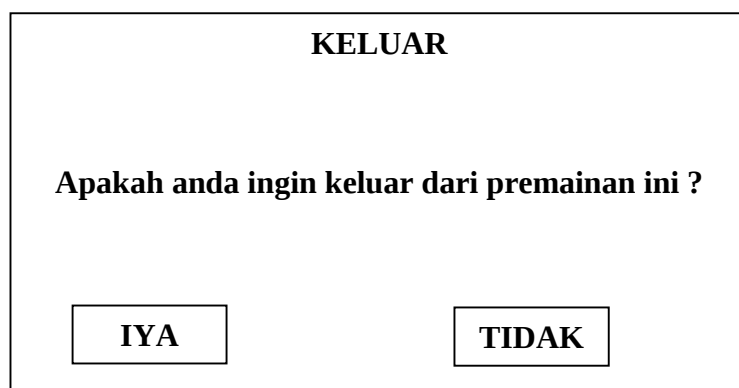
3. Pesawat pemain akan memiliki daya tahan, daya tahan/nyawa akan berkurang jika kita terkena pesawat musuh dan bertabrakan dengan pesawat musuh. BACK

4. Bila permainan berakhir pemain bisa melakukan permainan dari awal lagi

Gambar III.6. Tampilan Petunjuk Permainan

III.5.4. Rancangan Tampilan Menu Keluar

Pada tampilan menu *Keluar* merupakan menu yang harus kita perhatikan secara seksama bagaimana cara kita keluar dari game tersebut. Ketika kita membuka menu *Keluar* di tampilan tersebut akan memberi pilihan dengan kita tentang cara kita mau keluar dari game pesawat tersebut.

