PENGEMBANGAN APLIKASI PERMAINAN PESAWAT UDARA BERBASIS WEB DENGAN MENGGUNAKAN JAVASCRIPT

Laporan Tugas Akhir

Diajukan Untuk Melengkapi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer

Oleh:

RIFKI ABDILLAH

01502-050

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS MERCU BUANA
JAKARTA
2009

LEMBAR PERNYATAAN

Yang bertanda tangan dibawah ini:

NIM	:	01502-050
Nama	:	RIFKI ABDILLAH
Judul Skripsi	:	PENGEMBANGAN APLIKASI PERMAINAN PESAWAT UDARA BERBASIS WEB DENGAN MENGGUNAKAN JAVASCRIPT

Menyatakan bahwa skripsi tersebut diatas adalah hasil karya saya sendiri dan bukan plagiat. Apabila ternyata ditemukan didalam laporan skripsi saya terdapat unsur plagiat, maka saya siap untuk mendapatkan sanksi akademik yang terkait dengan hal tersebut.

Jakarta, Juni 2009

(RIFKI ABDILLAH)

LEMBAR PERSETUJUAN

NIM : 01502-050

Nama : RIFKI ABDILLAH

Judul Skripsi : PENGEMBANGAN APLIKASI PERMAINAN PESAWAT UDARA BERBASIS WEB DENGAN MENGGUNAKAN JAVASCRIPT

SKRIPSI INI TELAH DIPERIKSA DAN DISETUJUI

JAKARTA, Juni 2009

Ir. Nixon Erzed, MT	**Sarwati Rahayu, ST. MMSI**
Pembimbing I	Pembimbing II

Devi Fitrianah, S.KOM., MTI	**Abdusy Syarif, ST., MT**
Koord Tugas Akhir Teknik Informatika	KaProdi Teknik Informatika

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Allah SWT yang telah memberikan hidayah dan rahmat-Nya sehingga penulis dapat menyelesaikan laporan Tugas Akhir yang berjudul ”Pengembangan Aplikasi Permainan Pesawat Udara Berbasis Web Dengan Menggunakan Javascript” dengan baik. Laporan Tugas Akhir ini ditulis untuk melengkapi persyaratan mencapai gelar sarjana strata satu (S1) Program Studi Teknik Informatika Fakultas Teknologi Industri Universitas Mercu Buana. Penulis berharap laporan Tugas Akhir ini dapat memperkaya wawasan, pengetahuan dan pemahaman tentang Aplikasi permainan pesawat udara berbasis web dengan menggunakan javascript.

Dalam menyelesaikan laporan Tugas Akhir ini, penulis banyak mendapat bantuan berupa dukungan, sumbangan pikiran dan bimbingan yang sangat besar artinya. Untuk itu dalam kesempatan ini penulis mengucapkan terima kasih kepada :

1. Bapak dan Ibu tercinta yang tanpa henti mengalirkan do’a untuk keselamatan dan keberhasilan penulis serta memberikan dukungan dan semangat baik spiritual, moril dan materil sehingga tugas ini dapat selesai pada waktunya.
2. Bapak Abdusy Syarif, ST., MT., selaku Ketua Program Studi Teknik Informatika Universitas Mercu Buana.
3. Bapak Ir. Nixon Erzed, MT, selaku Dosen Pembimbing I Tugas Akhir yang telah banyak membantu penyusun dalam membuat laporan tugas akhir ini sampai selesai.

4. Ibu Sarwati Rahayu, ST. MMSI, selaku Dosen Pembimbing II Tugas Akhir yang telah banyak membantu penyusun dalam membuat laporan tugas akhir ini sampai selesai.
5. Bapak Raka Yusuf, ST, selaku pembimbing akademik pada Program Studi Teknik Informatika Universitas Mercu Buana.
6. Untuk kakak-kakak ku yang telah memberikan bantuan, motivasi dan do'a selama penulisan tugas akhir ini.
7. Seluruh dosen Teknik Informatika yang telah mengajarkan ilmunya serta karyawan kampus yang telah melayani penulis selama kuliah di Universitas Mercu Buana.
8. Suseno atas bantuan Desain aplikasi dan waktunya.
9. Seluruh teman-teman angkatan 2002 terutama Jurusan Teknik Informatika khususnya idup sukandar, agung, Mario, udin, agus, doni, zaki, dede, mulyadi, taftazani, mamen yang telah memberikan dorongan spiritual dan material, dan seluruh pihak yang tidak dapat penulis sebutkan satu persatu terima kasih atas bantuan dan dukungannya selama pembuatan laporan Tugas Akhir ini.
10. Dan terakhir dari yang terakhir Mila nurmillah yang telah memberikan dorongan semangat, materi dan doa yang sangat besar dalam pembuatan laporan ini.

Semoga Allah SWT memberikan dan melimpahkan rahmat dan karunia-Nya atas segala bantuan yang telah diberikan kepada penulis.

Akhir kata penulis mengharapkan tulisan ini dapat memberikan manfaat bagi penulis khususnya dan pembaca pada umumnya. Penulis menyadari bahwa tulisan ini tidak lepas dari kekurangan. Atas saran dan kritik yang membangun penulis mengucapkan terima kasih.

Jakarta, Juni 2009

Rifki Abdillah

ABSTRACT

Gaming application or computer game has been developed in a rapid pace which created a large differentiation and transformed it from 2 – dimentioned game to 3 – dimentioned one. Among many gaming applications which exist, some of them have complicated flow thus make the user bored. Because of that background of study, the writer has created a simple gaming application (game) which an airplane characteristic which can be an alternative for a game lover and can be easily learned even by a beginner in computer programming.

The games application with an airplane characteristic was created by using Javascript Programming language and implemented to website. In analyzing the need of that application, the writer used a visual modeling tool called Unified Modeling Language (UML). The UML diagram which were used by the writer were the Use case diagram, Sequential diagram and Activity diagram.

Key word: Gaming application, airplane gaming application, game programming by Javascript.

ABSTRAK

Aplikasi permainan komputer berkembang dengan pesat yang mengakibatkan beraneka ragamnya aplikasi permainan tersebut dari permainan berjenis 2 dimensi hingga permainan 3 dimensi yang kompleks. Diantara aplikasi-aplikasi permainan yang telah ada, banyak yang rumit alur permainannya sehingga membuat bosan para penggunanya. Dengan latar belakang tersebut, penulis ingin mengembangkan sebuah aplikasi permainan pesawat udara yang dapat dijadikan alternatif bagi para penggemar aplikasi permainan komputer dan dapat dipelajari dengan mudah.

Pengembangan aplikasi permainan komputer dengan karakter pesawat udara tersebut menggunakan Bahasa Pemrograman Javascript dan diimplementasikan ke dalam Situs Web. Dalam melakukan analisis kebutuhan untuk aplikasi tersebut penulis menggunakan alat pemodelan visual Unified Modeling Language (UML). Diagram-diagram UML yang digunakan oleh penulis adalah diagram Use Case, diagram Sekuensial, dan diagram Aktivitas.

Kata kunci: pengembangan aplikasi permainan, permainan komputer pesawat udara, pemrograman *game* dengan Javascript.

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

DAFTAR KODE

BAB I
PENDAHULUAN

1.1 Latar Belakang

Penulis mengembangkan permainan ini untuk mengembangkan kreatifitas para pembuat program baik tingkat pemula maupun ahli dalam menciptakan beragam hiburan dalam bentuk fungsi. Arah dan gerak dalam objek dalam permainan ini terjadi karena adanya vektor, skalar, dan titik koordinat. Seiring dengan beraneka ragamnya permainan komputer saat ini, dari permainan komputer 2D hingga 3D yang kompleks dan tidak jarang pula permainan komputer yang sangat rumit alur permainannya membuat bosan para penggunanya, maka penulis akan membuat pengembangan pada sebuah aplikasi permainan komputer sederhana dengan karakter pesawat udara (Cupid) yang dapat dipelajari dengan mudah oleh pemula dibidang pemrograman komputer dan dapat dijadikan alternatif bagi para penggemar aplikasi permainan komputer. Salah satu alasan mengapa permainan komputer merupakan proyek pemrograman yang cukup bagus adalah pembuatan program aplikasi permainan membutuhkan individu yang sangat kreatif. Selain hal tersebut, membuat program aplikasi permainan memberikan wawasan keahlian pemrograman yang sangat luas bagi pemula maupun bagi yang sudah mahir. Diantara keahlian tersebut yaitu kemampuan merancang antarmuka permainan, kemampuan komputer grafis, membuat kecerdasan buatan, membuat simulasi fisik, menggubah musik, optimisasi kode sumber, dan lain sebagainya.

Pada kesempatan ini, penulis membahas tentang pengembangan sebuah aplikasi permainan komputer dua dimensi dengan karakter pesawat udara dengan

menggunakan bahasa skrip yaitu Javascript, dan akan diimplementasikan pada sebuah situs web.

1.2 Perumusan Masalah

Terciptanya sebuah pengembangan aplikasi permainan komputer dengan menggunakan bahasa pemrograman Javascript yang akan diimplementasikan pada sebuah situs web. Permainan ini dibuat dengan berbasiskan web dan dimainkan secara offline (tanpa terhubung dengan internet).

1.3 Tujuan Pembahasan

Penelitian ini bertujuan untuk mengembangkan sebuah permainan dengan memakai bahasa skrip yaitu Javascript, diharapkan dapat berguna untuk penggemar permainan sebagai langkah awal dalam membuat sebuah permainan dan untuk diimplementasikan pada sebuah situs web.

1.4 Ruang Lingkup dan Batasan Masalah

Permasalahan yang diangkat dalam penulisan Tugas Akhir ini adalah bagaimana cara mengembangkan sebuah aplikasi permainan komputer dengan karakter pesawat udara menggunakan bahasa pemrograman Javascript. Aplikasi tersebut nantinya akan diimplementasikan pada sebuah situs web. Proses awal program permainan pesawat udara tersebut yaitu dengan memasukan gambar latar belakang, gambar pesawat pemain, gambar pesawat musuh, gambar peluru (peluru pemain dan musuh), deteksi papan ketik (*keyboard*) terhadap karakter pesawat pemain, deteksi tubrukan, dan memasukkan suara (suara efek) ke dalam aplikasi permainan komputer tersebut

sehingga permainan tersebut dapat dimainkan dan dinikmati oleh pengguna komputer. Dalam aplikasi permainan tersebut terdapat dua tingkat permainan dan terdapat skor yang berfungsi untuk mengetahui berapa banyak jumlah musuh yang berhasil dihancurkan oleh pesawat pemain. Pengembangan aplikasi permainan yang akan dibangun hanya dapat dimainkan oleh satu orang pemain dan berjenis animasi permainan 2 Dimensi.

Batasan masalah pada penulisan tugas akhir ini adalah sebagai berikut:

1. Aplikasi permainan komputer dengan karakter pesawat udara dengan menggunakan perangkat lunak Javascript.
2. Pengembangan yang dibuat oleh penulis adalah dengan mengembangkan/menambahkan jenis musuh dan musik pengiring serta diimplementasikan ke dalam situs web yang dibangun dengan HTML (*Hypertext Markup Language*).
3. Pembahasan hanya akan difokuskan pada satu proses pengembangan aplikasi permainan, yaitu permainan pesawat udara. Meskipun pada situs web terdapat beberapa permainan lainnya, namun tidak akan dibahas karena hanya berfungsi sebagai aplikasi tambahan.
4. Di luar hal-hal yang telah disebutkan di atas tidak akan dibahas.

1.5 **Metodologi Rekayasa Perangkat Lunak**

Metodologi rekayasa perangkat lunak yang digunakan oleh penulis dalam laporan Tugas Akhir ini adalah dengan menggunakan model Waterfall. Model Waterfall memiliki beberapa tahapan yaitu:

1. Analisis

   Pada tahap ini penulis menentukan kebutuhan fungsional dari program permainan pesawat udara yang akan dikembangkan.

2. Perancangan

   Pada tahap ini penulis melakukan perancangan antarmuka dari aplikasi yang akan dibuat sesuai dengan kebutuhan hasil tahap analisis.

3. Implementasi

   Setelah melakukan analisa dan perancangan, penulis melakukan pengkodean aplikasi permainan pesawat udara sesuai kebutuhan aplikasi tersebut dengan menggunakan Javascript.

4. Pengujian

   Setelah pengembangan program permainan selesai, maka program permainan tersebut diuji untuk mengetahui sesuai tidaknya dengan kebutuhan pada tahap analisis.

5. Penggunaan dan Perawatan

   Dalam penulisan Tugas Akhir ini, penulis tidak menyertakan tahap penggunaan dan perawatan dikarenakan dalam pengembangan program aplikasi permainan pesawat udara ini hanya sampai pada tahap pengujian.

1.6 Sistematika Penulisan

Laporan Tugas Akhir ini disusun dengan sistematika terstruktur. Lima (5) bab yang disusun agar pembaca dapat memahami secara jelas pokok pembahasan penulis. Sistematika penulisannya adalah sebagai berikut:

BAB I : PENDAHULUAN

Bab ini menguraikan latar belakang penulisan, perumusan masalah, tujuan pembahasan, ruang lingkup, metodologi rekayasa perangkat lunak, dan sistematika penulisan.

BAB II : LANDASAN TEORI

Pada bab ini akan dijelaskan mengenai teori-teori dasar yang digunakan dalam merancang dan membangun program permainan komputer dalam penulisan Tugas Akhir ini.

BAB III : ANALISIS DAN PERANCANGAN

Bab ini menjelaskan mengenai analisis masalah dan proses perancangan dari aplikasi permainan pesawat udara yang akan dibangun sehingga siap untuk diimplementasikan.

BAB IV : IMPLEMENTASI DAN PENGUJIAN

Bab ini menjelaskan mengenai pengkodean, tampilan antarmuka permainan, dan pengujian aplikasi permainan yang telah selesai dibangun.

BAB V : PENUTUP

Bab ini merupakan bagian akhir dari penulisan Tugas Akhir ini yang berisi kesimpulan dan saran-saran yang mengarah kepada pengembangan aplikasi permainan yang telah dibangun.

BAB II

LANDASAN TEORI

2.1 Pengertian Permainan Komputer

Permainan komputer (*computer game*) atau disebut juga dengan permainan video (*video game*) adalah sebuah program yang ditulis dalam sebuah bahasa pemrograman, seperti C++ atau Visual Basic, yang mengijinkan seorang pemain berinteraksi dan bereaksi terhadap kejadian-kejadian di dalam sebuah dunia imajinasi (Harbour, 2002:16), sedangkan permainan komputer menurut Marner (2002) dari Departemen Ilmu Pengetahuan Komputer, Universitas Copenhagen, Denmark, mempunyai arti sebagai suatu potongan dari perangkat lunak interaktif dengan tujuan utama untuk melayani pemakai dengan memakai komputer. Dilihat dari sudut pandang pemrograman, sebuah permainan dapat diartikan sebagai metode pelatihan yang saling berhubungan (Brat, 2003).

Ciri utama suatu permainan adalah mempunyai tujuan akhir yang ingin dicapai oleh pemain, ada sejumlah aturan yang menentukan batasan tindakan yang bisa dilakukan pemain, dan tindakan pemain diluar batasan tersebut akan dianggap sebagai tindakan curang.

Permainan menjadi semakin banyak jenisnya dan semakin terkenal setiap tahun. Bentuk umum dari permainan adalah komputer pribadi, terutama untuk Microsoft Windows, dan beberapa untuk Macintos dan Linux. Bentuk yang lain adalah permainan

yang dibuat untuk menghibur dan terhubung dengan televisi. Permainan ini meliputi mesin seperti *sony playstation* 1 dan 2, dan Microsoft xbox. Mesin bertarung, permainan di telepon selular dan permainan berbasis web adalah contoh lain dari bentuk permainan. Permainan yang dibuat untuk menghibur tanpa terhubung dengan televisi meliputi mesin seperti Nintendo. Media televisi juga digunakan untuk permainan dengan menggunakan telepon.

Di dalam dunia komputer, permainan mempunyai beberapa elemen, yaitu grafis, antarmuka, suara, kecerdasan buatan, dan algoritma. Grafis merupakan tampilan permainan di layar alat elektronik, bisa berupa kumpulan piksel (satuan unit gambar terkecil) atau salah satu bagian gambar tersebut menunjukkan gerakan yang adalah hasil manipulasi si pemain.

Antarmuka adalah alat penghubung yang menghubungkan antara pemain dengan permainan, dimana pemain dapat memasukkan perintah ke dalam permainan melalui *keyboard*, *mouse*, *joystick* ataupun tampilan-tampilan tertentu. Dalam sebuah permainan, aktivitas pemain dapat bersifat diegetic maupun extradiagetic. Aktivitas yang bersifat diegetic adalah bagaimana karakter si pemain merubah aktivitas. Aktivitas yang bersifat extradiegetic adalah yang terjadi secara fisik pada diri pemain.

Suara merupakan fasilitas tambahan yang dapat membuat suatu permainan menjadi lebih menarik. Format suara yang digunakan dapat berupa mp3, wav, dan midi.

Kecerdasan buatan mempunyai suatu peranan penting di dalam kesuksesan atau kegagalan permainan komputer dari segi mutu dan kompleksitas. Kecerdasan buatan membuat kinerja atau penampilan suatu permainan menjadi lebih menarik. Penelitian dari Johnson dan Wiles (2001) dari Universitas Queensland membuktikan jika

kecerdasan buatan di dalam suatu permainan difokuskan pada tingkat dan teknik pembuatan suatu permainan oleh pengembang permainan maka permainan tersebut menjadi lebih sulit atau mudah pada tingkatan dan menjadi lebih mudah dalam penggunaan teknik – teknik pembuatan permainan.

Sebuah algoritma, yakni jantung hati dari program permainan yang merupakan serangkaian program berisi otak dan seperangkat aturan yang mendasari dinamika permainan tersebut.

Dalam pengembangan aplikasi permainan pesawat udara berbasis web sumber script yang digunakan oleh penulis terdapat di beberapa sumber buku tentang javascript dan dari internet.

2.2 Model Waterfall

Salah satu metodologi untuk merancang sistem-sistem perangkat lunak adalah model Waterfall. Pendekatan model Waterfall berupaya membangun suatu lingkungan yang sangat terstruktur dimana proses pengembangan dilaksanakan secara sekuensial. Model Waterfall terdiri dari dari beberapa tahap yaitu:

1. Analisis

   Tahap pengembangan rekayasa perangkat lunak dimulai dengan analisis yang sasaran utamanya adalah mengidentifikasikan apa yang harus mampu dilaksanakan oleh sistem yang diajukan.

2. Perancangan

   Apabila analisis menitikberatkan pada apa yang harus mampu dilakukan oleh sistem yang diajukan, perancangan menitikberatkan pada bagaimana sistem melakukan hal tersebut. Pada tahap inilah struktur sistem perangkat lunak dibangun.

3. Implementasi

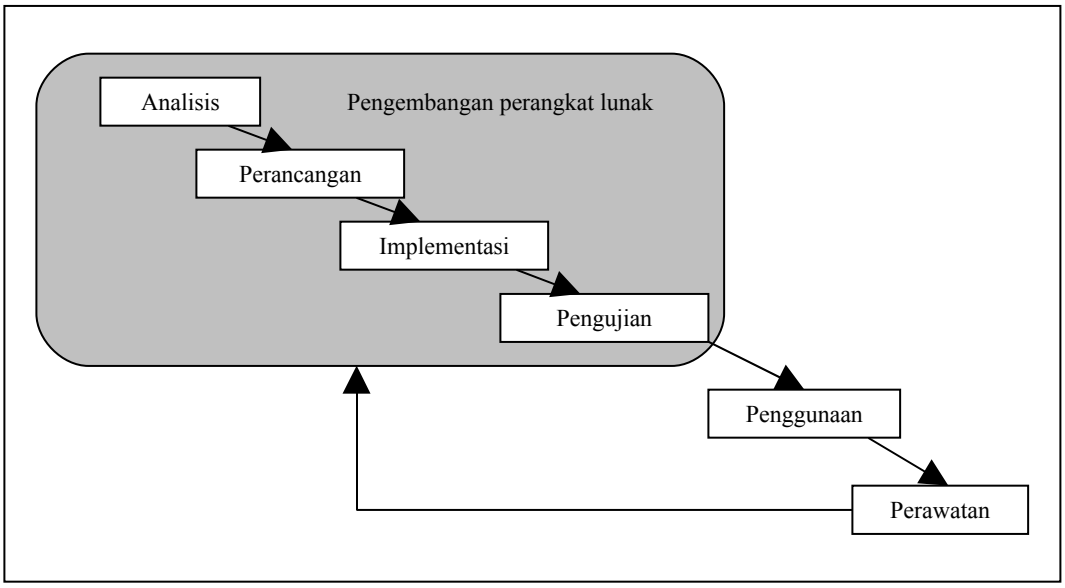
   Implementasi melibatkan penulisan aktual program-program, pembuatan berkas-berkas data, dan pengembangan basis data.

4. Pengujian

   Pengujian sangat terkait erat dengan implementasi, karena setiap modul sistem biasanya diuji ketika diimplementasikan.

5. Penggunaan dan modifikasi/perawatan

   Dalam penulisan Tugas Akhir ini, penulis tidak menyertakan tahap penggunaan dan perawatan karena pembuatan program permainan hanya sampai pada tahap pengujian. Gambar 2.1 memperlihatkan metodologi pengembangan rekayasa perangkat lunak model Waterfall (Brookshear, 2003:284).



Gambar 2.1. Model Waterfall

2.3 Unified Modelling Language (UML)

Menurut Munawar (2005:1), Unified Modelling Language (selanjutnya disebut UML) adalah bahasa pemodelan standar pada rekayasa perangkat lunak. UML adalah alat bantu yang sangat handal di dunia pengembangan sistem yang berorientasi objek. Hal ini disebabkan karena UML menyediakan bahasa pemodelan visual yang memungkinkan bagi pengembang sistem untuk membuat cetak biru atas visi mereka dalam bentuk yang baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi dan mengkomunikasikan rancangan mereka dengan yang lain.

Di proyek pengembangan sistem apapun, fokus utama dalam analisis dan perancangan adalah model dan diagram. Model menggambarkan pandangan yang lengkap tentang suatu sistem pada suatu tahapan tertentu dan dari perspektif tertentu. Sedangkan diagram menggambarkan atau mendokumentasikan beberapa aspek dari

sebuah sistem. Sebuah model mungkin mengandung satu atau lebih diagram. untuk model sederhana, satu diagram mungkin akan mencukupi. Akan tetapi biasanya sebuah model terdiri dari banyak diagram. Tabel 2.1 menampilkan jenis diagram resmi UML.

Tabel 2.1 Jenis diagram resmi UML versi 2

No.	Diagram	Kegunaan
1.	*Activity*	Perilaku prosedural dan paralel
2.	*Class*	*Class*, *fitur*, dan hubungan-hubungan
3.	*Communication*	Interaksi antar objek; penekanan pada jalur
4.	*Component*	Struktur dan koneksi dari komponen
5.	*Composite structure*	Dekomposisi *runtime* sebuah *class*
6.	*Deployment*	Penyebaran atau instalasi ke klien
7.	*Interaction overview*	Gabungan *sequence* dan *activity* diagram
8.	*Object*	Contoh konfigurasi dari contoh-contoh
9.	*Package*	Struktur hirarki saat kompilasi
10.	*Sequence*	Interaksi antar objek. Lebih menekankan pada urutan
11.	*State machine*	Bagaimana *even* mengubah objek selama aktif
12.	*Timing*	Interaksi antar objek; penekanan pada waktu
13.	*Use case*	Bagaimana pengguna berinteraksi dengan sebuah sistem

Pada penulisan tugas akhir ini, diagram UML yang akan dibahas adalah diagram *use case*, diagram aktifitas dan diagram sekuensial.

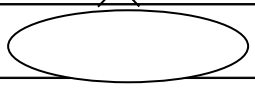
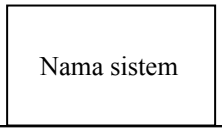
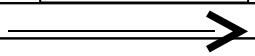
2.3.1 Diagram Use Case

Menurut Munawar (2005:63) *use case* adalah deskripsi fungsi dari sebuah sistem dari perspektif pengguna. *Use case* bekerja dengan cara mendeskripsikan tipikal interaksi antara pengguna sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai. Urutan langkah-langkah yang menerangkan antara pengguna dan sistem disebut skenario. Setiap skenario mendeskripsikan urutan kejadian. Setiap urutan diinisialisasi oleh orang, sistem yang lain, perangkat keras atau urutan waktu. Dengan demikian secara singkat bisa dikatakan *use case* adalah serangkaian skenario yang digabungkan bersama-sama oleh tujuan umum pengguna.

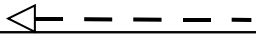
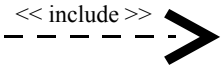
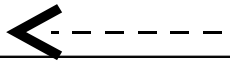
Dalam *use case*, pengguna biasanya disebut dengan aktor. Aktor adalah sebuah peran yang bisa dimainkan oleh pengguna dalam interaksinya dengan sistem. Aktor mewakili peran orang, sistem yang lain, atau alat ketika berkomunikasi dengan *use case*. Di dalam *use case* terdapat *stereotype* yaitu sebuah model khusus yang terbatas untuk kondisi tertentu. Untuk menunjukkan *stereotype* digunakan simbol ”<<” diawalnya dan ditutup ”>>” diakhirnya. <<include>> digunakan untuk menggambarkan bahwa suatu *use case* seluruhnya merupakan fungsionalitas dari *use case* lainnya. Biasanya <<include>> digunakan untuk menghindari penggandaan suatu *use case* karena sering dipakai. <<extend>> digunakan untuk menunjukkan bahwa suatu *use case* merupakan tambahan fungsional dari *use case* yang lain jika kondisi atau syarat tertentu dipenuhi.

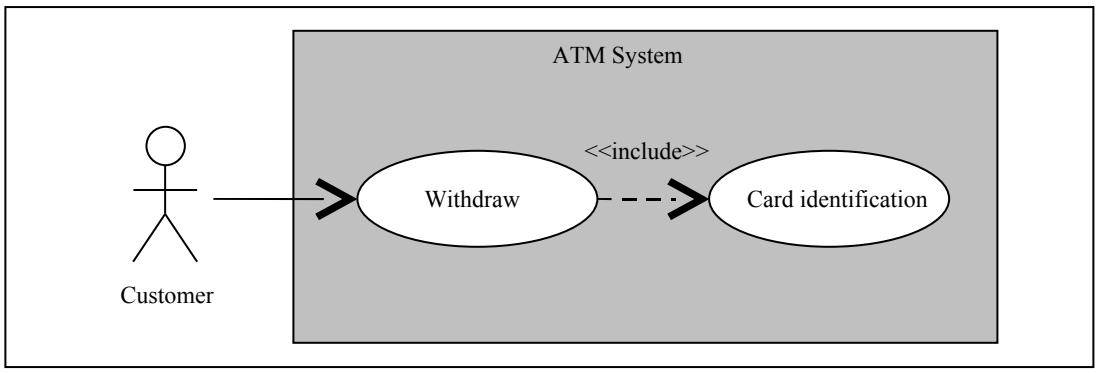
Di dalam Use Case juga terdapat generalisasi. Generalisasi diantara aktor adalah spesialisasi aktor yang bisa berpartisipasi di semua Use Case yang diasosiasikan dengan aktor yang lebih umum. Tabel 2.2 menampilkan notasi-notasi dalam pemodelan diagram Use Case menurut Booch dkk (1998:187).

Tabel 2.2. Notasi-notasi dalam pemodelan diagram Use Case menurut Booch dkk

No.	Notasi	Keterangan
1.		Aktor
2.		Use Case
3.		Batas sistem (*system boundary*)
4.		Garis penghubung (pengendali arah)

Tabel 2.2. Notasi-notasi dalam pemodelan diagram Use Case menurut Booch dkk (lanjutan)

5.		Gabungan (*association*)
6.		Generalisasi (*generalization*)
7.		Realisasi (*realization*)
8.		Stereotype penyertaan (*include*)
9.		Stereotype perluasan (*extend*)



Gambar 2.2. Contoh diagram Use Case sistem ATM
(www.sparxsystems.com/resources/uml2_tutorial/uml2_usecasediagram.html)

Menurut Munawar (2005:179) setiap *use case* harus dideskripsikan dalam dokumen yang disebut dengan dokumen aliran kejadian (*flow of event*). Dokumen ini mendefinisikan apa yang harus dilakukan oleh sistem ketika aktor mengaktifkan *use case*. Struktur dari dokumen *use case* ini bisa macam-macam, tetapi umumnya deskripsi ini paling tidak harus mengandung:

1. Deskripsi singkat (*brief description*).
2. Aktor yang terlibat.
3. Kondisi awal (*precondition*) yang penting bagi *use case* untuk memulai.
4. Deskripsi rinci dari aliran kejadian yang mencakup:

a. Aliran utama (*main flow*) dari kejadian yang bisa dirinci lagi.

b. Aliran bagian (*sub flow*) dari kejadian.

c. Aliran alternatif untuk mendefinisikan situasi perkecualian.

5. Kondisi akhir yang menjelaskan state dari sistem setelah *use case* berakhir.

Dokumen *use case* ini berkembang selama masa pengembangan. Di awal-awal penentuan kebutuhan sistem, hanya deskripsi singkat saja yang ditulis. Bagian-bagian lain dari dokumen ini ditulis secara gradual dan iteratif. Akhirnya sebuah dokumen lengkap bisa didapatkan di akhir fase spesifikasi. Biasanya pada fase spesifikasi ini sebuah prototipe yang dilengkapi dengan tampilan layar bisa ditambahkan. Pada tahap berikutnya, dokumen *use case* ini bisa digunakan untuk membuat dokumentasi untuk implementasi sistem.

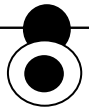
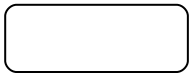
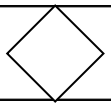
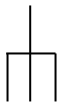
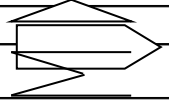
2.3.2 Diagram Aktivitas

Diagram Aktivitas adalah teknik untuk mendeskripsikan logika prosedural, proses bisnis, dan aliran kerja dalam banyak kasus. Diagram Aktivitas mempunyai peran seperti halnya bagan alir (Flowchart), akan tetapi perbedaannya dengan bagan alir adalah diagram Aktivitas dapat mendukung perilaku paralel sedangkan bagan alir tidak bisa (Munawar 2005:109).

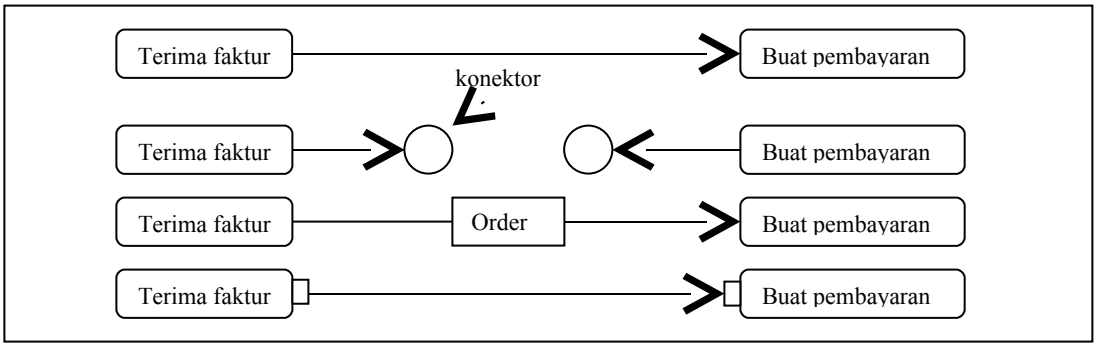
Berikut pada Tabel 2.3 adalah simbol-simbol yang sering digunakan pada saat pembuatan diagram aktifitas.

Tabel 2.3. Simbol-simbol yang sering digunakan pada diagram Aktivitas

No	Simbol	Keterangan

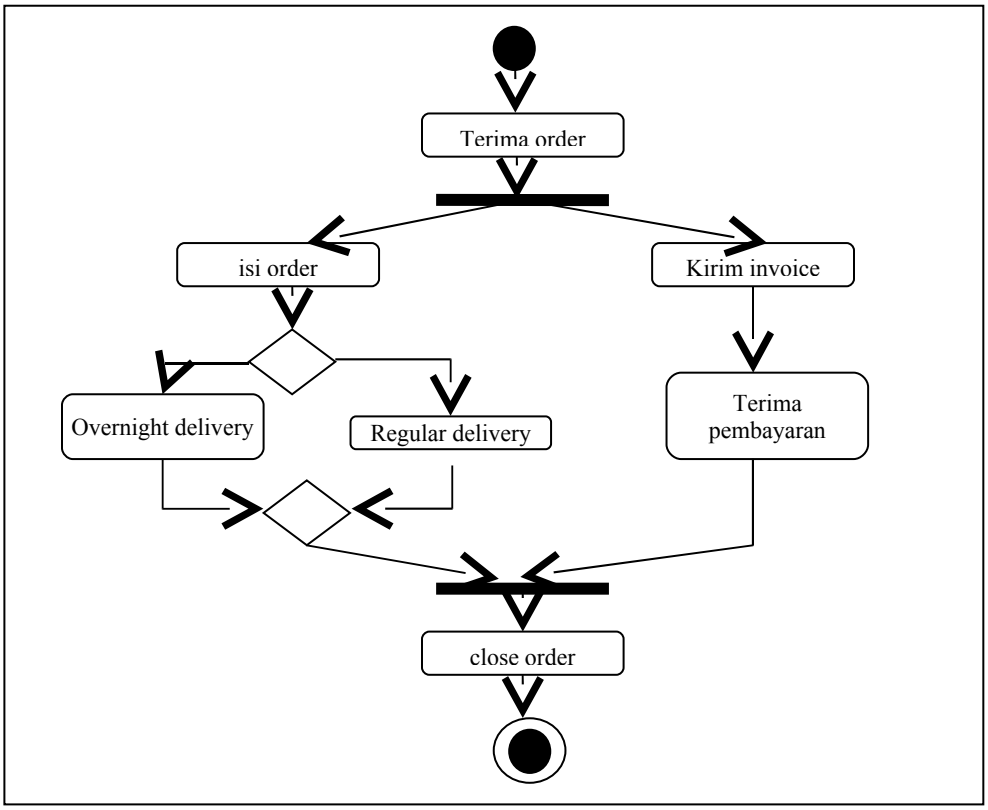
1.		Titik awal
2.		Titik akhir
3.		Aktivitas (*activity*)
4.		Pilihan untuk mengambil keputusan
5.		*Fork;* digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu
6.		*Rake;* menunjukkan adanya dekomposisi
7.		Tanda waktu
8.		Tanda pemgiriman
9.		Tanda penerimaan
10.		Aliran akhir (*Flow final*)

Untuk mendeskripsikan hubungan diantara dua aktivitas biasanya digunakan Flow & Edge. Ada empat cara yang bisa digunakan untuk menghubungkan dua aktivitas yang ditunjukkan oleh Gambar 2.3.



Gambar 2.3. Macam-macam Flow & Edge

Semua bentuk hubungan pada Gambar 2.3 adalah setara. Semua bisa digunakan sesuai dengan keperluan. Dalam banyak kasus cara yang pertama/paling atas adalah yang paling sering dipakai (Munawar, 2005:115).



Gambar 2.4. Contoh diagram Aktivitas terima order barang

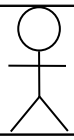
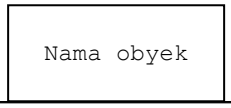
2.3.3 Diagram Sekuensial

Diagram sekuensial digunakan untuk menggambarkan perilaku pada sebuah skenario. Diagram ini menunjukkan sejumlah contoh obyek dan pesan (*message*) yang diletakkan diantara obyek-obyek ini di dalam Use Case (Munawar, 2005:87).

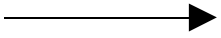
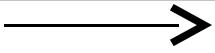
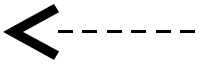
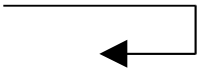
Komponen utama diagram sekuensial terdiri atas obyek atau disebut juga peserta (*participant*) yang dituliskan dengan kotak segi empat bernama, pesan diwakili oleh garis dengan tanda panah, dan waktu yang ditunjukkan dengan garis tegak lurus. Setiap

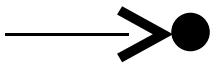
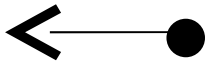
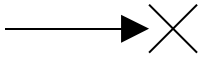
obyek terhubung dengan garis titik-titik yang disebut garis hidup (*lifeline*). Sepanjang garis hidup terdapat kotak yang disebut penggerakan (*activation*). Tabel 2.4 memperlihatkan notasi-notasi dalam pemodelan diagram Sekuensial. Contoh diagram Sekuensial ditunjukkan oleh Gambar 2.4.

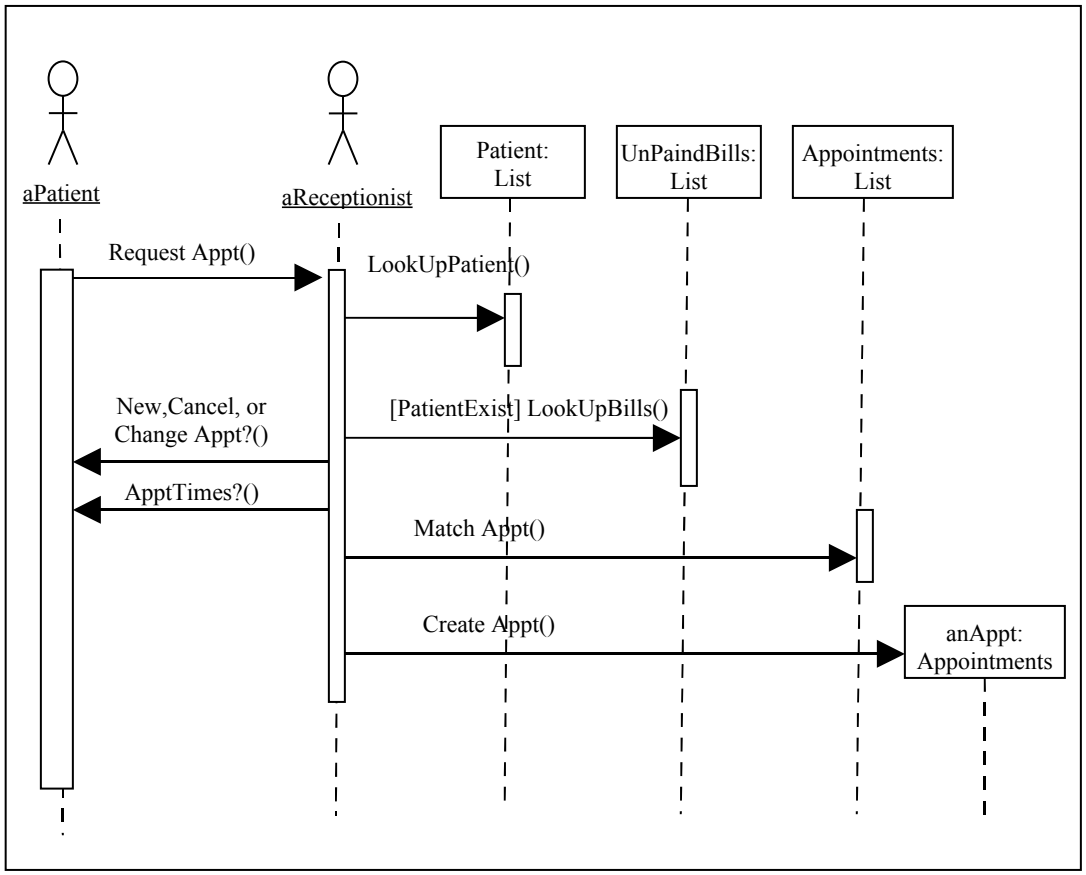
Tabel 2.4. Notasi-notasi dalam pemodelan diagram Sekuensial

No.	Notasi	Keterangan
1.		Aktor
2.		Obyek
3.		Batas (*boundary*)
4.		Kendali (*control*)
5.		Entitas (*entity*)
6.		Penggerakan (*activation*)

Tabel 2.4. Notasi-notasi dalam pemodelan diagram Sekuensial (lanjutan)

7.		Garis hidup (*lifeline*)
8.		Pesan selaras (*Synchronous message*)
9.		Pesan tidak selaras (*Asynchronous message*)
10.		Pesan kembali yang tidak selaras (*asynchronous return message*)
11.		Pesan rekursif (*self message*)

12.		Pesan hilang (*lost message*)
13.		Pesan ditemukan (*found message*)
14.		Pesan pembuatan obyek baru
15.		Pesan penghapusan obyek



Gambar 2.5. Contoh diagram Sekuensial (Dennis dkk, 2005:239)

2.4 Hypertext Markup Language

HTML dewasa ini dikenalkan sebagai bahasa standar untuk membuat membuat dokumen web. Sesungguhnya *Hypertext Markup Language (HTML)* justru tidak dibuat untuk mempublikasikan informasi di web, namun oleh karena kesederhanaan serta kemudahan penggunanya, HTML kemudian dipilih orang untuk mendistribusikan informasi di web.

Perintah – perintah HTML diletakkan dalam file berektensi *.html dan ditandai dengan mempergunakan tag (tanpa) berupa karakter “<”dan “>”. Tidak seperti bahasa

pemrograman berstruktur prosedural seperti Pascal atau C, HTML tidak mengenal *jumping* ataupun *looping*. Kode – kode HTML dibaca oleh browser dari atas ke bawah tanpa adanya lompatan – lompatan.

Struktur sebuah dokumen HTML pada dasarnya dibagi menjadi dua bagian besar, yaitu *header* dan *body*. Masing –masing ditandai oleh pasangan container tag <head> dan <body>. Bagian *head* berisikan judul dokumen dan informasi – informasi dasar lainnya, sedangkan bagian *body* adalah kata dokumennya.Pengaturan format texts dan pembentukan link dilakukan terhadap obyeknya langsung dengan ditandai oleh tag-tag HTML.

HTML diatur oleh Konsorsium WWW (W3C). Semua perubahan atas standar bahasa HTML harus disahkan terlebih dahulu oleh konsursium ini. Sejauh ini, HTML telah mengalami berbagai revisi sepanjang hidupnya. Standar paling akhir yang sekarang diperkenalkan adalah standar HTML 4.0, yang mendukung antara lain CSS (*cascading style sheet*), *dynamic content positioning* (penempatan isi secara dinamis), *downloadable font* (jenis font yang bisa didownload otomatis) dan sebagainya. Hingga kini, tidak semua *browser* web telah disesuaikan untuk mendukung standar HTML terbaru ini, sehingga banyak masalah inkompatibilitas antara macam – macam *bowser* web.

2.5 Pengantar JavaScript

Javascript adalah script sederhana yang berfungsi untuk menyusun aplikasi – aplikasi internet untuk *client* (dan *server*). Sepanjang sejarah internet, bahasa ini adalah bahasa skrip pertama untuk web. Bahasa ini adalah bahasa skrip untuk memberikan

kemampuan tambahan terhadap bahasa HTML dengan mengijinkan perintah di sisi user, yang artinya di sisi *browser* bukan di sisi *server web*. Jika javascript dijalankan di sisi *server web*, user tidak akan dapat menikmati kemampuan yang terdapat pada javascript karena jika web *server* rusak maka akan merusak juga data-data yang terdapat di dalamnya.

Javascript bergantung kepada *browser* (*navigator*) yang memanggil halaman web yang berisi skrip - skrip dari Javascript dan tentu saja terselip di dalam dokumen HTML. Javascript juga tidak memerlukan kompilator atau penterjemah khusus untuk menjalankannya (pada kenyataannya kompilator Javascript sendiri sudah termasuk di dalam *browser* tersebut). Lain halnya dengan bahasa "Java" (dengan mana JavaScript selalu dibandingkan) yang memerlukan kompilator khusus untuk menterjemahkannya di sisi user/klien.

Pada bahasa Javascript, tidak dimungkinkan penyembunyian kode skrip yang ditulis, karena kode langsung ditulis di dalam dokumen HTML dan sangat mudah terlihat, pada bahasa Java, kode sudah berbentuk setengah terkompilasi (dalam bentuk applet) dan tidak mungkin terlihat dalam dokumen HTML. Karena satu mesin virtual yang terdapat di sisi user bertanggung jawab untuk menterjemahkan program di dalam applet tersebut setiap kali halaman HTML yang memuat applet tersebut dipanggil oleh *browser*. Bila dibandingkan dengan applet java yang cukup lambat dibuka oleh *browser*, Javascript cukup cepat dipanggil oleh *navigator*.

Skrip dari JavaScript terletak di dalam dokumen HTML. Kode tersebut tidak akan terlihat dari dalam jendela *navigator*, karena di antara tag tertentu yang memerintahkan *navigator* untuk memperlakukan bahwa skrip tersebut adalah skrip dari JavaScript.

Contoh dari skrip yang menunjukkan bahwa skrip tersebut adalah skrip dari JavaScript adalah sebagai berikut :

```
<SCRIPT language="Javascript">
letakkan script disini
</SCRIPT>
```

4.1 Komentar Dan Skrip

Untuk menulis komentar di JavaScript, dapat menggunakan cara yang sama dengan aturan yang ada di bahasa C/C++ ataupun Java. Untuk menulis komentar dalam satu baris digunakan karakter dobel slash. // semua karakter di belakang // tidak akan di eksekusi. Untuk menulis komentar yang terdiri dari beberapa baris digunakan karakter /* dan */.