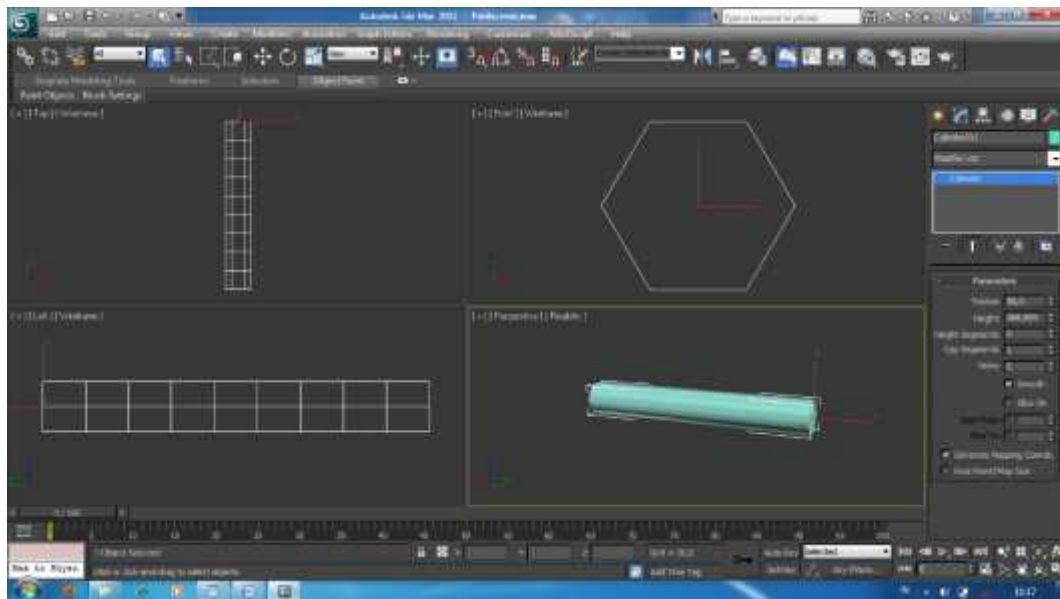


Gambar III.7 Tampilan Menu Keluar

III.5.5 Pembuatan Pesawat Pemain

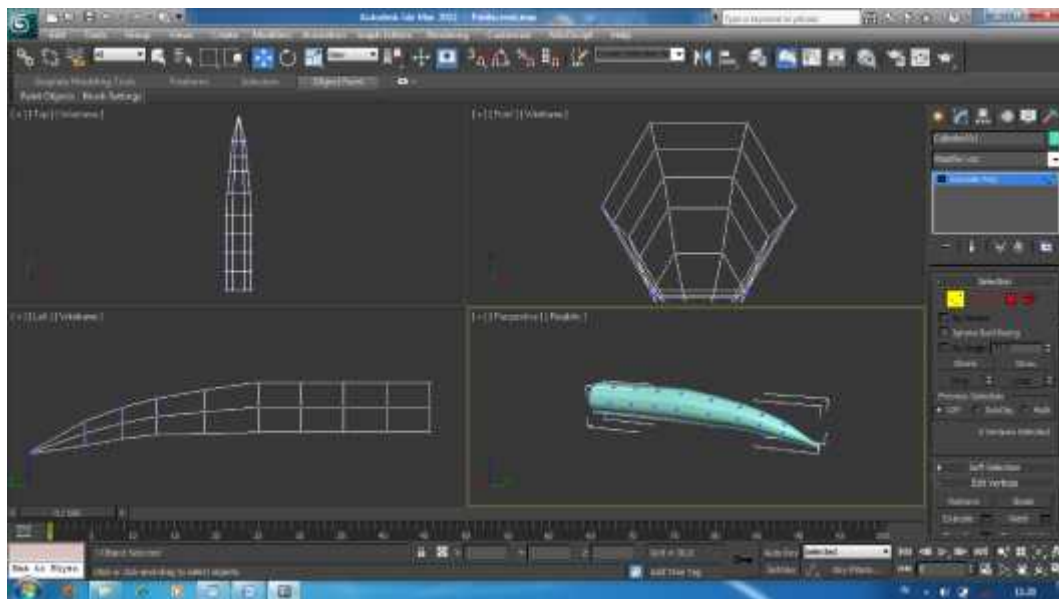
Adapun langkah-langkah dalam pembuatan Meteor adalah sebagai berikut :

1. Pada *Viewport Perspective* buat objek *Cylinder*. Caranya adalah klik *Create* → *Geometry* → *Standard Primitives* → *Cylinder* Kemudian isikan pada *Radius* = 20 dan *Height*= 270 *Sides*=5.
2. Convert *Cylinder* → menjadi Editable poly, Klik kanan → convert to *editable poly*. Pada action selection pilih *polygon* dan arahkan kursor ke ujung *Cylinder*.