Pada contoh berikut ini adalah contoh skrip JavaScript di dalam suatu dokumen HTML, disini akan dibuat satu program untuk menampilkan satu kotak dialog pada saat dokumen HTML dibuka.

```
<HTML>
<HEAD>
<TITLE>Contoh Program Javascript</TITLE>
</HEAD>
<BODY>
<SCRIPT language="Javascript">
<!--
alert("Hallo !");
// -->
</SCRIPT>
</BODY>
</HTML>
```

2.5.2 Obyek

Javascript merupakan bahasa skrip yang berbasis obyek. Digunakan pada aplikasi klien dan *server*. Pembuatan obyek dengan Javascript tidak ditujukan untuk pengkapsulan, pewarisan, atau abstraksi seperti pada bahasa berorientasi obyek, tetapi lebih ditujukan untuk memberi kemampuan mengakses obyek *eksternal*, misalnya milik *browser* Netscape atau Internet Explorer. Javascript juga memungkinkan pendefinisian atribut dan servis sebuah obyek. Obyek adalah entitas program saat *run* yang memiliki atribut/*property* dan servis/*method.* Obyek yang memiliki atribut dan servis yang sama dikatakan memiliki kelas yang sama. Kelas adalah definisi statik obyek.

2.5.2.1 Obyek Standart Javascript

Obyek - obyek ini distandarisasikan oleh asosiasi ECMA (European Computer Manufacturer Association). Berikut ini adalah daftar obyek standart JavaScript

2.5.2.1.1 Obyek Array

Obyek array adalah satu obyek yang memungkinkan untuk membuat dan memanipulasi tabel, berikut ini adalah sintaks untuk membuat tabel :

```
var x = new Array(elemen1[, elemen2, ...]);
```

Jika tidak ada elemen yang disebutkan dalam parameter, tabel itu akan menjadi tabel kosong pada saat pembuatannya, sebaliknya jika elemen diisi, maka isi tabel akan di inisialisasi oleh nilai dari elemen tersebut. Sebagai tambahan obyek array mempunyai dua karakteristik properti yaitu properti *input* (masukan) dan *length* (panjang).

2.5.2.1.2 Obyek Boolean

vObyek boolean adalah obyek standart dari JavaScript yang memungkinkan untuk memanipulasi nilai nilai jenis boolean. Berikut ini adalah sintaks yang digunakan untuk membuat obyek booelan :

```
var x = new Boolean(parameter)
```

Parameter bisa berupa bisa berupa satu nilai (*True* atau *False*) atau bisa juga satu ekspresi, yang mana ekspressi akan di perhitungkan sebagai nilai boolean. Jika tidak ada nilai parameter yang dilewatkan atau nilai *0* atau string *kosong* atau *null* atau *undefined* atau juga *NaN (Not a Number)*, nilai boolean akan diinisialisaikan ke *False*. Sebaliknya obyek boolean akan mempunyai nilai *True* (Benar).

2.5.2.1.3 Obyek Tanggal (Date)

Obyek tanggal (date) memungkinkan untuk bekerja dengan semua variabel yang berhubungan dengan penanggalan dan juga manajemen waktu. Sintaks - sintaks untuk membuat obyek tanggal (*date)* adalah berikut ini :

- Nama_dari_obyek = new Date()

  sintaks ini memungkinkan untuk menyimpan tanggal dan jam saat ini.
- Nama_dari_obyek = new Date("hari, bulan tanggal tahun jam:menit:detik")

  parameter berbentuk string dengan batas batas pemisah sepeti format diatas.
- Nama_dari_obyek = new Date(tahun, bulan, hari)

  parameter adalah 3 integer yang dipisahkan oleh tanda koma
- Nama_dari_obyek = new Date(tahun, bulan, hari, jam, menit, detik[,perseribudetik])

Parameter adalah 6 integer yang dipisahkan oleh tanda koma JavaScript menyimpan tanggal dalam bentuk string yang berisi hari, bulan, tahun, jam, menit, dan detik. Meskipun demikian sangat sulit untuk bisa mengakses satu elemen waktu diatas dengan menggunakan obyek string, karena setiap elemen mempunyai ukuran yang berbeda - beda.. Sebaliknya obyek date memungkinkan untuk mengakses dan memodifikasi satu elemen tersebut.

2.5.2.1.4 Obyek String

Obyek string adalah satu obyek yang berisi beberapa metoda dan properti untuk memanipulasi data jenis string. Obyek string sendiri hanya mempunyai satu properti yaitu properti *length* untuk memperoleh panjang dari variabel data string. Sintaks dari properti ini adalah sebagai berikut :

```
x = nama_variabel_string.length;
x = ('sembarang teks').length;
```

metoda dari obyek string memungkinkan untuk memperoleh satu potongan / bagian dari data string dan juga memodifikasinya.

2.6 Konsep Variabel

Variable adalah suatu obyek yang berisi data data, yang mana dapat di modifikasi selama pengeksekusian program. Di JavaScript dapat memberikan nama variabel sepanjang yang disuka, akan tetapi harus memenuhi kriteria berikut ini :

- Nama variabel harus dimulai oleh satu huruf (huruf besar maupun huruf kecil) atau satu karakter ″_″.

- Nama variabel bisa terdiri dari huruf huruf, angka angka atau karakter _ dan & (spasi kosong tidak diperbolehkan).
- Nama variabel tidak boleh memakai nama nama berikut ini :
  - *Abstract*
  - *boolean break byte*
  - *case catch char class const continue*
  - *debugger default delete do double*
  - *else export extends*
  - *false final finally float for function*
  - *Goto*
  - *if, implements, import, in, infinity, instanceof, int, interface*
  - *label, long*
  - *native, new, null*
  - *package, private, protected, public*
  - *return*
  - *short, static, super, switch, synchronized*
  - *this, throw, throws, transient, true, try, typeof*
  - *var, void, volatile*
  - *while, with*

2.6.1 Deklarasi Variabel

Penulisan variabel JavaScript sangatlah fleksibel, dan tidaklah terlalu rumit dan ketat, sehingga tidaklah terlalu sering menerima pesan error pada saat menjalankan

program. Sebagai contoh deklarasi variabel di JavaScript dapat dilakukan dengan dua cara :

- eksplisit : dengan menuliskan kata kunci *var* kemudian diikuti dengan nama variabel dan nilai dari variabel :

  var test = “halo”
- implisit : dengan menuliskan secara langsung nama dari variabel dan diikuti nilai dari variabel :

  test = “halo”

Navigator secara otomatis akan memperlakukan pernyataan itu sebagai deklarasi dari sebuah variabel. Pada *navigator* versi lama mungkin terjadi kasus di mana *navigator* tidak mengenali pendeklarasian variabel secara implisit, disarankan untuk menggunakan cara eksplist dalam menulis program JavaScript. Berikut ini adalah contoh pendeklarasian variabel dengan kedua cara tersebut.

```
<SCRIPT language="Javascript">
<!--
var VariabelKu;
var VariabelKu2 = 3;
VariabelKu = 2;
document.write(VariabelKu*VariabelKu2);
// -->
</SCRIPT>
```

2.6.2 Letak Variabel

Berdasarkan tempat dimana didekalarasikan suatu variabel, variabel bisa diakses dari seluruh bagian program atau hanya di dalam bagian tertentu dari program. Pada saat

suatu variabel di deklarasikan tanpa menggunakan kata kunci *var*, atau bisa disebut dengan cara rumit, maka variabel itu bisa di akses dari seluruh bagian program(semua fungsi di dalam program dapat memanggil dan memakai variabel ini), dan disebut variabel ini sebagai variabel global. Sebaliknya jika dideklarasikan dengan cara mudah suatu variabel JavaScript (pendeklarasian variabel dengan menggunakan kata kunci *var*), kemungkinan pengaksesan variabel tersebut bergantung lokasi dimana variabel tersebut dideklarasikan.

Jika variabel tersebut dideklarasikan dibagian awal dari skrip program, yang artinya sebelum pendeklarasian semua fungsi, maka semua fungsi di dalam program bisa mengakses variabel ini, dan variabel ini menjadi variabel global.

Jika varibel tersebut dideklarasikan dengan menggunakan kata kunci *var* di dalam suatu fungsi tertentu, maka variabel itu hanya bisa di akses dari dalam fungsi tersebut, dan artinya variabel ini tidak berguna bagi fungsi fungsi yang lain, dan disebut variabel ini menjadi variabel lokal. Lihat contoh berikut ini :

```
<SCRIPT language="Javascript">
<!--
var a = 12;
var b = 4;
function PerkalianDengan2(b) {
var a = b * 2;
return }
document.write("Dua kali dari ",b," adalah ",PerkalianDengan2(b));
document.write("Nilai dari a adalah",a);
// -->
</SCRIPT>
```

Dari contoh diatas, variabel a dideklarasikan secara eksplisit di awal dari skrip program dan juga di deklarasikan di dalam fungsi. Berikut ini hasil dari program diatas.

Dua kali dari 4 adalah 8

Nilai dari a adalah 12

Berikut ini adalah contoh lain dimana variabel di deklarasikan secara implisit di dalam suatu fungsi :

```
<SCRIPT language="Javascript">
<!--
var a = 12;
var b = 4;
function PerkalianDengan2(b) {
a = b * 2;
return a;
}
document.write("Dua kali dari ",b," adalah ",PerkalianDengan2(b));
document.write("Nilai dari a adalah",a);
// -->
</SCRIPT>
```

Berikut ini hasil dari program diatas.

Dua kali dari 4 adalah 8

Nilai dari a adalah 8

Dari contoh diatas bisa dilihat pentingnya untuk menggunakan kata *var* pada saat membuat variabel baru.

2.6.3 Jenis - Jenis Data Dari Variabel

Di JavaScript, tidak perlu mendeklarasikan jenis variabel yang akan digunakan, sebaliknya di bahasa bahasa pemrograman yang lain (yang lebih advanced) seperti bahasa *C* atau *Java* harus mendeklarasikan secara detail apakah variabel yang digunakan

tersebut adalah merupakan suatu bilangan bulat (*int*), bilangan desimal (*float*), karakter (*char*), dan sebagainya.

Sebenarnya di JavaScript sendiri, hanya bisa memanipulasi 4 jenis data yaitu :

- Bilangan : bulat atau desimal, yang disebut sebagai *integer* atau *float*
- Kata (kumpulan huruf) : disebut *string*
- Boolean : suatu variabel yang mempunyai dua nilai dan berfungsi untuk memeriksa suatu kondisi :

  - *true* : jika kondisinya benar

  - *false* : jika kondisinya salah
- variabel dengan jenis *null* : satu kata khusus (termasuk keyword juga) untuk menjelaskan bahwa tidak ada data di dalamnya.

2.6.3.1 Integer

Integer atau bilangan bulat dapat ditampilan dalam beberapa basis berikut ini :

- *basis desimal* : integer di tuliskan dalam urutan unit bilangan (dari 0 sampai dengan 9), permulaan bilangan tidak boleh dimulai oleh angka *0*
- *basis heksadesimal* : dituliskan dalam urutan unit bilangan dari 0 sampai dengan 9 atau urutan huruf dari A sampai dengan F (atau a sampai dengan f), permulaan bilangan dimulai oleh *0x* atau *0X*
- *basis oktal* : dituliskan dalam urutan unit angka dari 0 sampai dengan 7, permulaan bilangan dimulai dengan angka *0*

2.6.3.2 Float

Float atau bilangan desimal bisa disebut juga sebagai bilangan pecahan atau bilangan yang bisa dituliskan dalam bentuk menggunakan tanda koma. Bilangan ini juga bisa di tuliskan dengan beberapa cara berikut:

- *Bilangan bulat desimal* : 895
- *Bilangan dengan tanda koma* : 895,12
- *Bilangan pembagian* : 27/11
- *Bilangan eksponensial* : bilangan dengan tanda koma , kemudian diikuti oleh huruf e(atau E), kemudian diikuti oleh bilangan bulat yang artinya pangkat dari bilangan 10 (+ atau -, pangkat postitif atau negatif), contoh :

  var a = 2.75e-2;

  var b = 35.8E+10;

  var c = .25e-2;

2.6.3.3 String

String adalah kumpulan dari karakter, dideklarasikan variabel string menggunakan tanda (') atau ("), kedua tanda tersebut harus digunakan secara berpasangan dan tidak bisa digunakan secara sendiri sendiri atau bercampur. Berikut ini adalah beberapa cara untuk mendeklarasikan variabel string :

var a = "Hallo";

var b = 'Sampai Ketemu Lagi !';

Ada beberapa karakter spesial yang bisa digunakan untuk mensimulasikan bagian dari karakter yang tidak terlihat (non visual) dan juga untuk menghindarkan

kemungkinan *navigator* "mengalami kebingungan" dalam membedakan antara string dan skripnya sendiri, karakter spesial ini menggunakan simbol antislash (\), beberapa contoh karakter spesial tersebut

- \n : kembali ke baris awal
- \r : menekan tombol ENTER
- \t : tab
- \" : tanda petik ganda
- \' : tanda petik tunggal
- \\ : karakter antislash

2.6.3.4 Boolean

Boolean adalah satu variabel khusus yang berguna untuk mengevaluasi suatu kondisi tertentu, oleh karenanya boolean mempunyai dua nilai :

- *True* : diwakili oleh nilai 1
- *False* : diwakili oleh nilai 0

2.6.4 Konversi jenis variabel

Meskipun JavaScript memungkinkan pengaturan perubahan jenis variabel secara transparan, kadang - kadang perlu juga untuk melakukan konversi jenis variabel secara paksa. Ada 2 fungsi dasar yang memungkinkan merubah jenis variabel yang dilewatkan dengan parameter tertentu :

- ParseInt()

Fungsi ini mungkinkan merubah satu variabel yang dilewatkan dengan parameter tertentu (bisa dalam bentuk string ataupun dalam bentuk bilangan dalam basis yang disebutkan di parameter kedua) menjadi bilangan bulat. Sintaksnya adalah sebagai berikut :

parseInt(string[, basis]);

Agar supaya fungsi parseInt() mengembalikan nilai bilangan bulat, maka parameter yang dilewatkan harus dimulai dengan karakter bilangan [0-9], prefiks hexadesimal 0x, dan / atau karakter +,-,e,dan E. Selain daripada itu fungsi *parseInt()* akan mengembalikan nilai *NaN (Not a Number).* Jika karakter berikutnya tidak valid, maka akan diabaikan oleh fungsi *parseInt(),* dan akan ditampilkan terpotong jika di bagian depan karakter valid dan bagian belakang karakter tidak valid. Berikut ini salah satu contoh penggunaan fungsi parseInt()

```
var a = "123";
var b = "456";
document.write(a+b,"<BR>"); // hasil 123456
document.write(parseInt(a)+parseInt(b),"<BR>"); // hasil 579
```

- ParseFloat()

Adalah satu fungsi inti dari JavaScript yang memungkinkan merubah variabel yang dilewatkan dengan parameter tertentu menjadi bilangan desimal, Sintaks dari fungsi parseFloat adalah sebagai berikut :

parseFloat(string);

BAB III

ANALISIS DAN PERANCANGAN

3.1. Analisis Masalah

Tahap awal pembangunan sebuah perangkat lunak adalah tahap analisis. Untuk menganalisa masalah pada aplikasi permainan yang akan dikembangkan, diperlukan skenario dari permainan tersebut. Dalam mengembangkan aplikasi permainan komputer dengan karakter pesawat udara (Cupid) yang diimplementasikan pada situs web ini. Berikut ini akan dipaparkan skenario dari permainan pesawat udara (Cupid) yang akan dikembangkan.

3.1.1 Skenario Permainan

Hal pertama yang dilakukan penulis dalam pengembangan permainan adalah menambahkan jenis musuh dan memasukkan musik pengiring ke permainan Cupid yang akan diimplementasikan pada situs web. Setelah itu pemain menekan teks yang mengandung link yang menghubungkan situs web tersebut ke halaman permainan pesawat udara. Cupid merupakan permainan pesawat udara berbasis web. Dalam permainan cupid ini terdapat beberapa peraturan.

Peraturan yang pertama adalah mengenai pergerakan. Posisi, arah, gerak dan peluru yang berasal dari pemain dan musuh terdiri atas perpaduan antara titik-titik koordinat. Pemain harus dapat menghindari peluru yang berasal dari musuh. Selain itu, pemain juga harus dapat menghindari persentuhan dengan musuh dan tanah sehingga pemain dapat bertahan di dalam permainan

Pemain harus dapat menghancurkan musuh–musuh yang berada di dalam permainan dengan peluru yang ditembakkan oleh pemain. Pemain memiliki 2 jenis peluru. Jenis yang pertama adalah peluru biasa yang mempunyai arah dan gerak maju secara horizontal. Tujuan dari penggunaan peluru biasa ini adalah pemain dapat menghancurkan musuh yang berada tepat di depan pemain. Jenis peluru yang kedua adalah peluru kendali. Arah dari peluru kendali adalah dari atas menuju ke bawah dan bergerak mendekati musuh-musuh yang posisinya berada di bawah pemain. Peluru ini digunakan untuk menghancurkan musuh yang posisinya berada di bawah pemain.

Peraturan yang kedua adalah mengenai nilai . Nilai awal permainan dimulai adalah nol (0). Apabila pesawat pemain dapat menghancurkan tiap-tiap pesawat musuh yang muncul di layar permainan, maka pemain mendapat tambahan nilai sepuluh (10) sedangkan jika pemain dapat menghanurkan tiap-tiap musuh dengan peluru kendali baik yang di atas maupun yang di bawah akan mendapatkan nilai lima puluh (50). Nilai yang telah didapat tidak disimpan ke suatu basis data karena aplikasi permainan tersebut tidak menggunakan basis data.

Jika pemain berhasil menghancurkan musuh. Nilai yang diperoleh oleh pemain akan bertambah. Nilai yang diperoleh oleh pemain pada akhir permainan akan ditempatkan di kotak nilai tertinggi. Jika pada permainan kedua, pemain memperoleh nilai yang lebih tinggi dari permainan pertama, maka nilai pada permainan pertama yang terdapat pada kotak nilai tertinggi akan digantikan dengan nilai dari permainan kedua. Jika nilai yang didapat pada permainan berikutnya tidak lebih tinggi dari nilai permainan sebelumnya, maka nilai pada permainan sebelumnya

akan tetap berada di kotak nilai tertinggi sampai pemain berhasil melewati nilai yang diperoleh pada permainan sebelumnya.

Peraturan yang ketiga adalah mengenai nyawa. Di dalam permainan ini, pemain hanya mempunyai satu nyawa dalam menyelesaikan tugasnya. Jika pemain terkena tembakan, atau bersentuhan dengan musuh dan tanah, maka pemain akan dianggap kalah.

Peraturan yang Keempat adalah mengenai tingkat kesulitan. Permainan ini mempunyai dua tingkatan kesulitan, yaitu tingkat kesulitan mudah dan normal. Di dalam tingkat kesulitan mudah, musuh akan mengurangi banyaknya tembakan yang dilancarkan sebanyak satu (1) kali dari jumlah tembakan yang terdapat pada tingkat kesulitan normal. Pada tingkat kesulitan normal musuh dapat menembakan peluru sebanyak tiga (3) kali.

Ketika permainan mendekati akhir dari suatu tingkat (*level*) permainan tertentu, maka satu musuh yang muncul paling akhir akan tetap ditampilkan di layar permainan dan musuh lainnya tidak akan berdatangan lagi. Musuh tersebut dinamakan dengan Boss/musuh utama. Akan tetapi Boss tidak berbentuk pesawat melainkan berbentuk mobil tank yang dapat bergerak maju dan mundur. Boss cukup sulit untuk ditembak jatuh oleh peluru pemain karena kekuatan Boss lebih besar dari kekuatan musuh sebelumnya. Apabila pesawat pemain bertabrakan dengan Boss dan atau terkena peluru Boss, maka pemain akan kalah. Pesawat pemain akan terus muncul di layar permainan selama permainan belum berakhir.

Permainan berakhir apabila Boss telah dihancurkan. Apabila Boss telah dapat dihancurkan oleh pesawat pemain, maka skor keseluruhan akan muncul dan pesawat

pemain dapat melanjutkan ke tingkat (*level*) permainan berikutnya dengan catatan apabila tingkat permainan berikutnya tersebut masih tersedia.

3.1.2 Spesifikasi Kebutuhan Sistem

Untuk aplikasi permainan yang sesuai dengan skenario permainan yang telah diuraikan di atas dapat diketahui kebutuhan sistem dari aplikasi permainan yang akan dibangun yaitu sebagai berikut:

1. Elemen permainan, terdiri dari:
    a. Pesawat (pesawat pemain dan pesawat musuh)
    b. Peluru (peluru pesawat pemain dan peluru pesawat musuh)
    c. Ledakan
    d. Suara efek
2. Proses Perhitungan Nilai dan Kehancuran Pesawat, yaitu sebagai berikut:
    a. Nilai dapat diperoleh apabila pesawat pemain dapat menghancurkan tiap-tiap pesawat musuh yang muncul di layar permainan.
    b. Pesawat pemain dapat hancur yang disebabkan oleh tertembaknya pesawat pemain oleh pesawat musuh atau bertabrakan dengan pesawat musuh dan tanah.
    c. Pesawat musuh dapat hancur apabila musuh tersebut tertembak oleh pesawat pemain dan atau bertabrakan dengan pesawat pemain.
3. Aplikasi permainan dengan karakter pesawat tempur tersebut menerima respon dari pengguna melalui papan ketik (Keyboard) untuk memulai permainan dan mengontrol pesawat pemain.

3.1.3 Parameter Keberhasilan

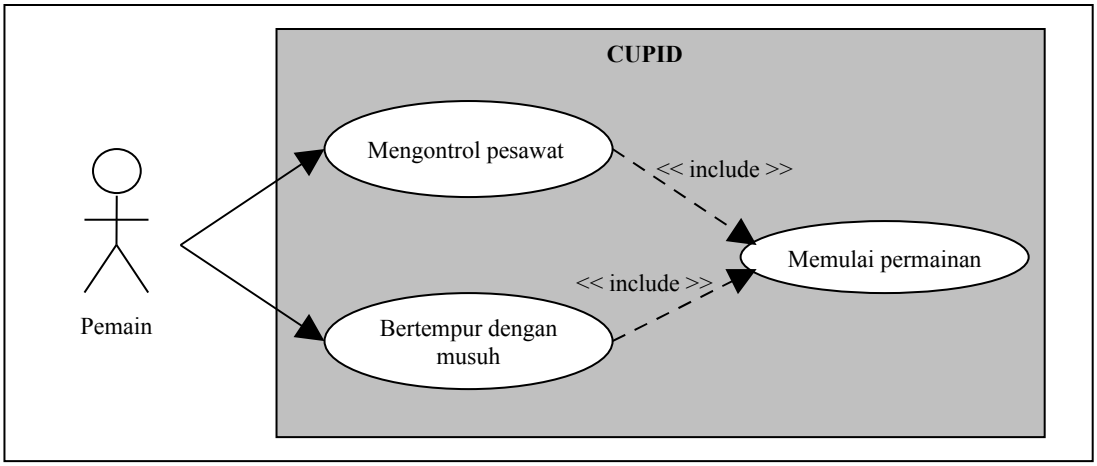
Aplikasi permainan dengan karakter pesawat udara (cupid) ini dinyatakan berhasil apabila sebagai berikut:

1. Pemain dapat mengendalikan pesawat (pesawat pemain).
2. Pesawat pemain dapat mengeluarkan peluru.
3. Pesawat musuh dapat mengeluarkan peluru.
4. Muncul ledakan akibat dari proses tubrukan, yaitu apabila:
    a. Pesawat musuh terkena peluru pesawat pemain atau sebaliknya pesawat pemain terkena peluru pesawat musuh.
    b. Pesawat pemain bertabrakan dengan pesawat musuh dan tanah.
5. musik pengiring dapat berfungsi.
6. Permainan pesawat tersebut dapat diimplementasikan ke dalam situs web.

3.2 Diagram Use Case

Diagram Use Case sangat membantu menganalisa kebutuhan-kebutuhan dari permainan yang akan dibangun. Karena diagram Use Case memperlihatkan apa yang dilakukan oleh pengguna terhadap sistem (dalam hal ini permainan komputer dengan karakter pesawat udara yang akan dibangun).

Secara umum diagram *use case* pada aplikasi permainan pesawat udara yang dibangun dapat dilihat pada Gambar 3.1.



Gambar 3.1. Diagram Use Case Permainan Pesawat Udara (Cupid)

Dari Gambar 3.1 dapat dilihat bahwa terdapat hanya satu aktor yang berhubungan dengan sistem yaitu aktor pemain dan terdapat dua Use Case yaitu Use Case mengontrol pesawat (Cupid) dan Use Case bertempur dengan musuh serta terdapat Stereotype <<include>> memulai permainan untuk semua Use Case.

Dokumentasi dari Use Case-Use Case tersebut di atas ditunjukkan oleh Tabel 3.1, Tabel 3.2, dan Tabel 3.3.

Tabel 3.1. Spesifikasi naratif untuk Use Case memulai permainan (bagian 1)

Nama Use Case	Memulai Permainan
Deskripsi Singkat	*Use case* ini memungkinkan pemain untuk memulai permainan agar dapat melakukan aktifitas yang ada di dalam permainan pesawat tersebut, yaitu mengendalikan pesawat dan bertempur dengan musuh.
Aktor	Pemain
Pra Kondisi	Permainan belum dimulai

Tabel 3.1. Spesifikasi naratif untuk Use Case memulai permainan (lanjutan)

Tindakan Utama	1. Pemain menekan teks yang mengandung link yang menghubungkan situs web ke halaman permainan. 2. Sistem menampilkan halaman permainan. 3. Pemain memilih tingkat kesulitan permainan 4. Pemain menekan tombol START atau S 5. Sistem menampilkan layar permainan sehingga pemain dapat memulai permainan
Tindakan alternatif	-
Pasca Kondisi	Permainan di mulai

Tabel 3.1. Spesifikasi naratif untuk Use Case memulai permainan (bagian 2)

Nama Use Case	Memulai permainan
Deskripsi Singkat	Use Case ini memungkinkan pemain memulai permainan agar dapat mengontrol pesawat dan bertempur dengan musuh.
Aktor	Pemain
	Aplikasi permainan tersebut telah dipasang (*installed*), dijalankan, dan siap untuk dimainkan.
Tindakan Utama	1. Pemain menekan tombol S untuk memulai permainan.
Tindakan alternatif	-
Pasca Kondisi	Jika Use Case berhasil dijalankan maka permainan dimulai dan pemain dibawa ke suasana pertempuran.

Tabel 3.2. Spesifikasi naratif untuk Use Case mengendalikan pesawat pemain

Nama Use Case	Mengontrol Pesawat pemain
Deskripsi Singkat	Use Case ini memungkinkan pemain mengendalikan pesawat Wing melalui penekanan tombol J, L, K, I atau angka 2, 6, 4, 8 pada papan ketik. Sehingga pesawat dapat bergerak di dalam layar permainan.
Aktor	Pemain
Pra Kondisi	Permainan dimulai
Tindakan Utama	1. Pemain menekan tombol J atau agka 2 pada papan ketik untuk menggerakkan pesawat ke arah kiri 2. Pemain menekan tombol L atau angka 6 pada papan ketik untuk menggerakkan pesawat ke arah kanan 3. Pemain menekan tombol K atau angka 4 pada papan ketik untuk menggerakkan pesawat ke arah bawah 4. Pemain menekan tombol I atau angka 8 pada papan ketik untuk menggerakkan pesawat ke arah atas
Tindakan alternatif	-
Pasca Kondisi	Jika Use Case sukses dijalankan maka, pesawat dapat bergerak ke atas, bawah, kiri, dan kanan pada layar permainan.

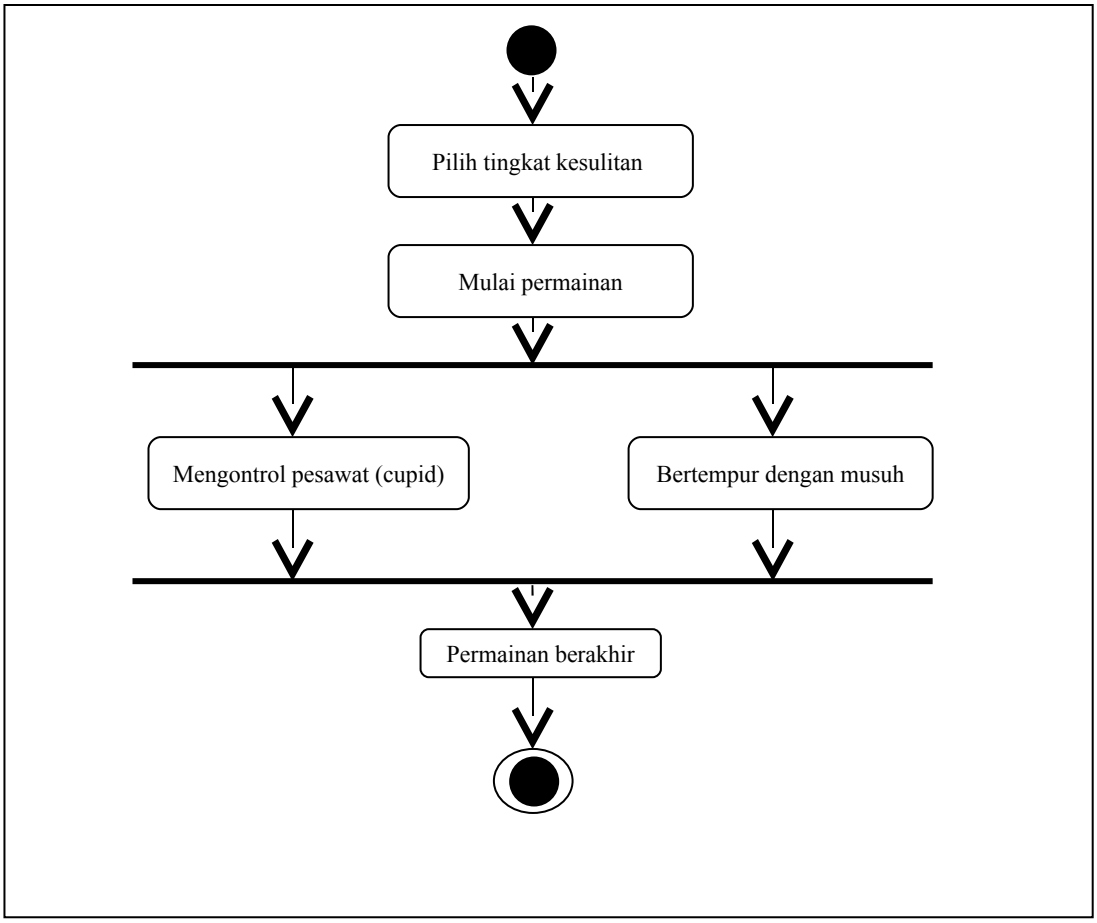
Tabel 3.3. Spesifikasi naratif untuk Use Case bertempur dengan musuh

Nama Use Case	Bertempur dengan musuh
Deskripsi Singkat	Use Case ini memungkinkan pemain menembakkan peluru dari pesawat yang dimainkannya untuk menghancurkan musuh.
Aktor	Pemain
Pra Kondisi	Permainan telah dimulai dan sedang berjalan.
Tindakan Utama	1. Pemain menekan tombol "spasi" atau 0 dan karakter Z atau X pada papan ketik untuk menembakkan senjata dan peluru kendali. 2. Untuk menghindar dari pesawat musuh dan tembakannya, pemain dapat menekan tombol pada papan ketik seperti yang telah dijelaskan pada Tabel 2.1.
Tindakan alternatif	-
Pasca Kondisi	Jika Use Case berhasil dijalankan maka akan terjadi pertempuran antara pesawat dengan pesawat musuh. Namun jika Use Case tidak dijalankan, maka pesawat tidak akan dapat bertempur dengan musuh yang dapat menyebabkan hancurnya pesawat karena tembakan musuh.

3.3 Diagram Aktivitas

Diagram aktifitas merupakan pemodelan alur kerja (*workflow*) sebuah proses, atau urutan aktifitas dalam suatu proses. Diagram Aktivitas menggambarkan langkah yang mana yang harus dijalankan secara berurutan dan langkah mana yang bisa dijalankan secara bersamaan.

Diagram Aktivitas untuk memulai permainan ditunjukkan oleh Gambar 3.2, diagram Aktivitas untuk mengontrol pesawat ditunjukkan oleh Gambar 3.3, dan diagram Aktivitas untuk bertempur dengan musuh ditunjukkan oleh Gambar 3.4.

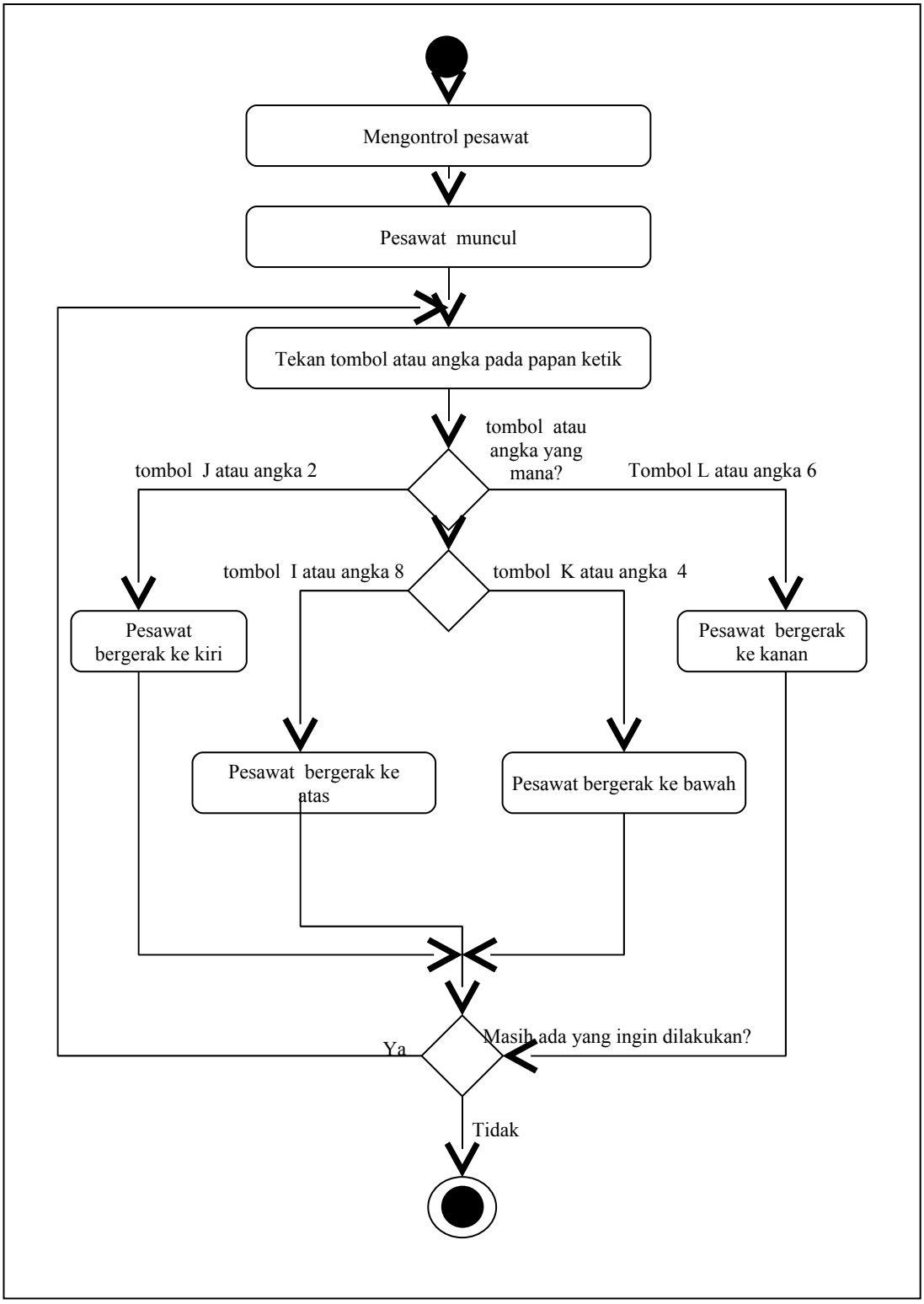


Gambar 3.2. Diagram Aktivitas untuk Use Case memulai permainan.

Pada Gambar 3.2, aktivitas dimulai dengan memilih tingkat kesulitan yang akan dimainkan., aktivitas selanjutnya adalah memulai permainan. Setelah permainan dimulai, aktivitas selanjutnya adalah mengontrol pesawat dan bertempur dengan musuh. Kedua aktivitas tersebut dapat dilakukan secara bersamaan. Untuk lebih jelas mengenai aktivitas mengontrol pesawat, lihat Gambar 3.3 dan untuk lebih jelas mengenai aktivitas bertempur dengan musuh lihat Gambar 3.4.

Pada Gambar 3.3 Aktivitas mengontrol pesawat diawali dengan munculnya pesawat ke layar permainan. Pemain dapat mengontrol pesawat melalui penekanan tombol atau angka pada papan ketik (Keyboard). Pada diagram Aktivitas tersebut terdapat pohon keputusan untuk menentukan penekanan tombol mana yang ditekan

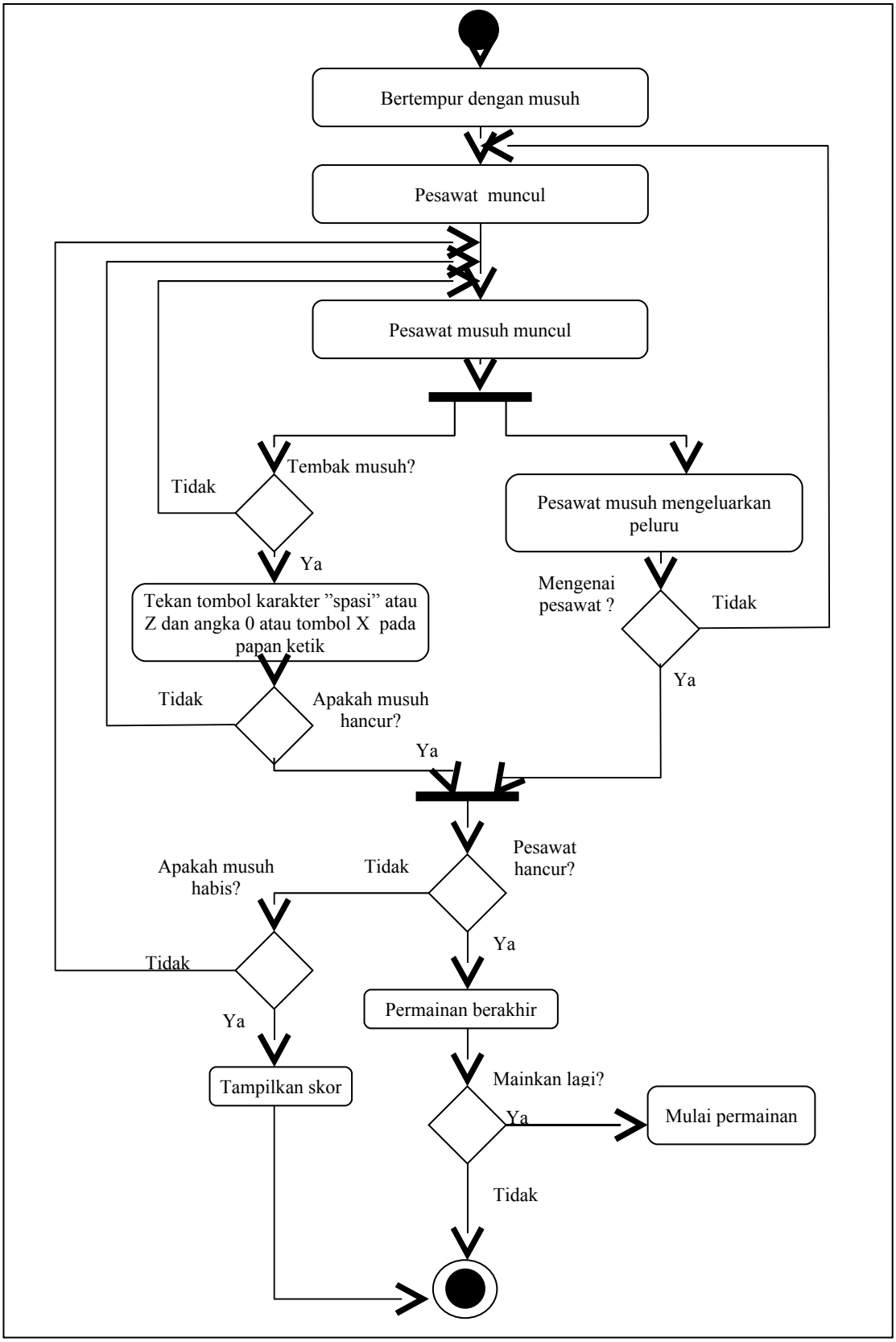
oleh pemain. Apabila tombol J atau agka 2 pada papan ketik ditekan, maka pesawat akan bergerak ke arah kiri pada layar permainan. Apabila tombol L atau angka 6 ditekan, maka pesawat akan bergerak ke arah kanan. Apabila tombol K atau angka 4 ditekan, maka pesawat akan bergerak ke arah bawah. Apabila tombol I atau angka 8 ditekan, maka pesawat akan bergerak ke arah atas. Aktivitas selanjutnya adalah pemain harus menentukan apakah masih ada yang akan dilakukannya. Jika masih ada yang akan dilakukan, maka kembali kepada aktivitas menekan tombol atau angka pada papan ketik. Jika tidak, maka aktivitas berhenti.



Gambar 3.3. Diagram Aktivitas untuk Use Case mengontrol pesawat

Pada Gambar 3.4, aktivitas diawali ketika pesawat muncul di layar permainan, kemudian pesawat musuh muncul dari arah kanan dan bawah pada layar permainan.

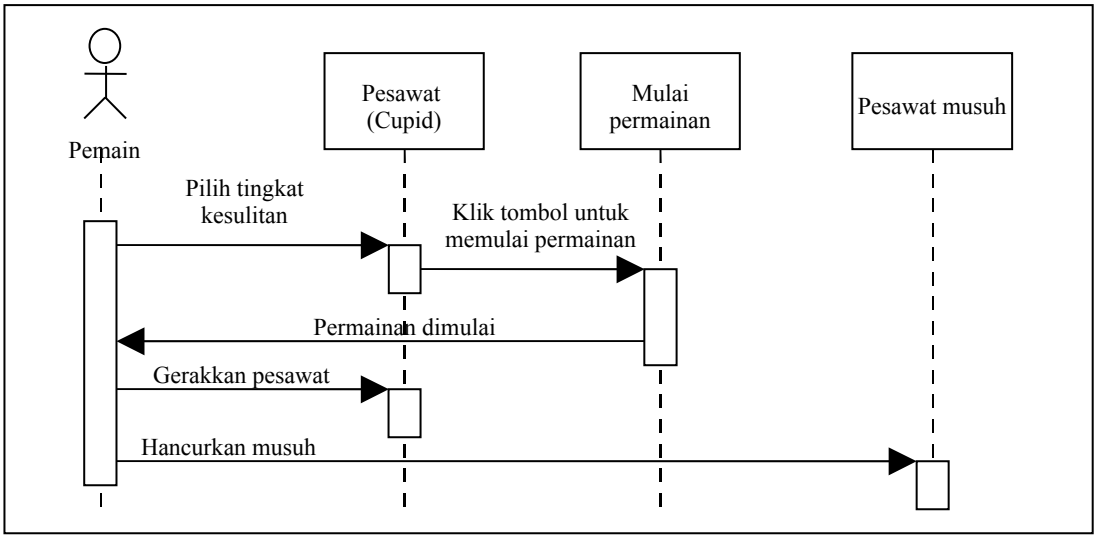
Pesawat musuh mengeluarkan peluru sebanyak kurang lebih tiga peluru ke arah kiri dan atas dari layar permainan dengan kecepatan peluru acak. Apabila pesawat peluru mengenai pesawat atau menabrak tanah maka pesawat hancur dan permainan berakhir. Tetapi apabila pesawat tidak hancur maka pesawat tetap muncul di layar permainan. Pesawat dapat menembak musuh yang muncul di layar permainan dengan menekan tombol "spasi" atau tombol Z dan angka 0 atau tombol X pada papan ketik. Apabila pesawat musuh telah habis dan pesawat tidak hancur maka skor keseluruhan dari permainan akan ditampilkan. Tetapi jika pesawat terkena tembakan pesawat musuh dan hancur maka permainan berakhir. Aktivitas selanjutnya adalah menentukan apakah akan memulai permainan kembali, jika ya maka permainan akan dimulai kembali sesuai dengan level terakhir yang telah dimainkan dan jika tidak maka aktivitas berakhir. Aktivitas pesawat menembak dan pesawat musuh mengeluarkan peluru dapat dilakukan secara bersamaan sehingga digunakan simbol Fork pada diagram Aktivitas tersebut.