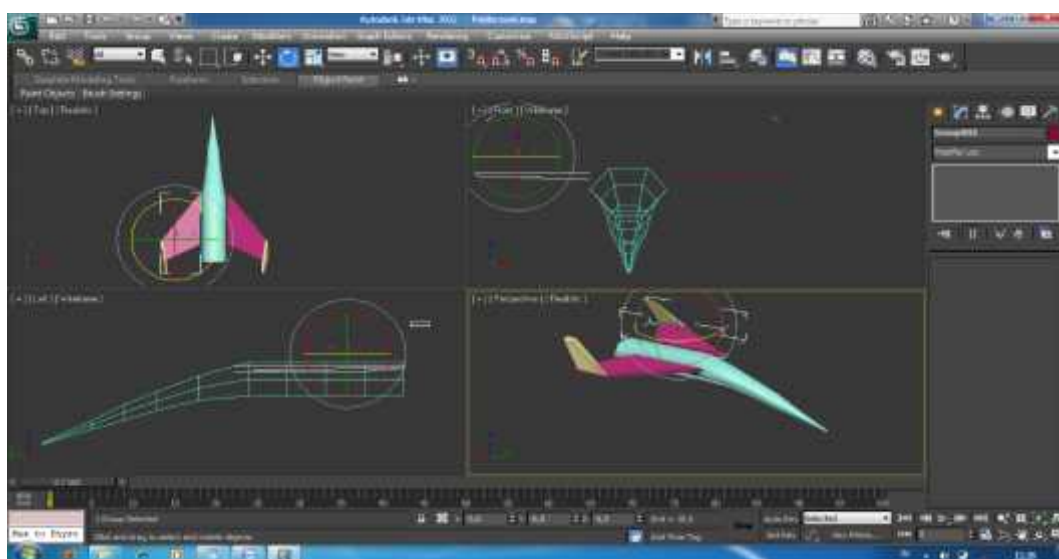
Gambar III.8. Tampilan Badan Pesawat

3. Dengan tombol *select and squash* buatlah ujung tabung tersebut meruncing sehingga kita akan mendapatkan bentuk kepala pesawat, Pilih *polygon* bagian atas dan move ke sumbu z untuk membentuk kokpit. Kemudian pilih selection menjadi bentuk vertex.
4. Pilih viewport dari TOP dengan menggeser masing masing vertex buatlah bentuk badan pesawat menjadi lebih nyata.
5. Setelah badan pesawat maka kita akan membuat *combution cumber* untuk *engine* pesawat dengan cara membuat *sylyder* dan rotasi ke sumbu y 90 derajat kemudian convert to *editable poly*. Pilih *polygon* bagian belakang dan lakukan teknik *Extrude* untuk mendapatkan bagian *exauce* yang kelihatan nyata. Setelah badan pesawat terbentuk maka kita tinggal membuat sayap pesawat dengan langkah sebagai berikut



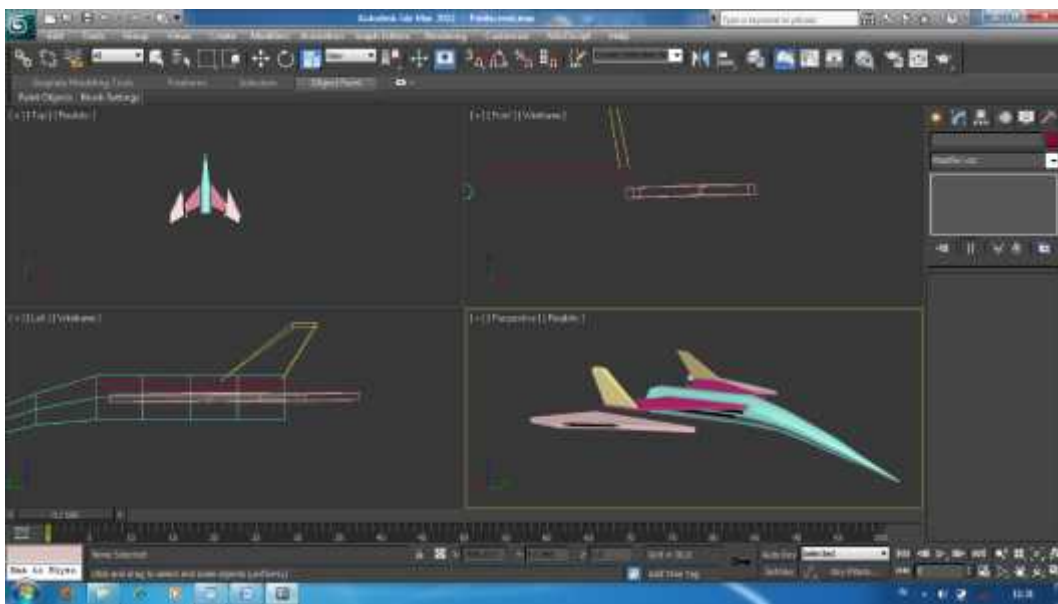
Gambar III.9. Tampilan Depan Bawan Pesawat

6. Buat sebuah box isikan *length* 60 *width* 50 dan *height* 2, Convert to editable poly pilih viewport top. Pilih selection menjadi vertex geser ujung ujung vertex sehingga box tadi akan menjadi bentuk sayap. Gabungkan badan pesawat dengan sayap.