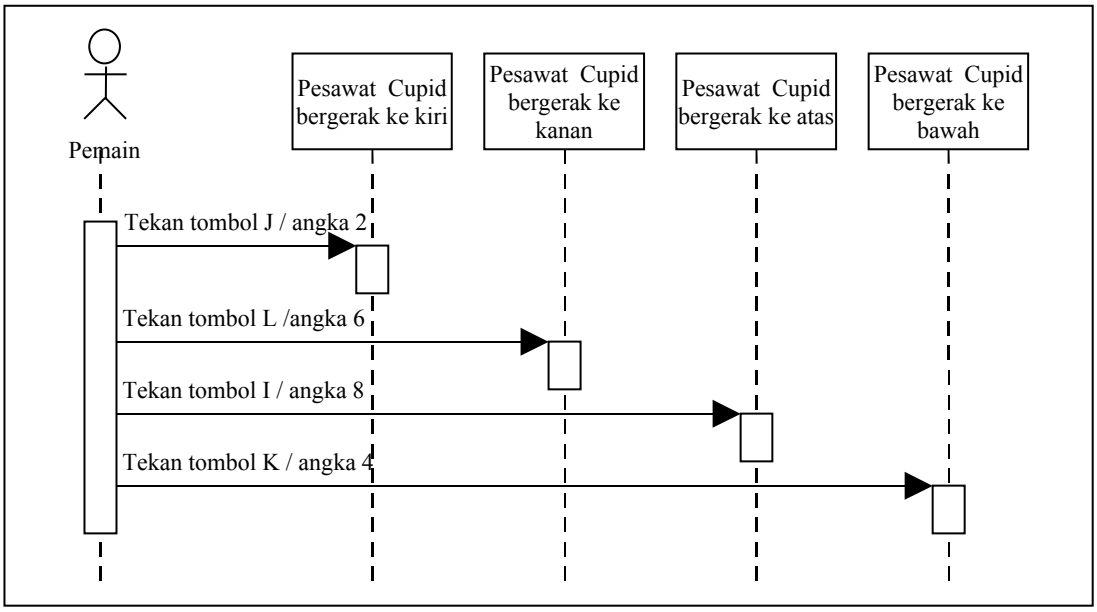
Gambar 3.4. Diagram Aktivitas untuk Use Case bertempur dengan musuh

3.4 Diagram Sekuensial

Diagram Sekuensial untuk Use Case memulai permainan ditunjukkan oleh Gambar 3.5, diagram Sekuensial untuk Use Case mengontrol pesawat Cupid ditunjukkan oleh Gambar 3.6, dan diagram Sekuensial untuk Use Case bertempur dengan musuh ditunjukkan oleh Gambar 3.7.



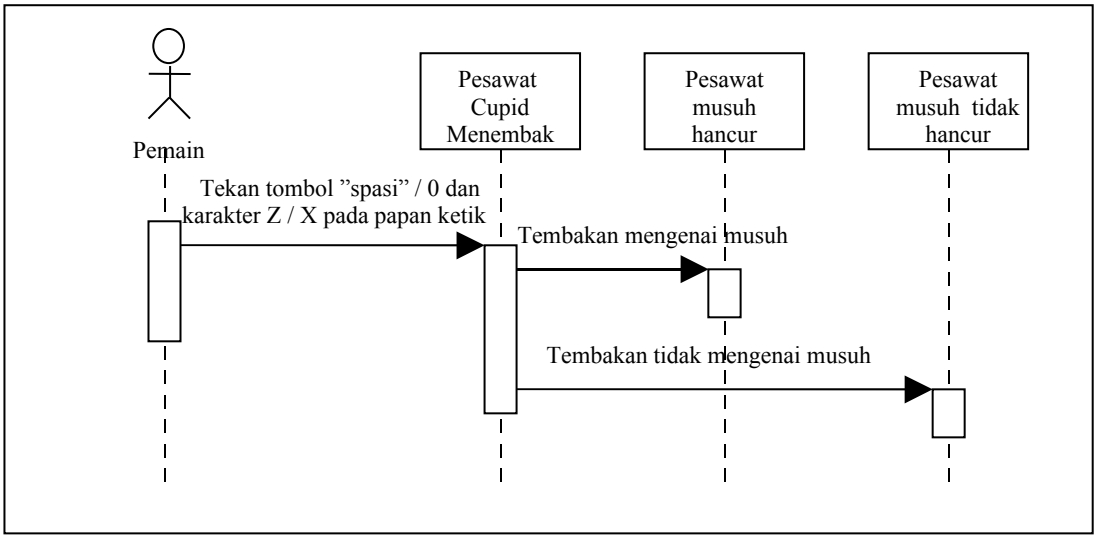
Gambar 3.5. Diagram Sekuensial untuk Use Case memulai permainan



Gambar 3.6. Diagram Sekuensial untuk Use Case mengendalikan pesawat

Pada Gambar 3.5, jika aktor pemain ingin memulai permainan, maka ia diharuskan memilih tingkat kesulitan permainan, setelah itu pemain harus menekan tombol S untuk memulai permainan pada aplikasi permainan tersebut. Selanjutnya jika permainan telah dimulai pemain dapat menggerakkan pesawat Cupid dan dapat menghancurkan musuh (bertempur dengan musuh).

Pada Gambar 3.6, ketika aktor pemain menekan tombol J atau agka 2 pada papan ketik ditekan, maka pesawat akan bergerak ke arah kiri pada layar permainan. Apabila tombol L atau angka 6 ditekan, maka pesawat akan bergerak ke arah kanan. Apabila tombol K atau angka 4 ditekan, maka pesawat akan bergerak ke arah bawah. Apabila tombol I atau angka 8 ditekan, maka pesawat akan bergerak ke arah atas.



Gambar 3.7. Diagram Sekuensial untuk Use Case bertempur dengan musuh

Pada Gambar 3.7, ketika aktor pemain menekan tombol “spasi” atau tombol Z dan angka 0 atau tombol X pada papan ketik pada papan ketik, maka obyek pesawat Cupid mengeluarkan tembakan. Jika tembakan pesawat cupid mengenai obyek pesawat musuh, maka obyek pesawat musuh tersebut akan hancur. Namun jika

tembakan pesawat Cupid tidak mengenai obyek pesawat musuh, maka obyek pesawat musuh tersebut tidak hancur.

3.5 Algoritma Perhitungan Skor

Berikut ini adalah algoritma perhitungan skor yang akan diterapkan pada aplikasi permainan:

1. Skor awal adalah nol (0).
2. Bila salah satu pesawat musuh berhasil ditembak jatuh (dihancurkan) oleh pesawat pemain maka akan mendapat nilai kelipatan 10 dan kelipatan 50 jika pemain berhasil menghancurkan musuh dengan peluru kendali..
3. Ulangi langkah nomor 2 hingga permainan berakhir.

3.6 Algoritma Kehancuran Pesawat

Berikut ini adalah algoritma kehancuran pesawat:

1. Nilai awal pesawat adalah nol (0).
2. Apabila pesawat pemain terkena peluru pesawat musuh atau bertabrakan dengan pesawat musuh dan tanah, maka pesawat akan hancur (kalah) dan permainan berakhir.

Sedangkan algoritma kehancuran pesawat musuh sama dengan algoritma kehancuran pesawat, yang membedakan adalah pesawat musuh tidak hancur jika bertabrakan dengan tanah.

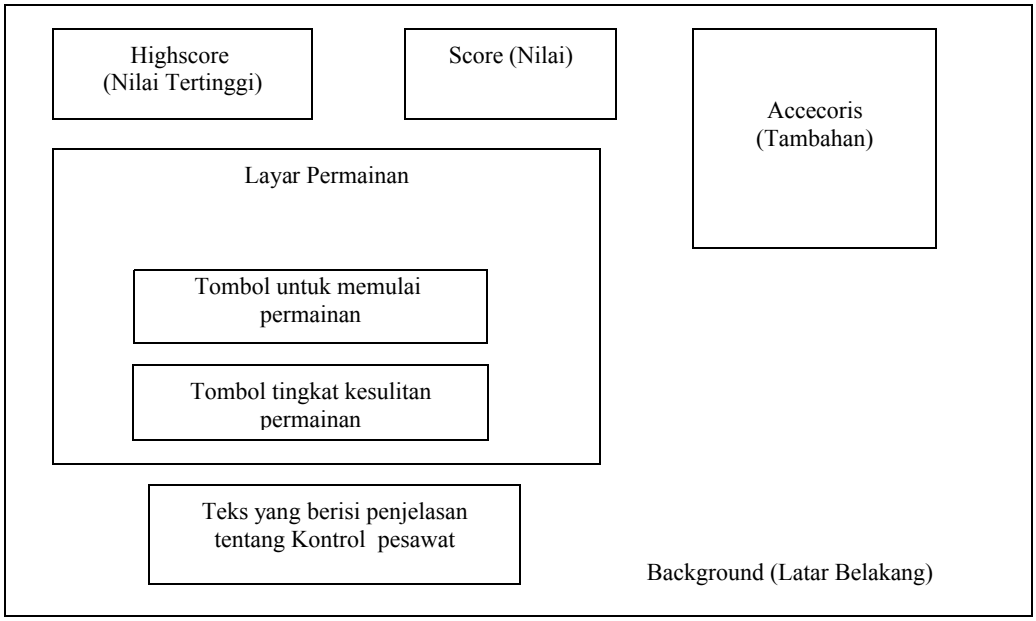
3.7 Antarmuka Permainan

Pada aplikasi permainan, terdapat satu (1) buah jendela yang berfungsi sebagai antarmuka bagi pengguna dengan aplikasi permainan Jendela tersebut yaitu jendela permainan. Pada subbab berikut akan dijelaskan mengenai jendela tersebut.

3.7.1 Jendela Inisialisasi

Gambar 3.8 memperlihatkan jendela inisialisasi yang berfungsi sebagai jendela yang muncul di layar monitor ketika program aplikasi permainan Cupid dijalankan.

Pada Gambar 3.8 Pemain dapat memulai permainan dengan memilih tingkat kesulitan lalu menekan tombol S untuk memulai permainan. Pemain dapat melihat kontrol dari pesawat yang akan dimainkan di bawah jendela permainan atau layar permainan dan nilai tertinggi dari para pemain di kiri atas dan nilai permainan di kanan atas serta accecoris tambahan di sebelah kanan layar permainan.



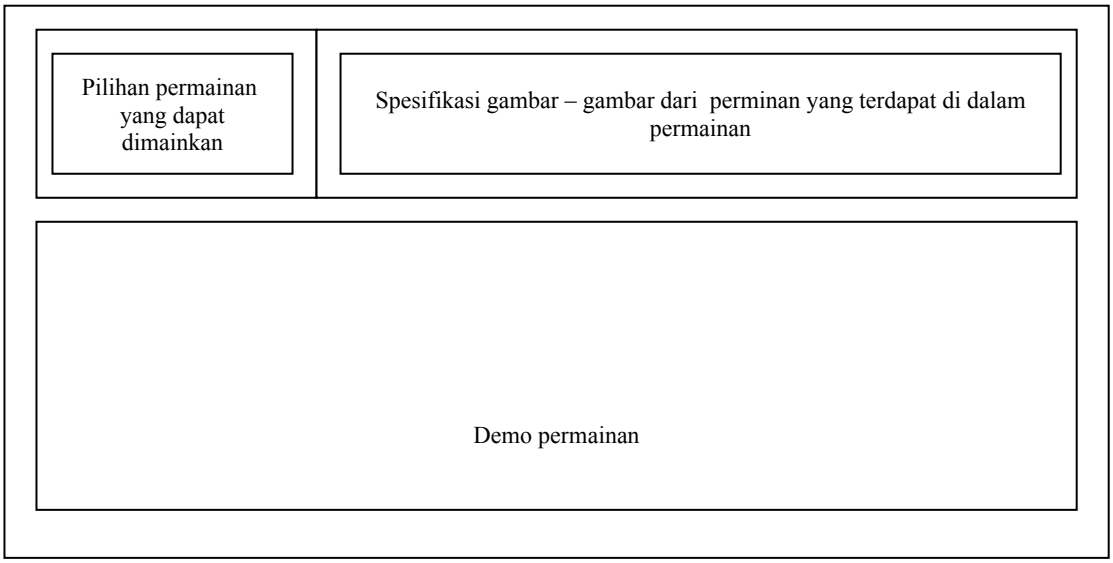
Gambar 3.8. Jendela inisialisasi

3.8 **Perancangan Antarmuka Situs Web**

Setelah antarmuka permainan pesawat udara (cupid), tahap selanjutnya adalah merancang antarmuka situs web yang akan digunakan untuk mengimplementasikan permainan pesawat udara tersebut. Pada subbab berikut akan dijelaskan mengenai perancangan dari antarmuka situs web tersebut.

3.8.1 Perancangan Antarmuka Awal

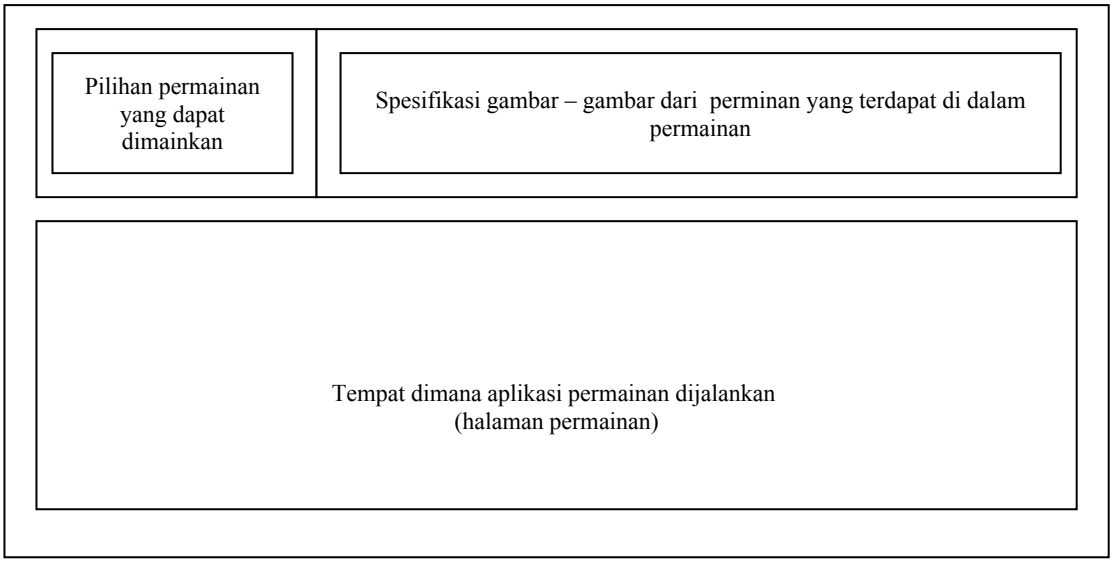
Antarmuka awal adalah antarmuka ketika situs web pertama kali diakses. Pada antarmuka awal ini, terdapat pilihan permainan yang dapat dimainkan oleh pengguna aplikasi. Tentu saja semua permainan yang disediakan adalah permainan yang dibangun menggunakan java, tetapi hanya bersifat tambahan. Pada spesifikasi permainan terdapat teks yang menghubungkan ke halaman permainan. Pada spesifikasi permainan terdapat teks yang menghubungkan ke halaman permainan. Lihat Gambar 3.9.



Gambar 3.9 Perancangan antarmuka ketika situs web pertama kali diakses

3.8.2 Perancangan Antarmuka Halaman Permainan

Setelah pengguna menekan teks yang menghubungkan ke halaman permainan, maka halaman permainan tersebut akan ditampilkan di bawah dari halaman awal. Pada halaman permainan inilah, pengguna dapat memainkan permainan pesawat udara yang dibangun. Untuk lebih jelas lihat Gambar 3.10.



Gambar 3.10. Perancangan antarmuka halaman permainan

BAB IV

IMPLEMENTASI DAN PENGUJIAN SISTEM

4.1 Implementasi

Setelah tahap analisis dan perancangan selesai dilakukan, maka tahap selanjutnya adalah tahap implementasi. Pada subbab berikut akan dijelaskan implementasi dari aplikasi permainan komputer dengan karakter pesawat udara (Cupid) yang terdiri dari lingkungan implementasi, pengkodean, dan antarmuka dari aplikasi permainan tersebut.

4.1.1 Lingkungan Implementasi

Aplikasi permainan komputer dengan karakter pesawat udara dikembangkan dan diimplementasikan pada seperangkat komputer pribadi (*Personal Computer*) dengan spesifikasi sebagai berikut:

1. Perangkat Keras yang digunakan:
    a. Prosesor : Intel Pentium IV 1.7 Ghz.
    b. RAM : 128 MB
    c. VGA : 32 MB
    d. Harddisk : Seagate 40 Gb 7200 Rpm
    e. Monitor : SPC 15 inc
2. Perangkat Lunak yang digunakan:
    a. Microsoft Windows XP Professional, sebagai Sistem Operasi.
    b. *Web Browser* Internet Explorer versi 5.0 dan Opera 6.0.

c. Javascript versi 1.3, sebagai alat (*tool*) untuk membangun aplikasi permainan tersebut.

4.1.2 Pengkodean

Pada subbab ini penulis menjelaskan cara membuat program ini dengan menggunakan Javascript versi 1.3. Program ini dapat dimainkan melalui Internet Explorer 5.0. Di dalam memainkan program ini, skrip dan gambar yang terdapat harus diletakkan di dalam kotak yang sama, karena jika skrip dan gambar diletakkan pada kotak yang berbeda, maka program ini tidak dapat dimainkan. Karena *path default* (hubungan normal) yang terdapat di dalam javascript, menghubungkan antara variabel, skrip dan gambar dalam kotak yang sama.

Di dalam memulai program harus dituliskan <html> dan <script language="javascript">. Kode tersebut harus dituliskan pada bagian <script language="javascript"> agar skrip kode yang dituliskan dapat dibaca oleh segala versi skrip. Sebelum variabel terlebih dahulu harus menuliskan <!— *begin* (mulai) agar program tersebut mempunyai awalan. Setelah menuliskan kode tersebut, penulisan variabel dimaksudkan untuk mengacu pada suatu nilai dalam suatu program.

Pendeklarasian variebel dilakukan dengan cara eksplisit, yaitu pendeklarasian dengan cara menuliskan kata kunci var kemudian diikuti dengan nama variabel dan nilai dari variabel. Variabel terdiri dari 4 jenis tipe varibel, yaitu *integer* (bilangan bulat), *float* (bilangan decimal), *string* (karakter), dan b*oolean* (khusus). Variabel-variabel yang dideklarasikan di atas mempunyai fungsi masing-masing di dalam menjalankan program ini, seperti :

- **flimit=9** adalah harga awal yang diberikan kepada pemain ketika memulai permainan tanpa memilih tingkat kesulitan, di dalam hal ini pemain diberikan tingkat kesulitan normal.
- **bosstm=0** adalah harga awal dari waktu yang diberikan kepada pemimpin dari musuh.
- **bosslm=700** merupakan harga awal dari hitpoint (kekuatan) yang diberikan kepada pimpinan musuh.
- **easy=0** adalah nilai yang diberikan kepada variabel easy.
- **timerid=null** adalah nilai dari waktu permainan yang tidak terbatas,
- **timercounter=0** merupakan nilai dari waktu pengulangan ketika pemain kalah oleh musuh, untuk varibel yang memakai objek array, seperti cw = new Array(2) berarti nilai yang diberikan untuk di tampung pada urutan ke dua di dalam variabel cw.
- **var timecounter=0** adalah waktu pengulangan permainan jika permainan telah berakhir.
- **var int=25** adalah nilai awal integer yang diberikan di dalam permainan, nilai awal ini berguna untuk pergerakan.
- **var k=0** adalah harga awal yang diberikan penulis terhadap pemasukan melalui papan ketik.
- **var x=100** adalah letak posisi pemain secara horizontal pada awal permulaan pemain.
- **var y=100** letak posisi pemain secara vertikal pada awal permulaan pemain.
- **var dx=0** adalah harga awal yang diberikan untuk pergerakan pemain secara horizontal dan bernilai positif.

- **var dy=0** adalah harga awal yang diberikan untuk pergerakan pemain secara vertikal dan bernilai positif.
- **var ix =0** adalah harga awal yang diberikan untuk pergerakan pemain secara horizontal dan bernilai negatif.
- **var iy=0** adalah harga awal yang diberikan untuk pergerakan pemain secara vertikal dan bernilai negatif.
- **Sdx=0** adalah harga awal yang diberikan untuk pergerakan pemain secara diagonal yaitu perpotongan antara titik –x dengan –y.
- **var sdy=0** adalah harga awal yang diberikan untuk pergerakan pemain secara diagonal yaitu perpotongan antara titik x dengan –y.
- **var six=0** adalah harga awal yang diberikan untuk pergerakan pemain secara diagonal yaitu perpotongan antara titik –x dengan y.
- **var siy =0** adalah harga awal yang diberikan untuk pergerakan pemain secara diagonal yaitu perpotongan antara titik x dengan y.
- **var endflg =0** adalah variabel yang memberitahukan bahwa permainan terhenti.
- **var efx=new array(8)** adalah pembuatan baris baru sebanyak delapan buah yang berguna untuk pergerakan musuh secara horizontal dan bernilai positif.
- **var efy=new array(8)** adalah pembuatan baris baru sebanyak delapan buah yang berguna untuk pergerakan musuh secara vertikal dan bernilai positif**.**
- **var evx= new array(8)** adalah pembuatan baris baru sebanyak delapan buah yang berguna untuk pergerakan musuh secara horizontal dan bernilai negatif.
- **Var evy=new array(8)** adalah pembuatan baris baru sebanyak delapan buah yang berguna untuk pergerakan musuh secara vertikal dan bernilai negatif**.**

- **var fmax=3** adalah jumlah maksimum pengulangan permainan pada tingkat kesulitan mudah.
- **var emx=200** adalah harga awal yang diberikan kepada pemain untuk menghindari serangan dari musuh secara horizontal.
- **var emy=284** adalah harga awal yang diberikan kepada pemain untuk menghindari serangan dari musuh secara vertikal.
- **var temp1=0** adalah harga awal untuk nilai sementara yang diberikan kepada pemain.
- **var temp2=0** adalah harga awal untuk nilai sementara yang diberikan kepada musuh.
- **var endflg=1** adalah nilai yang diberikan untuk mengetahui pemasukan dilakukan melalui papan ketik.
- **Var bossflg=0** adalah nilai yang memberitahukan bahwa pemain berhasil mengalahkan bos dari musuh.
- **var bossx=0** adalah harga awal untuk posisi bos secara horizontal untuk menghadapi pemain.
- **var bossh=0** adalah harga awal untuk menembakan peluru khusus yang dikeluarkan oleh bos musuh.
- **var canonx=-200** adalah posisi awal peluru secara horizontal yang dikeluarkan oleh pemain ataupun musuh.
- **var canony=-200** adalah posisi awal peluru secara vertikal yang dikeluarkan oleh pemain ataupun musuh.
- **Var canonc=0** adalah jumlah awal dimana peluru ditembakan oleh pemain ataupun musuh.

- **var tmpbf=0** adalah nilai sementara dari pergerakan bos.
- **var sf=0** adalah harga awal untuk pergerakan peluru kendali yang dikeluarkan oleh bos.
- **var spcount=0** adalah harga awal yang digunakan untuk menghitung nilai yang didapat oleh pemain.

Pengkodean diperlukan agar rancangan dapat diterjemahkan ke dalam bentuk mesin dan dapat dijalankan. Pada subbab ini penulis akan menjelaskan penggalan baris program pada pengembangan aplikasi permainan komputer dengan karakter pesawat udara. Berikut ini penggalan kode program yang terdapat pada aplikasi permainan tersebut:

1. Potongan kode untuk peletakan gambar dan pergerakan number (angka) pada layar permainan.

   Sebelum membuat program, penulis menentukan gambar yang akan dipakai di dalam permainan serta pergerakan number (angka). Pemilihan gambar tersebut bertujuan untuk memperindah tampilan dari permainan yang dibuat. Pada kode 4.1, berikut adalah potongan kode yang digunakan di dalam meletakan objek

Kode 4.1. Potongan kode program peletakan gambar

```
1.  <DIV STYLE='position:absolute; left:16; top:0'><IMG WIDTH=50 HEIGHT=16
    SRC="highc.gif"></DIV>
2.  <DIV STYLE='position:absolute; left:60; top:0'><IMG WIDTH=50 HEIGHT=16
    SRC="score.gif"></DIV>
3.  <DIV ID="L2I" STYLE='position:absolute; left:120; top:0'><IMG WIDTH=16
    HEIGHT=160 SRC="number.gif"></DIV>
4.  <DIV ID="L3I" STYLE='position:absolute; left:130; top:0'><IMG WIDTH=16
    HEIGHT=160 SRC="number.gif"></DIV>
5.  <DIV ID="L4I" STYLE='position:absolute; left:140; top:0'><IMG WIDTH=16
    HEIGHT=160 SRC="number.gif"></DIV>
6.  <DIV ID="L5I" STYLE='position:absolute; left:150; top:0'><IMG WIDTH=16
    HEIGHT=160 SRC="number.gif"></DIV>
7.  <DIV ID="L6I" STYLE='position:absolute; left:160; top:0'><IMG WIDTH=16
    HEIGHT=160 SRC="number.gif"></DIV>
8.  <DIV STYLE='position:absolute; left:290; top:0'><IMG WIDTH=50 HEIGHT=16
    SRC="score.gif"></DIV>
9.  <DIV ID="L7I" STYLE='position:absolute; left:350; top:0'><IMG WIDTH=16
    HEIGHT=160 SRC="number.gif"></DIV>
10. <DIV ID="L8I" STYLE='position:absolute; left:360; top:0'><IMG WIDTH=16
    HEIGHT=160 SRC="number.gif"></DIV>
11. <DIV ID="L9I" STYLE='position:absolute; left:370; top:0'><IMG WIDTH=16
    HEIGHT=160 SRC="number.gif"></DIV>
12. <DIV ID="L10I" STYLE='position:absolute; left:380; top:0'><IMG WIDTH=16
    HEIGHT=160 SRC="number.gif"></DIV>
13. <DIV ID="L11I" STYLE='position:absolute; left:390; top:0'><IMG WIDTH=16
    HEIGHT=160 SRC="number.gif"></DIV>
14. <DIV STYLE='position:absolute; left:16; top:16'><IMG WIDTH=400 HEIGHT=300
    SRC="skybg.png"></DIV>
```

Pada Kode 4.1, potongan kode program pada baris ke 1 adalah peletakan untuk gambar *highscore* (nilai tertinggi). Letak dari gambar nilai tertinggi tersebut terdapat di titik koordinat 16 dari kiri, sedangkan img width=50 berarti ukuran lebar dari gambar tersebut 50, dan height=16 berarti tinggi dari ukuran gambar tersebut 16, di dalam hal ini ukuran dari gambar tersebut memakai satuan piksel. Baris ke 2 merupakan peletakan untuk gambar *score* (nilai), letak dari gambar nilai tersebut terdapat di titik koordinat 60 dari kiri, sedangkan img width=50 berarti ukuran lebar dari gambar tersebut 50, dan height=16 berarti tinggi dari ukuran gambar tersebut 16 piksel. Baris ke 3 hingga baris ke 7 menunjukkan pergerakan dari pergantian angka dari dua sampai enam yang terdapat di dalam gambar number (angka). Posisi dari pergerakkan angka tersebut dimulai dari titik

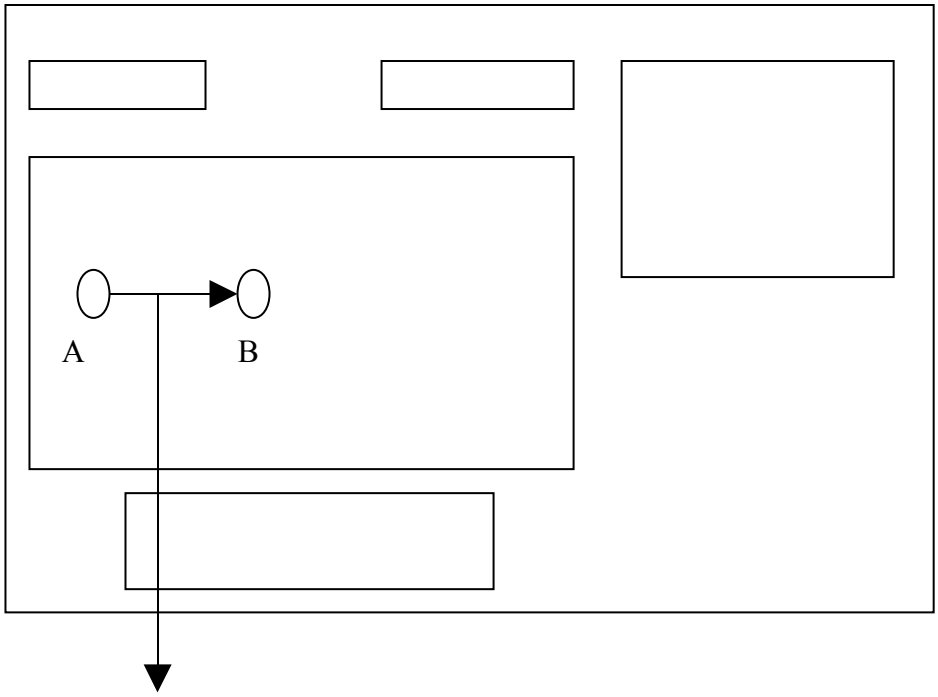
koodinat 120 dan di akhiri di titik 160, sesuai dengan jarak yang terdapat di dalam gambar angka tersebut. Sedangkan baris ke 8 hingga baris ke 13 adalah pergerakan dari pergantian angka dari tujuh sampai satu yang terdapat di dalam gambar number (angka). Pergantian nilai tersebut dimulai dari titik 290 dari kiri, dan untuk peletakan angka tersebut dimulai dari titik 350 dari kiri sampai 390, dan ukuran lebar dari gambar angka tersebut adalah 16 sedangkan ukuran tingginya adalah 160 piksel. Dan baris ke 14 adalah peletakan dari latar belakang 2 dari permainan ini, potongan kode program latar belakang tersebut diletakkan di titik 16 koordinat dari kiri, untuk ukuran lebar dari latar belakang 2 tersebut adalah 400 dan tinggi 300 piksel.

2. Potongan kode program pergerakan objek

   Untuk menggerakan objek pemain dari tempat yang satu ke tempat yang lain dipakai fungsi emove (), seperti yang ditunjukkan oleh Kode 4.2.

Kode 4.2. Potongan kode program pergerakan objek

```
1.  function Efmove(){
2.  for (tmp1=0; tmp1<Fmax; tmp1++){
3.  if (EFX[tmp1] != -100){
4.  EFX[tmp1]=EFX[tmp1]+EVX[tmp1];EFY[tmp1]=EFY[tmp1]+EVY[tmp1]
5.  if (EFX[tmp1] < -16){EFX[tmp1]=-100;EVX[tmp1]=0;EVY[tmp1]=0}
6.  if (EFX[tmp1] > 400){EFX[tmp1]=-100;EVX[tmp1]=0;EVY[tmp1]=0}
7.  if (EFY[tmp1] < -16){EFX[tmp1]=-100;EVX[tmp1]=0;EVY[tmp1]=0}
8.  if (EFY[tmp1] > 300){EFX[tmp1]=-100;EVX[tmp1]=0;EVY[tmp1]=0}}
9.  if (EFX[tmp1] >= X-4){
10. if (EFX[tmp1] <= X+26){
11. if (EFY[tmp1] >= Y+8){
12. if (EFY[tmp1] <= Y+24){
13. mycr.visibility="hidden"
14. DOC2c.visibility="hidden"
15. DOCc.visibility="visible"
16. spMOV(Cname[cflag],X,Y)
17. Endflg=1;EFX[tmp1]=-100;cflag=cflag+1;if (cflag >= 2){cflag=0}}}}}
18. spMOV(tmp1+18,EFX[tmp1],EFY[tmp1])}}
```



Pergerakan objek dari titik A ke titik B

Gambar 4.1. Pergerakan Objek

Pada kode 4.2, potongan kode program pada baris ke 1 memanggil fungsi emove untuk menggerakan objek pemain.Sedangkan pada baris ke 2 hingga baris ke 18 di atas mempunyai arti untuk nilai awal tmp1=0 berarti tmp1 mempunyai nilai lebih kecil dari fmax, maka nilai tmp1 ditambah dengan 1, dan jika pergerakan sementara dititik x dari pemain -100 dari letak awal tempat pemain berada, maka ditambahkan nilai efx[tmp1] dengan nilai yang dimasukkan karena efx merupakan variabel array dan begitu juga dengan letak pemain berada di titik y ditambahkan nilai tmp1 dengan nilai masukan dari efy[tmp1], untuk baris ketiga jika nilai dari efx[tmp1]<-16 yaitu titik x dimana pemain berada adalah -16 dari posisi awal, maka posisi pemain berada dititik x -100, untuk nilai evx dan evy yaitu posisi pemain dari sebelah kanan bernilai 0, begitu juga dengan baris-baris berikutnya, untuk x adalah titik horizontal dan untuk y adalah titik vertikal. Semua skrip tersebut mengatur pergerakan pemain dan musuh agar perpindahan

pemain dan musuh dari titik pertama ke titik yang lain tidak terlalu jauh. Di baris ke 14 fungsi *emove* ini terdapat doc2c.visibility="hidden" dan docc.visibility="visible", skrip doc2c.visibility="hidden" berarti pergerakan untuk pemain yang memilih tingkat kesulitan normal, sedangkan docc.visibility="visible" berarti pergerakan untuk pemain yang memilih tingkat mudah.

3. Potongan kode program penambahan nilai

   Untuk menambahkan nilai yang didapat dengan menghancurkan musuh – musuh digunakan fungsi *loadscore* (). Skrip dari penambahan nilai adalah :

   Kode 4.3. Potongan kode program penambahan nilai

```
1.  function loadSCORE(){
2.  if (document.cookie){
3.  CUPIDtop = document.cookie.indexOf("JSCUPID", 0)
4.  if (CUPIDtop != -1){
5.  tscore = document.cookie.substring(CUPIDtop+9,CUPIDtop+14)
6.  }else{tscore="0"}
7.  }else{tscore="0"}
8.  with (Math) {hscore=parselnt(tscore,10)
9.  h5=floor(hscore/100)
10. h4=floor((hscore-h5*10)/100)
11. h3=floor(hscore-h5*10-h4*10)/100))
12. h2=floor(hscore-h5*10-h4*10-h3*10)/100))
13. h1=hscore-h5*10-h4*10-h3*10-h2*10}
14. L2c.top=-16,h5
15. L3c.top=-16,h4
16. L4c.top=-16,h3
17. L5c.top=-16,h2
18. L6c.top=-16,h1}
```

   Pada kode 4.3, baris ke 1 hingga baris ke 8 memanggil fungsi menambah nilai dan nilai awal perminan adalah nol, nilai hscore=parseint(tscore,10) berarti nilai tertinggi yang pemain dapat selama memainkan program ini. Nilai *hscore* didapat dengan cara menghancurkan musuh-musuh. Setiap kali pemain menghancurkan satu musuh, maka pemain mendapatkan nilai sebanyak 10, tetapi nilai tersebut tidak akan langsung dimasukkan ke dalam kotak nilai tertinggi karena harus

melalui h5=floor(hscore/10000) dengan begitu pemain akan merasakan tantangan di dalam memainkan program ini. Pada baris ke 9 hingga baris ke 13 dimana h5=floor(hscore/100), h4=floor((hscore-h5*10)/100), h3=floor((hscore-h5*10-h4*10)/100), h2=floor((hscore-h5*10-h4*10-3*10)/100), h1=hscore-h5*10-4*10-h3*10-h2*10, berarti untuk h5=floor(hscore/100) berarti nilai yang diperoleh pemain dibagi dengan 100 dan jika hasil dari pembagian tersebut bernilai kurang dari 1, maka dibulatkan menjadi 0, untuk h4=floor((hscore-h5*10)/100) berarti nilai pada h5 dikalikan dengan 10 kemudian dikalikan dengan nilai yang diperoleh pemain dan dibagi 100 dan seterusnya untuk nilai tertinggi pada posisi ke 3, 2, dan 1. H5 adalah nilai tertinggi pertama, h4 adalah nilai tertinggi kedua dan seterusnya sampai h1. Nilai yang terdapat pada kotak nilai tertinggi akan terus berganti selama pemain mengulang permainan tetapi jika pemain tidak dapat melebihi yang didapat pada permainan pertama, maka nilai tertinggi pada permainan pertama akan tetap berada pada kotak nilai tertinggi sampai pemain dapat melewati nilai yang didapat pada permainan pertama. Perumusan nilai ini dimaksudkan agar pergantian nilai tertinggi tidak terjadi dengan mudah. Sedangkan pada baris ke 14 hingga baris ke 18 dimana L2c.top=-16,h5, L3c.top=-16,h4, L4c.top=-16,h3, L5c.top=-16,h2, L6c.top=-16,h1 merupakan posisi dari perubahan nilai yang didapat ketika menghancurkan musuh, untuk L2c.top=-16,h5 berarti posisi pertama dari nilai tertinggi tersebut, nilai dari h5 diletakkan dikotak nilai tertinggi yang terdapat pada titik koordinat -16, berikut juga dengan posisi ke empat sampai ke satu.

4. Untuk menampilkan suara setiap kali menjalankan program, harus menambahkan skrip setelah </head>. Skrip dari penambahan suara adalah :

Kode 4.4. Potongan kode program musik pengiring

```
1.  </HEAD>
2.  <!-- ////////////////////////////////////// -->
3.  <BODY BGCOLOR="#00003F" TEXT="#FFFFFF" ONLOAD="onLD()">
4.  <SCRIPT LANGUAGE="JavaScript">
5.  <!--  Begin
6.  var MSIE=navigator.userAgent.indexOf("MSIE");
7.  var IE=navigator.userAgent.indexOf("IE");
8.  var OPER=navigator.userAgent.indexOf("Opera");
9.  if((MSIE>-1) || (OPER>-1)) {
10. document.write("<BGSOUND SRC=sound.mid LOOP=INFINITE>");
11. } else {
12. document.write("<EMBED SRC=sound.mid AUTOSTART=TRUE ");
13. document.write("HIDDEN=true VOLUME=100 LOOP=TRUE>");
14. }
15. // End -->
```

Pada kode 4.4, baris ke 1 merupakan *Header* dari dokumen HTML. Pada baris ke 2 merupakan keterangan tentang manipulasi yang sedang dikerjakan. Pada baris ke 3 berarti suara dimulai setiap menjalankan program karena adanya perintah onload. Pada baris ke 4 hingga baris ke 5 Tag dimulainya JavaScript dalam HTML. Dan pada baris ke 12 yaitu document.write("<embed src=sound.mid autostart=true "); jika autostart=true bernilai false, maka suara tidak akan dapat terdengar setiap program dijalankan dan pada bagian <embed src=sound.mid digunakan untuk memilih tipe musik yang diinginkan, untuk tipe lagu yang dipakai didalam permainan ini adalah soung.mid. Pada baris ke 13 yang berisi document.write("hidden=true volume=100 loop=true>"); suara selalu berulang karena loop bernilai true (benar), sedangkan pada baris ke 5 hingga baris ke 8 dari penambahan suara seperti di atas berarti jenis dari browser yang digunakan sebagai media output program.

5. Potongan kode program pemasukan melalui papan ketik

Penulis melakukan pemasukan melalui papan ketik dengan 2 cara, pertama penulis melakukan pemasukan melalui huruf kecil. Potongan kode dari pemasukan melalui huruf kecil adalah :

Kode 4.5. potongan kode program melalui huruf kecil

```
1.  function keyDown(DnEvents){
2.  if (systm != "C"){
3.  k=DnEvents.which
4.  }else{
5.  k=window.event.keyCode
6.  }
7.  if (k == 50){IY=1}
8.  if (k == 56){DY=1}
9.  if (k == 54){IX=1}
10. if (k == 52){DX=1}
11. if (k == 98 ){IY=1}
12. if (k == 104){DY=1}
13. if (k == 102){IX=1}
14. if (k == 100){DX=1}
15. if (k == 75){IY=1}
16. if (k == 73){DY=1}
17. if (k == 76){IX=1}
18. if (k == 74){DX=1}
19. if (k == 107){IY=1}
20. if (k == 105){DY=1}
21. if (k == 108){IX=1}
22. if (k == 106){DX=1}
23. if (k == 49){SDX=1;SIY=1}
24. if (k == 51){SIX=1;SIY=1}
25. if (k == 55){SDX=1;SDY=1}
26. if (k == 57){SIX=1;SDY=1}
27. if (k == 32){ff=1}
28. if (k == 90){ff=1}
29. if (k == 122){ff=1}
30. if (k == 0){ff=1}
31. if (k == 48){if (bc == 0){bf=1}}
32. if (k == 96){if (bc == 0){bf=1}}
33. if (k == 13){if (bc == 0){bf=1}}
34. if (k == 88){if (bc == 0){bf=1}}
35. if (k == 120){if (bc == 0){bf=1}}
36. if (k == 83){if (Endflg == 1){if (sf == 1){Restart()}}}
37. if (k == 115){if (Endflg == 1){if (sf == 1){Restart()}}}
38. if (k == 69){if (Endflg == 1){Flimit=3;EASY()}}
39. if (k == 101){if (Endflg == 1){Flimit=3;EASY()}}
40. if (k == 78){if (Endflg == 1){Flimit=9;NORMAL()}}
41. if (k == 110){if (Endflg == 1){Flimit=9;NORMAL()}}
42. if (k == 81){QuitPlay()}
43. if (k == 113){QuitPlay()}}
```

Selain pemasukan melalui huruf kecil penulis juga melakukan pemasukan melalui huruf besar. Potongan kode dari pemasukan melalui huruf besar adalah :

Kode 4.6. potongan kode program melalui huruf besar

```
1.  function keyUp(UpEvents){
2.  if (systm != "C"){
3.  k=UpEvents.which
4.  }else{
5.  k=window.event.keyCode
6.  }
7.  if (k == 50){IY=0}
8.  if (k == 56){DY=0}
9.  if (k == 54){IX=0}
10. if (k == 52){DX=0}
11. if (k == 98 ){IY=0}
12. if (k == 104){DY=0}
13. if (k == 102){IX=0}
14. if (k == 100){DX=0}
15. if (k == 75){IY=0}
16. if (k == 73){DY=0}
17. if (k == 76){IX=0}
18. if (k == 74){DX=0}
19. if (k == 107){IY=0}
20. if (k == 105){DY=0}
21. if (k == 108){IX=0}
22. if (k == 106){DX=0}
23. if (k == 49){SDX=0;SIY=0}
24. if (k == 51){SIX=0;SIY=0}
25. if (k == 55){SDX=0;SDY=0}
26. if (k == 57){SIX=0;SDY=0}
27. }
```

Kode 4.6 di atas berarti pemasukan melalui huruf besar. Pemasukan huruf tersebut dimasukan melalui sandi ascii yang diterjemahkan ke dalam tombol huruf dan tombol angka yang terdapat di dalam papan ketik. Sedangkan kode 4.5 pada baris ke 36 hingga baris ke 41 di atas berarti pemasukan huruf untuk mengulang permainan. Pemilihan tingkat kesulitan baik mudah maupun normal tidak terpengaruh melalui besar atau kecil huruf yang dimasukkan, untuk k == 83 berarti pemasukan melalui huruf S (besar), untuk k == 115 berarti pemasukan melalui huruf s (kecil), untuk k == 69 berarti pemasukan melalui huruf E (besar),

k == 101 adalah huruf e (kecil), dan untuk k == 78 berarti pemasukan melalui huruf N (besar), untuk k == 110 berati pemasukan melalui huruf h (kecil).

Pada bagian skrip {if (Endflg == 1){if (sf == 1){Restart()}} dari if (k == 115){if (Endflg == 1){if (sf == 1){Restart()}} berarti setiap pemain menekan tombol s sebanyak satu kali, maka program akan mengulang ke titik awal dari permainan. Pada bagian skrip {if (Endflg == 1){Flimit=3;EASY()}} dari if (k == 101){if (Endflg == 1){Flimit=3;EASY()}} berarti setiap pemain menekan tombol e sebelum program dimulai, maka pemain memilih untuk bermain dengan tingkat kesulitan mudah dan pemain hanya mempunyai tiga kali kesempatan untuk bermain dengan tingkat kesulitan mudah. Dan pada bagian skrip {if (Endflg == 1){Flimit=9;NORMAL()}} dari if (k == 110){if (Endflg == 1){Flimit=9;NORMAL()}}berarti setiap pemain menekan tombol h sebelum program dimulai, maka pemain memilih untuk bermain dengan tingkat kesulitan normal dan pemain hanya mempunyai sembilan kali kesempatan untuk bermain dengan tingkat kesulitan normal. Pemasukan tersebut memakai kode ascii, berikut adalah hasil dari kode ascii di atas setelah diterjemahkan oleh papan ketik.