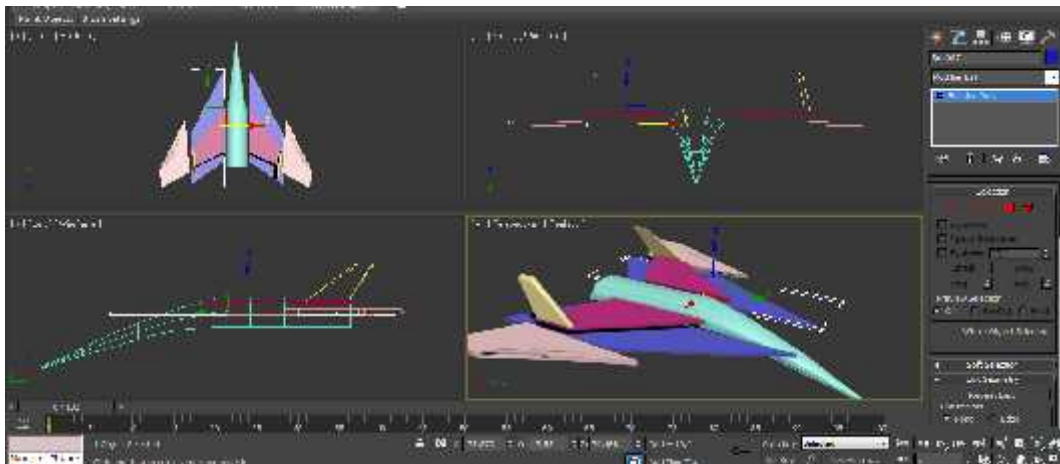
Gambar III.10. Tampilan Sayap Pertama Pesawat

7. Buat sebuah box dari Create > Geometry > Standard Primitives > Box. Klik dan drag pada viewport top. Klik kanan objek box tersebut kemudian pilih Convert To Editable Poly. Pilih submenu vertex dan gunakan Select and Move untuk mengubah bentuk objek box seperti sayap pesawat. Klik Mirror dan aktifkan Copy kemudian klik OK.



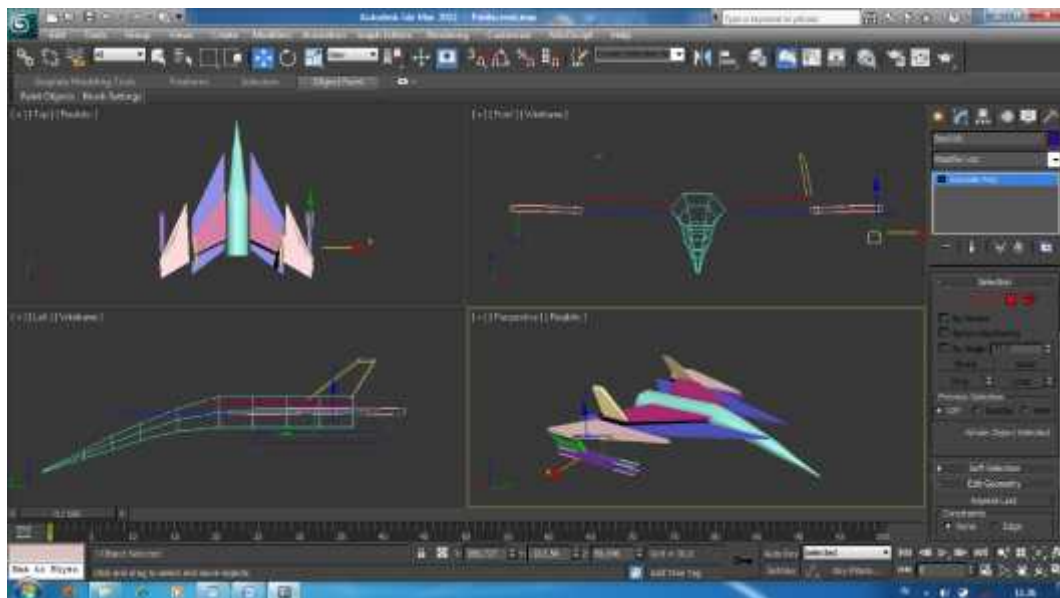
Gambar III.11. Tampilan Sayap Kedua Pesawat

8. Buat kembali objek box dari Create > Geometry > Standard Primitives > Box. Klik dan drag pada viewport top. Klik kanan objek box kemudian pilih Convert to Editable Poly. Aktifkan submenu vertex dan select and move untuk mengubah bentuk objek box seperti say bagian dalam.