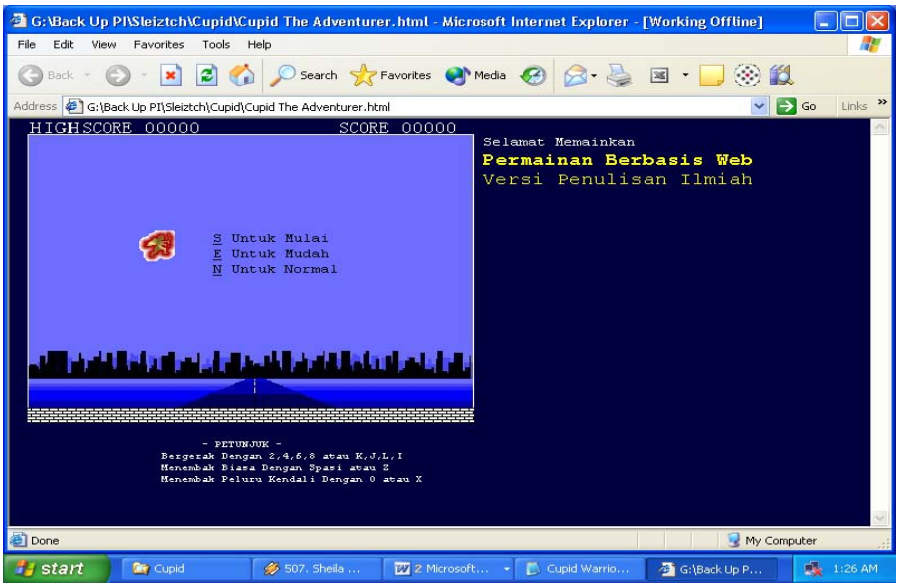
50 = 2	98 = b	75 = K	107 = k	49 = 1
56 = 8	104 = h	73 = I	105 = I	51 = 3
54 = 6	102 = f	76 = L	108 = l	55 = 7
52 = 4	100 = d	74 = J	106 = j	57 = 9
32 = (spasi)	90 = Z	112 = p	0 = esc	88 = X
120 = x	83 = S	115 = s	69 = E	101 = e
78 = N	110 = n	81 = Q	113 = q	

4.1.3 Jendela Permainan

Pada subbab ini penulis akan menampilkan beberapa implementasi jendela dari aplikasi permainan pesawat udara, diantaranya :

4.1.3.1 Jendela inisialisasi

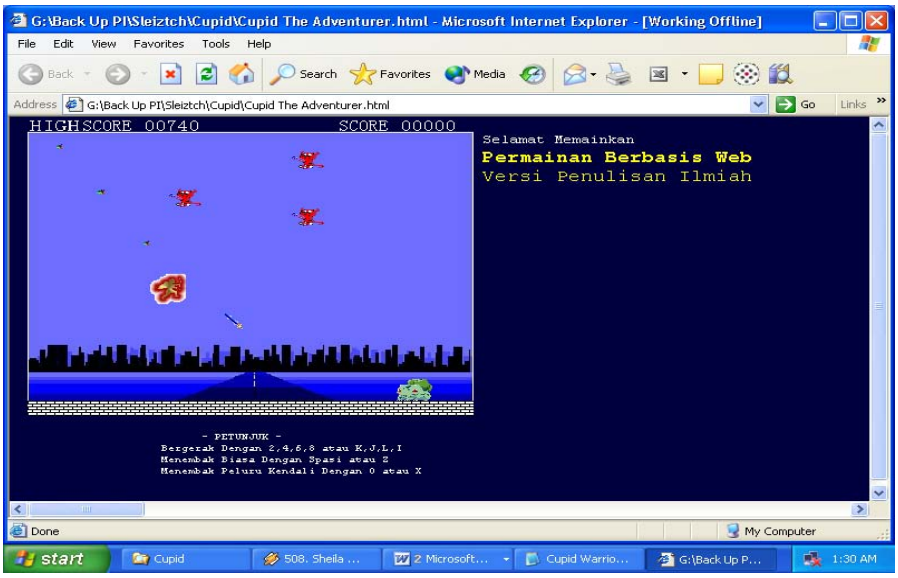
Ketika pemain menjalankan aplikasi permainan komputer dengan karakter pesawat udara, yang pertama kali ditampilkan kepada pemain adalah jendela inisialisasi. Pada jendela tersebut pemain dapat memulai permainan dengan permainan dengan memilih tingkat kesulitan lalu menekan tombol S untuk memulai permainan. Pemain dapat melihat kontrol dari pesawat yang akan dimainkan di bawah jendela permainan atau layar permainan dan nilai tertinggi dari para pemain di kiri atas dan nilai permainan di kanan atas serta accecoris tambahan di sebelah kanan layar permainan. Untuk lebih jelas lihat Gambar 4.2.



Gambar 4.2. Tampilan Jendela Inisialisasi

4.1.3.2 Jendela Permainan

Pada jendela permainan, pemain dapat memulai permainan dengan menekan tombol S. Untuk lebih jelas lihat Gambar 4.3.



Gambar 4.3. Tampilan jendela permainan ketika permainan sedang berlangsung

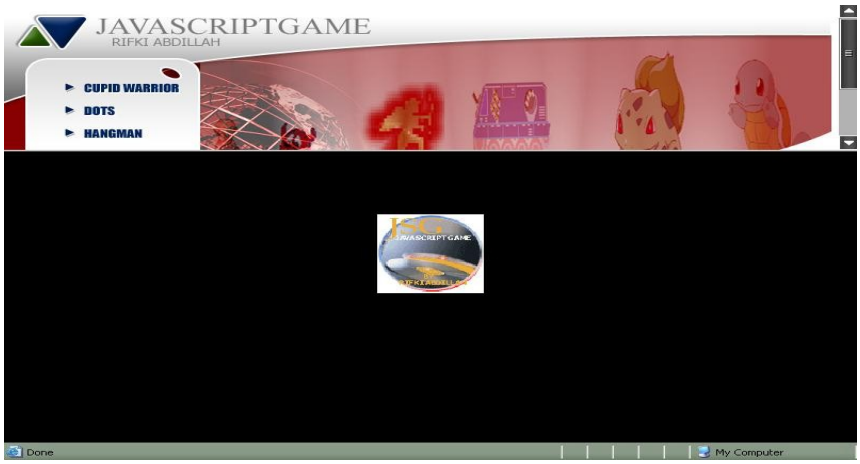
4.1.4 Jendela Permainan Pesawat udara Setelah Diimplementasikan Pada Situs Web.

Setelah rancangan permainan pesawat udara selesai diimplementasikan dengan menggunakan Javascript versi 1.3, selanjutnya permainan pesawat udara tersebut diimplementasikan pada sebuah situs web.

4.1.4.1 Jendela Halaman Pertama Situs Web

Pada antarmuka halaman pertama situs web terdapat beberapa pilihan permainan, yaitu petualangan, ular dan menembak. Ketika pengguna memilih salah satu permainan yang ada pada situs web tersebut, maka di sebelah kiri atas situs web

akan ditampilkan spesifikasi dari permainan yang dipilih. Pada halaman spesifikasi terdapat teks yang menghubungkan pengguna dengan halaman permainan. Untuk lebih jelas mengenai antarmuka halaman pertama situs web ini lihat Gambar 4.4.



Gambar 4.4. Jendela halaman pertama situs web

4.1.4.2 **Jendela Halaman Permainan**

Pada antarmuka halaman permainan ini, permainan yang telah dipilih pada menu pilihan permainan akan ditampilkan di sebelah kiri atas situs web, sehingga dapat dimainkan oleh pengguna. Untuk lebih jelas lihat Gambar 4.5.

Gambar 4.5. Jendela halaman permainan pada situs web

4.2 Pengujian

Pengujian dilakukan untuk menguji dan memastikan bahwa aplikasi permainan pesawat udara yang tersebut dapat berfungsi dengan baik dan sesuai dengan rumusan yang telah ditetapkan sebelumnya. Tanpa adanya pengujian, maka tidak dapat diketahui apakah aplikasi yang telah dibuat sesuai dengan spesifikasinya.

Kebenaran dari aplikasi permainan komputer dua dimensi dengan karakter pesawat udara adalah sebagai berikut:

1. Kemampuan untuk memulai permainan dengan menekan tombol S.
2. Kemampuan untuk menggerakkan pesawat Cupid ke depan / maju dengan menekan angka 6 atau tombol karakter L pada papan ketik.
3. Kemampuan untuk menggerakkan mobil ke belakang / mundur dengan menekan angka 4 atau tombol karakter J pada papan ketik.
4. Kemampuan untuk menggerakkan mobil ke atas dengan menekan angka 8 atau tombol karakter I pada papan ketik.
5. Kemampuan untuk menggerakkan mobil ke bawah dengan menekan angka 2 atau tombol karakter K pada papan ketik.
6. Kemampuan untuk menambahkan poin ketika pesawat pemain menghancurkan musuh.
7. Kemampuan untuk menghentikan permainan dengan menekan tombol F2.
8. Kemampuan untuk memulai kembali permainan dengan menekan tombol karakter S.

4.2.1 Metode Pengujian

Pengujian dilakukan agar dapat diketahui apakah aplikasi permainan komputer dengan karakter pesawat udara dapat berjalan sesuai dengan kebutuhan hasil analisis pada bab III. Terdapat 2 metode pengujian yang dapat dilakukan, seperti :

- Metode *White Box Testing* yaitu metode desain yang menggunakan struktur kontrol desain prosedural agar perekayasa sistem dapat melakukan *test case* yang : (1) memberikan jaminan bahwa semua jalur *independen* pada suatu modul telah digunakan paling tidak sati kali, (2) menggunakan semua keputusan logis pada sisi *true* dan *false*, (3) mengeksekusi semuan loop pada batasan mereka dan pada batas operasional mereka, (4) menggunakan struktur data intrnal untuk menjaminvaliditasnya.
- Metode *Black Box Testing* yaitu Pengujian untuk menemukan kesalahan dalam lingkup kategori sebagai berikut: (1) fungsi-fungsi yang tidak benar atau hilang, (2) kesalahan Antarmuka, (3) kesalahan dalam struktur data atau akses basis data, dan (4) kesalahan kinerja (Pressman, 2002:551-552).

Dalam melakukan pengujian pada aplikasi permainan tersebut dilakukan berdasarkan metode *Black Box Testing.*

Pengujian-pengujian yang akan dilakukan terdiri atas skenario pengujian, hasil pengujian, dan analisis hasil pengujian yang akan dijelaskan pada subbab berikut ini.

4.2.2 Skenario Pengujian

Skenario pengujian digunakan untuk menentukan langkah-langkah dalam melakukan pengujian. Pengujian dilakukan dengan menjalankan aplikasi permainan

terlebih dahulu. Kemudian dilakukan pengujian dengan menekan tombol-tombol yang terdapat pada jendela-jendela dari aplikasi permainan tersebut apakah sesuai dengan yang diharapkan. Selanjutnya pengujian dilakukan ketika permainan sedang berlangsung.

Skenario pengujian untuk aplikasi permainan komputer dua dimensi dengan karakter pesawat udara yang lebih lengkap dapat dilihat pada Tabel 4.1.

Tabel 4.1. Tabel skenario pengujian

No	Antarmuka yang diuji	Bagian dari Antarmuka yang diuji	Status permainan	Skenario pengujian	Hasil yang diharapkan
1.	Jendela permainan	Tombol untuk memulai permainan	Permainan belum dimulai	Tekan tombol (*S*) untuk memulai permainan	Permainan dimulai.
2.	Jendela permainan	-	Permainan telah dimulai	Pemain menekan angka 2,6,4,8 atau tombol j,l,k,I pada papan ketik	Pesawat Cupid akan bergerak ke arah yang sesuai dengan penekanan pada tombol panah papan ketik.
3.	Jendela permainan	-	Permainan telah dimulai	Pemain menekan tombol karakter Z atau spasi dan O atau Xpada papan ketik	Pesawat Cupid akan mengeluarkan peluru.
4.	Jendela permainan	-	Permainan telah dimulai	Arahkan peluru pesawat Cupid ke pesawat musuh	Bila peluru pesawat Cupid mengenai musuh, maka musuh tersebut akan meledak dan akan terdengar suara ledakan.
5.	Jendela permainan	-	Permainan telah dimulai	Dekatkan pesawat Cupid dengan peluru musuh	Bila pesawat Cupid terkena peluru musuh, dapat menyebabkan pesawat Cupid hancur.

Tabel 4.1. (lanjutan) Tabel skenario pengujian

No	Antarmuka yang diuji	Bagian dari Antarmuka yang diuji	Status permainan	Skenario pengujian	Hasil yang diharapkan
6.	Jendela permainan	-	Permainan telah dimulai	Dekatkan pesawat Cupid dengan pesawat musuh	Bila pesawat Cupid bertubrukan dengan pesawat musuh, pesawat musuh dan pesawat Cupid tersebut akan hancur.
7.	Jendela permainan	-	Permainan telah dimulai	Dekatkan pesawat Cupid dengan tanah	Bila pesawat Cupid terkena tanah dapat menyebabkan pesawat Cupid hancur.
8.	Jendela permainan	-	Permainan telah dimulai	Pemain menekan tombol F2 pada papan ketik	Permainan akan berakhir dan musik pengiring akan berhenti.
9.	Jendela permainan	Tombol untuk memulai kembali permainan	Permainan berakhir	Klik tombol untuk memulai permainan kembali (*S*)	Bila pesawat Cupid hancur ditengah pertempuran dan permainan berakhir, akan ditampilkan tombol untuk memulai permainan kembali. Bila pemain menekan tombol tersebut, maka permainan akan dimulai kembali sesuai dengan level terakhir yang sedang dimainkan.

4.2.3 Hasil Pengujian

Setelah aplikasi permainan komputer dengan karakter pesawat tempur tersebut selesai diuji berdasarkan skenario pengujian pada Tabel 4.1, maka dapat diperoleh hasil pengujian dari aplikasi permainan tersebut. Untuk lebih jelas lihat Tabel 4.2.

Tabel 4.2. Tabel hasil pengujian

No	Antarmuka yang diuji	Bagian dari antarmuka yang diuji	Status permainan	Hasil pengujian
1.	Jendela permainan	Tombol untuk memulai permainan	Permainan belum dimulai	Permainan dapat dimulai
2.	Jendela permainan	-	Permainan telah dimulai	Pesawat Cupid dapat bergerak ke arah yang sesuai dengan penekanan angka 2,6,4,8 atau tombol j,l,k,I pada papan ketik.
3.	Jendela permainan	-	Permainan telah dimulai	Pesawat Cupid dapat mengeluarkan peluru.
4.	Jendela permainan	-	Permainan telah dimulai	Ketika peluru pesawat Cupid terkena pesawat musuh, pesawat musuh tersebut meledak.
5.	Jendela permainan	-	Permainan telah dimulai	Ketika pesawat Cupid terkena peluru musuh, pesawat Cupid akan meledak
6.	Jendela permainan	-	Permainan telah dimulai	Ketika pesawat Cupid bertubrukan dengan pesawat musuh, pesawat musuh dan pesawat cupid akan hancur.
7.	Jendela permainan	-	Permainan telah dimulai	Ketika pesawat Cupid bertubrukan dengan tanah, pesawat Cupid akan hancur.
8.	Jendela permainan	-	Permainan telah dimulai	Permainan dapat berakhir, aplikasi berhenti dijalankan, dan musik pengiring berhenti dimainkan.
9.	Jendela permainan	Tombol untuk memulai kembali permainan	Permainan berakhir	Pemain dapat memulai permainan kembali sesuai dengan level terakhir yang sedang dimainkan.

4.2.4 Analisis Hasil Pengujian

Setelah dilakukan pengujian secara menyeluruh terhadap aplikasi permainan komputer dengan karakter pesawat udara tersebut, didapatkan beberapa keterbatasan dari aplikasi permainan tersebut yaitu sebagai berikut:

1. Suara efek ketika peluru pesawat Cupid ditembakkan dan suara efek ketika terjadi ledakan tidak dapat berbunyi secara bersama-sama. Solusinya adalah ketika menembakkan peluru, tekan tombol karakter Z pada papan ketik secara bertahap. Sehingga ketika peluru terkena pesawat musuh dan terjadi ledakan, suara ledakan akan terdengar.
2. Setelah selesai dengan pengkodean program pemain menguji program yang telah dibuat dengan Internet Explorer 5.0 dan Opera 6.0 secara offline (tanpa terhubung dengan internet). Penulis mendapat hasil bahwa program tersebut dapat berjalan di kedua perangkat lunak tersebut. Kemudian penulis menguji coba program dengan menggunakan perangkat lunak Netscape 4.0 dan penulis mendapat hasil bahwa program tersebut tidak dapat berjalan di Netscape 4.0. Karena program ActiveX pada Netscape yang dapat mengkonfigurasikan variabel-variabel pada aplikasi yang penulis buat tidak dapat dijalankan (disable) dan tidak dapat dirubah, serta tidak terdapatnya Javascript console di aplikasi Netscape.

BAB V

PENUTUP

5.1 Kesimpulan

Setelah aplikasi permainan komputer dengan karakter pesawat udara selesai dikembangkan dan diuji, maka dapat diambil beberapa kesimpulan sebagai berikut:

1. Aplikasi permainan komputer dengan karakter pesawat udara yang telah dikembangkan hanya dapat dijalankan pada komputer bersistem operasi Microsoft Windows.
2. Pengembangan yang dibuat oleh penulis adalah dengan menambahkan musik pengiring dan musuh.
3. Program permainan tidak dapat berjalan di Netscape 4.0. Karena program ActiveX pada Netscape yang dapat mengkonfigurasikan variabel-variabel pada aplikasi yang penulis buat tidak dapat dijalankan (disable) dan tidak dapat dirubah, serta tidak terdapatnya Javascript console di aplikasi Netscape.

5.2 Saran

Aplikasi permainan yang telah dikembangkan dirasakan masih memiliki beberapa kekurangan. Penulis menjadikan kekurangan tersebut sebagai saran untuk pengembangan aplikasi selanjutnya di masa yang akan datang. Beberapa kekurangan tersebut adalah sebagai berikut:

1. Aplikasi permainan komputer dengan karakter pesawat udara tersebut hanya dapat dimainkan oleh satu pemain. Disarankan untuk mengembangkan

permainan tersebut agar dapat dimainkan oleh dua pemain atau banyak pemain (*multiplayer*).

2. Pesawat Cupid (pesawat pemain) tidak dapat mengeluarkan senjata berjenis bomb. Disarankan untuk melengkapi pesawat Cupid dengan bomb yang dapat digunakan untuk menghancurkan seluruh musuh yang sedang muncul di layar permainan secara serentak.
3. Bila pesawat Cupid tertembak, maka pesawat Cupid langsung hancur. Disarankan untuk tidak langsung hancur apabila pesawat Cupid tersebut terkena peluru musuh atau bertabrakan dengan musuh dan tanah dengan catatan bahwa kekuatan pesawat Cupid bisa di tambah.
4. Dapat menambahkan batas waktu bermain pada permainan pesawat udara agar pemain dapat berlomba-lomba untuh mendapat score yang tertinggi (*High Score*).

DAFTAR PUSTAKA

Erwin Philipus, 1*2 Game Dengan Javascript*, Yogyakarta: Penerbit Andi 2004.

Jeffrey Manurung, *Bermain Dengan Javascript* , Jakarta: Penerbit Megindo, 2002.

Lani Sidharta, *Belajar Sendiri JavaScript* , Jakarta: Penerbit PT Elex Media, 1997.

M. Shalahuddin, Rosa A. S, *JAVA DI WEB,* Bandung: Penerbit Informatika, 2008

Munawar. 2005. *Pemodelan Visual dengan UML*. Yogyakarta: Penerbit Graha Ilmu.

Pressman, Roger S. 2002. *Rekayasa Perangkat Lunak Buku 1.* Yogyakarta: Penerbit Andi.

LAMPIRAN 2

KODE PROGRAM

Tag-tag Yang Digunakan Dalam HTML

Tag-tag Dasar	
<html></html>	Membuat dokumen HTML
<head></head>	Menetapkan judul dan infomasi lain yang tidak ditampilkan di browser
<body></body>	Menetapkan tampilan dokumen di browser

Tag Header	
<title></title>	Menamai dokumen HTML

Atribut Body	
<body bgcolo=?>	Menetapkan warna latar, yaitu menggunakan nama warna atau kode heksadesimal
<Body text=?>	Menetapkan warna text, yaitu menggunakan nama warna atau kode heksadesimal
<Body link=?>	Menetapkan warna link, yaitu menggunakan nama warna atau kode heksadesimal
<Body vlink=?>	Menetapkan warna link yang mengikutinya, yaitu menggunakan nama warna atau kode heksadesimal

<Body alink=?>	Menetapkan warna link sewaktu diklik

Tag Teks

<pre></pre>	Membuat text tanpa format huruf
	Membuat headline (h1-h6)
	Text dicetak tebal
	Text dicetak miring
<tt></tt>	Membuat teletype atau text bergaya huruf mesin ketik
<cite></cite>	Membuat kutipan, biasanya cetak miring
	Memperjelas kata (cetak miring atau tebal).
	Memperjelas kata (cetak miring atau tebal).
<font size=?><font>	Menetapkan ukuran huruf (1 – 7)
<font color=?><font>	Menetapkan warna huruf, yaitu nilainya berupa nama atau kode heksadesimal

Tag Link

< a href=”URL”>	Membuat hyperlink
	Membuat link mail to
	Membuat target lokasi dalam satu dokumen
	Terhubung ke suatu target lokasi dalam satu dokumen

Tag Format	
	Membuat paragraf baru
	Mengatur letak paragraf: kanan, kiri, tengah
 	Baris baru
<blockquote></blockquote>	Text indent dua sisi
<dl></dl>	Membuat list definisi
<dt>	Mendahului tiap term definisi
<dd>	mendahului tiap definisi
	Membuat numbered list
	Mendahului tiap item dari suatu list dan menambahkan angka
	membuat bullet list
	tag generik yang digunakan untuk memformat suatu blok besar HTML, juga digunakan untuk sylesheet

Elemen Grafis	
<img src=nama”>	Mengatur letak gambar: kiri, kanan, tengah; bawah, atas, tengah
<img src=”nama”align=?>	menetapkan ukuran border gambar
<hr>	Membuat garis lurus horizontal
<hr size=?>	Menetakan tinggi garis lurus
<hr width=?>	Menetapkan lebar garis lurus
<hr noshade>	Membuat garis lurus tanpa shadow

Tabel	
<table></table>	Membuat Tabel
<tr></tr>	Menetapkan tiap baris dalam tabel
<td></td>	Menetapkan tiap sel dalam baris
<th></th>	Menetapkan header tabel

Atribut Tabel	
<table border=#>	Menetapkan tebal border tabel
<table cellspacing=#>	Menetapkan jumlah spasi antarsel dalam tabel
<table cellpadding=#>	Menetapkan jumlah spasi antara border dan isinya
<table width=# atau %>	menetapkan lebar didalam satuan pixel atau persentase
<tr align=?> atau <td valign=?>	menetapkan rata vertikan untuk sel (atas, tengah, atau bawah >
<td colspan=#>	Menetapkan jumlah baris dari suatu sel yang harus diperpanjang
<td rowspan=#>	Menetapkan jumlah baris dari suatu sel yang harus diperpanjang
<td nowrap>	Mencegah pemisahan satu baris

Frame	
<frameset></frameset>	Mengganti tag <body> dalam dokumen yang terdiri dari banyak frame; bisa juga diletakan dalam frameset lain

<frameset rows=”nilai,nilai”>	Menentukan banyak baris dalam frameset, lebarnya menggunakan angka dalam satuan pixel atau persentasi
<frameset cols=”nilai,nilai”>	Menentukan banyak kolom dalam frameset, lebarnya menggunakan angka dalam satuan pixel atau persentasi
<frame>	Menetukan frame tunggal – atau wilayah – di dalam sebuat frameset
<noframe></noframe>	menetapkan apa ayng harus ditampilkan pada browser yang tidak mendukung tag frame