Gambar III.12. Tampilan Sayap Dalam Pesawat

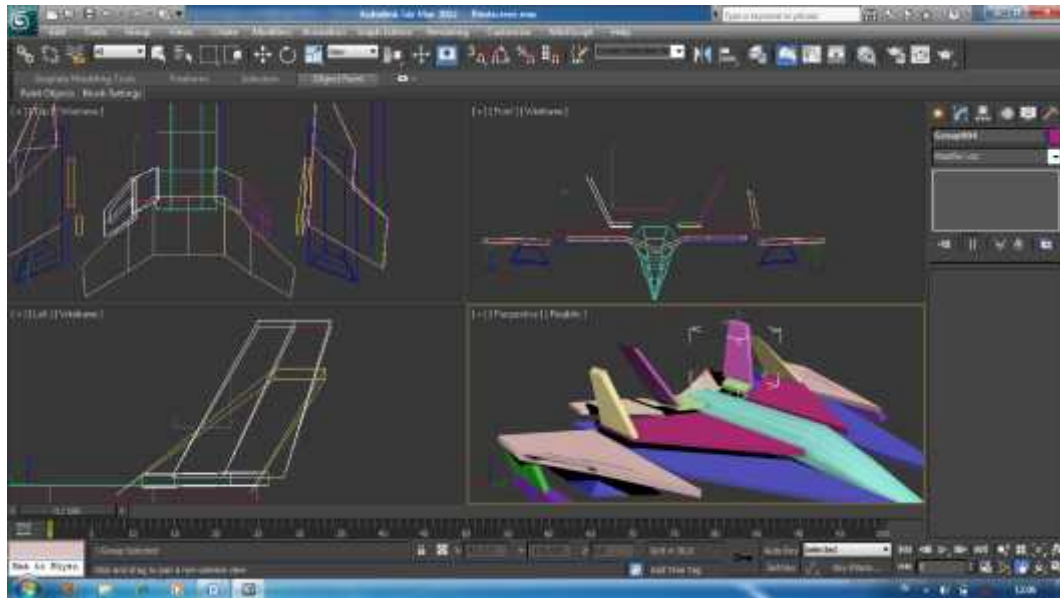
9. Buat kembali objek box. Klik dan drag pada viewport front. Klik kanan objek box dan pilih Convert to Editable Poly. Aktifkan submenu vertex dan select and move untuk membuat senjata pesawat.



Gambar III.13. Tampilan Senjata Pesawat

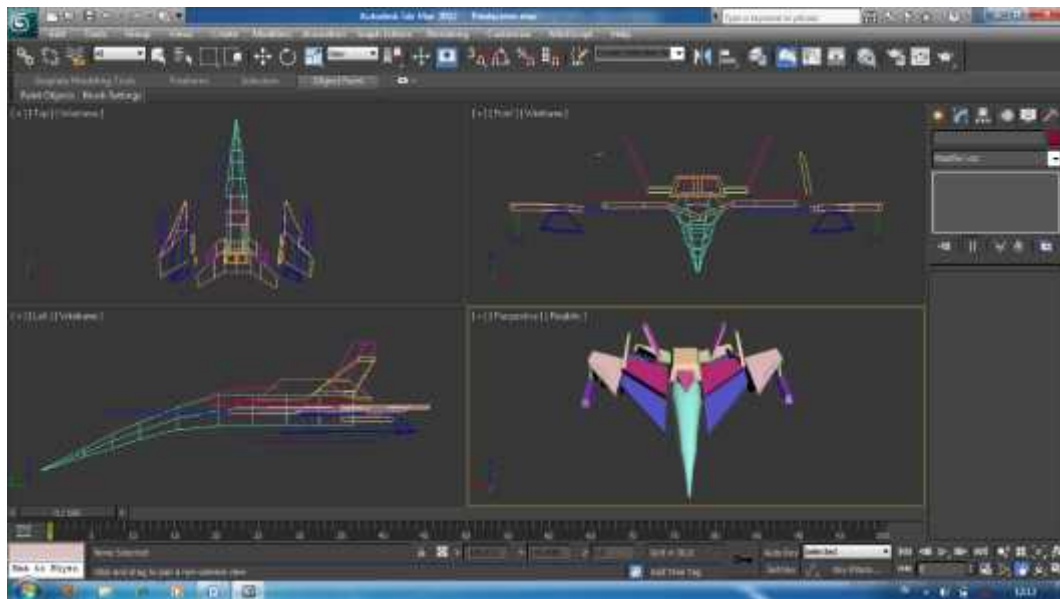
10. Buat objek box kembali dari Create > Geometry > Standard Primitives > Box. Klik dan drag pada viewport left. Klik kanan objek box kemudian pilih convert

to editable poly. Aktifkan submenu vertex dan select and move untuk mengubah bentuk objek box seperti pendorong pesawat.



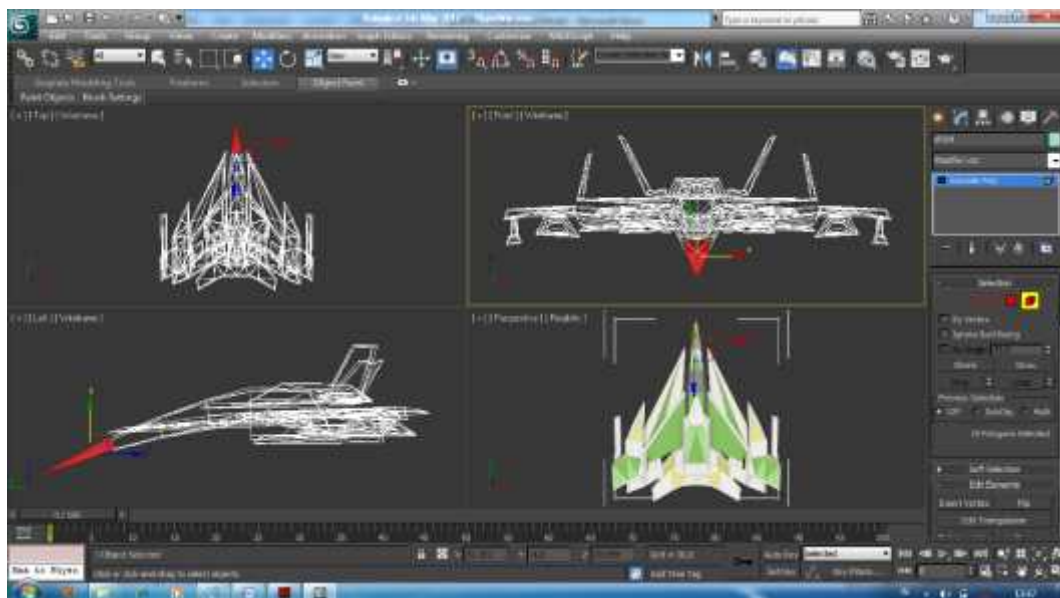
Gambar III.14. Tampilan Pendorong Pesawat

11. Klik dan drag pada viewport top. Klik kanan objek box dan pilih convert to editable poly. Aktifkan submenu vertex dan select and move untuk mengubah bentuk objek box menjadi kabin pesawat.



Gambar III.15. Tampilan Kabin Pesawat

12. Pada panel *Modifier List* pilih *Noise* atur *scalenya* 18, *strenght* $x=10$, *strenght* $y=10$, dan *strenght* $z=10$. Kemudian centang *Fractal* pada panel *noise* ,isi *Iterations* =8 .



Gambar III.16. Tampilan Pesawat Pemain