Atribut Frame	
<frame src=”URL”>	Menetukan dokumen HTML yang harus ditampilkan
<frame name=”name”>	Memberi nama frame, atau wilayah, sehingga bisa menjadi target oleh frame lain
<frame marginwidth=#>	Menetapkan frame margin kanan dan kiri; nilainya harus sama dengan atau lebih dari 1
<frmae marginheight=#>	Menetapkan frame margin atas dan bawah; nilainya harus samam dengan atau lebih dari 1

Form	
<form></form>	Membuat semua form
<option>	Menetapkan tiap item di menu

<textarea name=”NAMA”cols=40 rows=8></select>	Membuat kotak text. Kolom menyatakan lebar, dan baris menyatakan tinggi
<input type=”checkbox” name=”NAMA”>	Membuat checkbox. Text mengikuti tag
<input type=”radio” name=”NAMA” value=”x”>	Membuat tombol radio. Text mengikuti tag
<input type= “text” name=”saya” size=20>	Membuat text area satu baris, menetapkan panjang karakternya
<input type=”submit value=”NAMA”>	Membuat tombol submit, Text mengikuti tag
<input type=”image” border=0 name=”NAMA” src=”nama.gif”>	Membuat tombol submit dengan menggunakan image
<input type=”reset”>	Membuat tombol reset

LAMPIRAN 2

KODE PROGRAM

Kode Program

```
<HTML>
<HEAD>
<SCRIPT LANGUAGE="JavaScript">
<!-- Begin
var Flimit=9
var BossTM=0
var BossLM=700
var Easy=0
var timerID = null
var TimeCounter = 0
var INT = 25
var k=0
var X=100
var Y=100
var DX=0
var DY=0
var IX=0
var IY=0
var SDX=0
var SDY=0
var SIX=0
var SIY=0
var Endflg=0
var ff=0
var fc=0
var FX = new Array(12)
var FY = new Array(12)
var bf=0
var bc=0
var BX=0
var BY=-100
var BDX=0
var BDY=0
var Cname = new Array(2)
var Bflag=0
var GND=0
//var MTN=0
var EFX = new Array(8)
var EFY = new Array(8)
var EVX = new Array(8)
var EVY = new Array(8)
var Fmax = 3
var EMX = 200
var EMY = 284
var PTN = 0
var Mvx = 0
var Mvy = 0
var mCount = 100
var EX = new Array(20)
var EY = new Array(20)
var CH = new Array(2)
var CW = new Array(2)
var tmp1 = 0
var tmp2 = 0
```

```
var Endflg = 1
var Bossflg = 0
var BossX = 0
var BossH = 0
var CanonX=-200
var CanonY=-200
var CanonC=0
var Bossflg=0
var tmpBF=0
var sf=0
var spCOUNT=0

// // // // // // // // // // // // // // // // // // // // // // // // //
        systm=""
        ver=navigator.appVersion
        len=ver.length
        for(iln=0;iln<len;iln++) if(ver.charAt(iln)=="(") break
        systm = ver.charAt(iln+1).toUpperCase()
// // // // // // // // // // // // // // // // // // // // // // // // //
function interval1()
{clearTimeout(timerID)
ccount0=ccount0+1
if (ccount0 >= 2){cla0.left=-100}
ccount1=ccount1+1
if (ccount1 >= 2){cla1.left=-100}
ccount2=ccount2+1
if (ccount2 >= 2){cla2.left=-100}
TimeCounter=TimeCounter+1
with (Math){Fmax=floor(TimeCounter/100)+3}
if (Fmax >= Flimit){Fmax=Flimit}
if (TimeCounter == Blimit){Bossflg=1;tmpbgc=100;tmpbgf=0}
if (Bossflg >= 1){tmpbgc=tmpbgc-5
if (tmpbgc >= 0){if (systm!="C"){}}
if (tmpbgf == 0){if (tmpbgc <= 0){if (systm!="C"){tmpbgf=1}else{tmpbgf=1}}}tmpBF=0
for (tmp1=0; tmp1<10; tmp1++){if (EX[tmp1] >= -50){tmpBF=1}}
if (tmpBF == 0){BossMode()}
GND=GND-3;if (GND <= -16){GND=0}GNDs.left=GND
//MTN=MTN-1;if (MTN <= -64){MTN=0}MTNs.left=MTN
with (Math) {for (tmp1=0; tmp1<3; tmp1++){EX[tmp1]=EX[tmp1]-5
if (EX[tmp1] <= -30){if (floor(random()*12) == 3 ){if (Bossflg == 0)

        {EX[tmp1]=400;EY[tmp1]=floor(random()*12)*20+20}}}spMOV(27+tmp1,EX[tmp1],EY[
tmp1])
with (Math){if (floor(random()*5) == 3 ){Efire(EX[tmp1],EY[tmp1])}}}}}
if (TimeCounter >= 300){with (Math) {for (tmp1=3; tmp1<6; tmp1++){EX[tmp1]=EX[tmp1]-
CW[tmp1-3]
if (EY[tmp1] <= 5){CH[tmp1-3]=2;CW[tmp1-3]=4}
if (EY[tmp1] >= 240){CH[tmp1-3]=-2;CW[tmp1-3]=2}EY[tmp1]=EY[tmp1]+CH[tmp1-3]
if (EX[tmp1] <= -30){if (floor(random()*12) == 3 ){if (Bossflg == 0)

        {EX[tmp1]=400;EY[tmp1]=floor(random()*12)*20+20}}}spMOV(27+tmp1,EX[tmp1],EY[
tmp1])
with (Math){if (floor(random()*5) == 3 ){Efire(EX[tmp1],EY[tmp1])}}}}}
if (TimeCounter >= 200){with (Math){tmpTM=TimeCounter-floor(TimeCounter/100)*100}
if (tmpTM <= 40){tmpTMX=3}else{if (tmpTM <= 70){tmpTMX=1}else{tmpTMX=-1}}
for (tmp1=0; tmp1<3; tmp1++){EX[tmp1+6]=EX[tmp1+6]-tmpTMX
if (EX[tmp1+6] <= -30){if (EX[6] <= 365){if (EX[7] <= 365){if (EX[8] <= 365){if (Bossflg ==
0){with (Math){if (floor(random()*40) == 3 ){EX[tmp1+6]=400}}}}}}}
```

```
                    spMOV(33+tmp1,EX[tmp1+6],EY[tmp1+6])
with (Math){if (floor(random()*5) == 3 ){Efire(EX[tmp1+6],EY[tmp1+6])}}}}
if (EX[9] <= -600){if (Bossflg == 0){EX[9]=400}}
                    EX[9]=EX[9]-1;if (EX[9] <= 250){EX[9]=EX[9]-4}
if (EX[9] == 365){EMX=EX[9]+7;EMY=EY[9]-10;mCount=0}
if (EX[9] == 310){EMX=EX[9]+7;EMY=EY[9]-10;mCount=0}
if (EX[9] == 255){EMX=EX[9]+7;EMY=EY[9]-10;mCount=0}
                    spMOV(36,EX[9],EY[9])
if (bf == 1){BX=X;BY=Y+30;bc=1;bf=0;BDX=12}
if (bc == 1){BDY=BDY+2
if (BDY == 10){spMOV(45+Bflag,-100,-100)
                    Bflag=1}
if (BDY >= 10){BDX=BDX+3;BX=BX+BDX}
                    BY=BY+BDY;tmpBMX=-100;tmpBMY=-100
for (tmp2=0; tmp2<10; tmp2++){if (BX >= EX[tmp2]-30){if (BX <= EX[tmp2]+16){if (BY >=
EY[tmp2]-3){
if (BY <= EY[tmp2]+16){tmpBMX=BX-8;tmpBMY=BY-4}}}}}
if (BY >= 284){tmpBMX=BX-8;tmpBMY=244}
if (tmpBMX != -100){spMOV(45+Bflag,-100,-100)
                    spMOV(50,tmpBMX,tmpBMY)
                    Bflag=0;ccount2=0;bc=0;BY=-100;BDX=0;BDY=0
for (tmp2=0; tmp2<10; tmp2++){if (BX >= EX[tmp2]-56){if (BX <= EX[tmp2]+40){if (EY[tmp2]
>= tmpBMY-18){if (EY[tmp2] <= tmpBMY+63){spMOV(33+tmp2,-100,-100)
                    spMOV(Cname[cflag],EX[tmp2],EY[tmp2]-6)
if (cflag == 0){ccount0=0}else{ccount1=0}
                    cflag=cflag+1;if (cflag >= 2){cflag=0}
                    EX[tmp2]=-100;FX[tmp1]=500;score=score+50}}}}}}
                    spMOV(45+Bflag,BX,BY)}
if (ff == 1){FX[fc]=X;FY[fc]=Y+12
                    fc=fc+1;if (fc >= 13){fc=0}
                    ff=0}
for (tmp1=0; tmp1<13; tmp1++){FX[tmp1]=FX[tmp1]+32
for (tmp2=0; tmp2<10; tmp2++){if (FX[tmp1] >= EX[tmp2]-8){if (FX[tmp1] <= EX[tmp2]+32){if
(FY[tmp1] >= EY[tmp2]-8){if (FY[tmp1] <= EY[tmp2]+19){spMOV(27+tmp2,-100,-100)
                    spMOV(Cname[cflag],EX[tmp2],EY[tmp2]-6)
if (cflag == 0){ccount0=0}else{ccount1=0}
                    cflag=cflag+1;if (cflag >= 2){cflag=0}
                    EX[tmp2]=-100;FX[tmp1]=500;score=score+10}}}}}
                    spMOV(tmp1+1,FX[tmp1],FY[tmp1])}
                    Efmove()
                    Emiss()
                    tmpIX=IX+SIX;if (tmpIX >= 1){tmpIX=5+Easy}
                    tmpIY=IY+SIY;if (tmpIY >= 1){tmpIY=5+Easy}
                    tmpDX=DX+SDX;if (tmpDX >= 1){tmpDX=5+Easy}
                    tmpDY=DY+SDY;if (tmpDY >= 1){tmpDY=5+Easy}
                    X=X+tmpIX-tmpDX
                    Y=Y+tmpIY-tmpDY
if (X >= 368){X=368}
if (X <=  0){X=  0}
if (Y <=  0){Y=  0}
if (Y >= 258){mycr.visibility="hidden"
                    DOC2c.visibility="hidden"
                    DOCc.visibility="visible"
                    spMOV(Cname[cflag],X,Y)
                    Endflg=1;EFX[tmp1]=-100;cflag=cflag+1;if (cflag >= 2){cflag=0}}
for (tmp2=0; tmp2<10; tmp2++){if (X >= EX[tmp2]-20){if (X <= EX[tmp2]+20){if (Y >=
EY[tmp2]-20){if (Y <= EY[tmp2]+10){
                    mycr.visibility="hidden"
```

```
                    DOC2c.visibility="hidden"
                    DOCc.visibility="visible"
                    spMOV(Cname[cflag],X,Y)
                    Endflg=1;EFX[tmp1]=-100;cflag=cflag+1;if (cflag >= 2){cflag=0}}}}}}
                    spMOV(0,X,Y)
with (Math) {s5=floor(score/10000)
                    s4=floor((score-s5*10000)/1000)
                    s3=floor((score-s5*10000-s4*1000)/100)
                    s2=floor((score-s5*10000-s4*1000-s3*100)/10)
                    s1=score-s5*10000-s4*1000-s3*100-s2*10}
                    L7c.top=-16*s5
                    L8c.top=-16*s4
                    L9c.top=-16*s3
                    L10c.top=-16*s2
                    L11c.top=-16*s1
if (Endflg != 1){timerID = setTimeout("interval1()",INT)}else{if (systm!="C"){}else{}
if (hscore <= score){document.cookie="JSFALCON="+score+"; expires=Fri. 31-Dec-1999 23:59:59
GMT"
                    hscore=score
                    loadSCORE()}}}
// // // // // // // // // // // // // // // // // // // // // // // // // //
function onLD()
{
if(systm!="C"){W1c=document.layers["W1"]
          mycr=W1c.layers["Mychr"]
                    cla0=W1c.layers["clash"]
                    cla1=W1c.layers["clash1"]
                    cla2=W1c.layers["clash2"]
                    GNDs=W1c.layers["GND"]
//                  MTNs=Wlc.layers["MTN"]
                    DOCc=W1c.layers["doc"]
                    DOC2c=W1c.layers["doc2"]
                    EAS=W1c.layers["EASY"]
                    BOS0=W1c.layers["boss0"]
                    BOS1=W1c.layers["boss1"]
                    BOSR=W1c.layers["bossR"]
                    L2c=document.layers[2]
                    L3c=document.layers[3]
                    L4c=document.layers[4]
                    L5c=document.layers[5]
                    L6c=document.layers[6]
                    L7c=document.layers[8]
                    L8c=document.layers[9]
                    L9c=document.layers[10]
                    L10c=document.layers[11]
                    L11c=document.layers[12]}
                    else
                    {W1c=document.all.W1.style
                    mycr=document.all.W1.document.all.Mychr.style
                    cla0=document.all.W1.document.all.clash.style
                    cla1=document.all.W1.document.all.clash1.style
                    cla2=document.all.W1.document.all.clash2.style
                    GNDs=document.all.W1.document.all.GND.style
//          MTNs=document.all.W1.document.all.MTN.style
                    DOCc=document.all.W1.document.all.doc.style
                    DOC2c=document.all.W1.document.all.doc2.style
                    EAS=document.all.W1.document.all.EASY.style
                    BOS0=document.all.W1.document.all.boss0.style
```

```
                    BOS1=document.all.W1.document.all.boss1.style
                    BOSR=document.all.W1.document.all.bossR.style
                    L2c=L2I.style
                    L3c=L3I.style
                    L4c=L4I.style
                    L5c=L5I.style
                    L6c=L6I.style
                    L7c=L7I.style
                    L8c=L8I.style
                    L9c=L9I.style
                    L10c=L10I.style
                    L11c=L11I.style
                    W1c.clip="rect(0,400,400,0)"}
                    loadSCORE()
                    sf=1
                    QuitPlay()}
// // // // // // // // // // // // // // // // // // // // // // //
function EASY()
{
                    Easy=1
                    EAS.visibility="visible"}
// // // // // // // // // // // // // // // // // // // // // // //
function NORMAL()
{
                    Easy=0
                    EAS.visibility="hidden"}
// // // // // // // // // // // // // // // // // // // // // // //
function BossMode()
{
if (Bossflg == 1){BossX=BossX-10
if (BossX==250){Bossflg=3
if (Flimit != 0){EMX=BossX+30;EMY=218;mCount=0}
          EVX[0]=-5;EVY[0]=-5
                    EVX[1]=-6;EVY[1]=-4
                    EVX[2]=-4;EVY[2]=-6
                    EVX[3]=-2;EVY[3]=-7
                    EVX[8]=-7;EVY[8]=-7
                    EFX[0]=BossX+101;EFY[0]=225
                    EFX[1]=BossX+98;EFY[1]=229
                    EFX[2]=BossX+104;EFY[2]=222
                    EFX[3]=BossX+109;EFY[3]=222
                    EFX[8]=BossX+7;EFY[8]=232}
if (BossX==0){Bossflg=2}}
if (Bossflg == 2){BossX=BossX+10
if (BossX==150){Bossflg=6
                    EVX[4]=-5;EVY[4]=-5
                    EVX[5]=-6;EVY[5]=-4
                    EVX[6]=-4;EVY[6]=-6
                    EVX[7]=-2;EVY[7]=-7
                    EVX[8]=-7;EVY[8]=-7
                    EFX[4]=BossX+101;EFY[4]=225
                    EFX[5]=BossX+98;EFY[5]=229
                    EFX[6]=BossX+104;EFY[6]=222
                    EFX[7]=BossX+109;EFY[7]=222
                    EFX[8]=BossX+7;EFY[8]=232}
if (BossX==270){Bossflg=1}}
                    BOS0.left=BossX
if (Bossflg == 3){BossH=BossH-5
```

```
if (BossH == 170){Bossflg=4;CanonC=0}}
if(Bossflg==4){
if(CanonC==0){CanonX=BossX+55;CanonY=195;CanonC=1}
else
{CanonC=CanonC+1;if (CanonC==10){Bossflg=5}}}
if (Bossflg == 5){BossH=BossH+5
if (BossH == 220){Bossflg=1}}
if (Bossflg == 6){BossH=BossH-5
if (BossH == 170){Bossflg=7;CanonC=0}}
if(Bossflg==7){if (CanonC==0){CanonX=BossX+55;CanonY=195;CanonC=1}
else
{CanonC=CanonC+1;if (CanonC==10){Bossflg=8}}}
if (Bossflg == 8){BossH=BossH+5
if (BossH == 220){Bossflg=2}}
if (BossH == 220){tmpBHx=-200}
                    else
                    {tmpBHx=BossX+30}
                    CanonX=CanonX-25;CanonY=CanonY-25
                    spMOV(15,tmpBHx,BossH)
                    spMOV(14,CanonX,CanonY)
                    Fmax=Flimit;Efmove()
for (tmp2=0; tmp2<10; tmp2++){if (X >= BossX-20){if (X <= BossX+124){if (Y >= 188){
                    mycr.visibility="hidden"
                    DOC2c.visibility="hidden"
                    DOCc.visibility="visible"
                    spMOV(Cname[cflag],X,Y)
                    Endflg=1;cflag=cflag+1;if (cflag >= 2){cflag=0}}}}}
for (tmp2=0; tmp2<10; tmp2++){if (X >= BossX+20){if (X <= BossX+105){if (Y >= BossH-16){
                    mycr.visibility="hidden"
                    DOC2c.visibility="hidden"
                    DOCc.visibility="visible"
                    spMOV(Cname[cflag],X,Y)
                    Endflg=1;cflag=cflag+1;if (cflag >= 2){cflag=0}}}}}
                    Cchkflg=0
if(X>=CanonX-16){if(X<=CanonX+16){if(Y>=CanonY-16){if(Y<= CanonY+16){Cchkflg=1}}}}
if(X>=CanonX){if(X<=CanonX+34){if(Y>=CanonY){if(Y<= CanonY+34){Cchkflg=1}}}}
if (Cchkflg == 1){mycr.visibility="hidden"
                    DOC2c.visibility="hidden"
                    DOCc.visibility="visible"
                    spMOV(Cname[cflag],X,Y)
                    Endflg=1;cflag=cflag+1;if (cflag >= 2){cflag=0}}
                    spMOV(47,-100,-100)
for (tmp1=0; tmp1<13; tmp1++){if (FX[tmp1] >= BossX-8){if (FX[tmp1] <= BossX+128){if
(FY[tmp1] >= 216){FX[tmp1]=500
                    spMOV(47,BossX,FY[tmp1])}}}}
for (tmp1=0; tmp1<13; tmp1++){if (FX[tmp1] >= BossX+38){if (FX[tmp1] <= BossX+70){if
(FY[tmp1] >= BossH-4){if (FY[tmp1] < 216){FX[tmp1]=500
                    spMOV(47,BossX+38,FY[tmp1])}}}}}
                    tmpBMX=-100
if (BX >= BossX-30){if (BX <= BossX+144){if (BY >= 216){tmpBMX=BX-8;tmpBMY=BY-4}}}
if (BX >= BossX+2){if (BX <= BossX+94){if (BY >= BossH-4){tmpBMX=BX-8;tmpBMY=BY-
4}}}
if (BY >= 284){tmpBMX=BX-8;tmpBMY=244}
if (tmpBMX != -100){spMOV(45+Bflag,-100,-100)
                    spMOV(50,tmpBMX,tmpBMY)
                    Bflag=0;ccount2=0;bc=0;BY=-100;BDX=0;BDY=0
if (BossH != 220){if (tmpBMX >= BossX+25){if (tmpBMX <= BossX+108){if (tmpBMY <=
BossH+22){if (tmpBMY <= 219){
```

```
                    DOC2c.visibility="visible"
                    BOS1.left=-200
                    spMOV(16,tmpBHx,BossH)
                    Endflg=1;score=score+Flimit*1000}}}}}}}
// // // // // // // // // // // // // // // // // // // // // // // //
function QuitPlay()
{
focus()
                    Endflg=1
                    DOC2c.visibility="hidden"
                    DOCc.visibility="visible"
                    clearTimeout(timerID)}
// // // // // // // // // // // // // // // // // // // // // // // //
function Restart()
{
                    DOCc.visibility="hidden"
                    DOC2c.visibility="hidden"
for (tmp1=0; tmp1<50; tmp1++){spLEFT(tmp1,-200)}
for (tmp1=0; tmp1<10; tmp1++){EX[tmp1]=0;EY[tmp1]=0}
if (systm!="C"){}
                    else{}
                    TimeCounter = 0
                    score=0
                    cflag=0
                    Bossflg=BossTM
                    BossX=400
                    BossH=220
                    Blimit=BossLM
                    CanonX=-200
                    CanonY=-200
                    CanonC=0
                    X=100
                    Y=100
                    DX=0
                    DY=0
                    IX=0
                    IY=0
                    SDX=0
                    SDY=0
                    SIX=0
                    SIY=0
                    Endflg=0
                    ff=0
                    fc=0
                    ccount0=0
                    ccount1=0
                    ccount2=0
for (tmp1=0; tmp1<13; tmp1++){FX[tmp1]=0;FY[tmp1]=-100}
                    bf=0
                    bc=0
                    BX=0
                    BY=-100
                    BDX=0
                    BDY=0
                    Bflag=0
                    GND=0
//          MTN=0
                    Mvx=0
```

```
                    Mvy=0
                    mCount=100
                    Fmax = 3
for (tmp1=0; tmp1<9; tmp1++){EFX[tmp1]=-100;EFY[tmp1]=-100
                    spMOV(tmp1+18,EFX[tmp1],EFY[tmp1])}
                    tmpE0 = 0
for (tmp1=0; tmp1<3; tmp1++){with (Math) {EX[tmp1] = 460 + floor(random()*200)}
with (Math) {EY[tmp1] = floor(random()*12)*20+20}}
                    EX[3]=400;EX[4]=440;EX[5]=520
                    EY[3]=100;EY[4]=60 ;EY[5]=200
                    CH[0]=2  ;CH[1]=-2 ;CH[2]=-2
                    CW[0]=4  ;CW[1]=2 ;CW[2]=2
for         (tmp1=0; tmp1<3; tmp1++){spMOV(tmp1+30,EX[tmp1],EY[tmp1])}
                    EX[6]=400;EX[7]=496;EX[8]=530;EX[9]=500
for (tmp1=0; tmp1<4; tmp1++){EY[6+tmp1]=260
                    spMOV(tmp1+33,EX[tmp1],EY[tmp1])}
                    spMOV(45,-100,-100)
                    spMOV(46,-100,-100)
                    mycr.visibility="visible"
if (BossTM != 0){for (tmp1=0; tmp1<10; tmp1++){EX[tmp1]=-100;Blimit=5}}
                    timerID = setTimeout("interval1()",INT)}
// // // // // // // // // // // // // // // // // // // // // // // // // //
function Emiss()
{
mCount=mCount+1
if (mCount == 50){spLEFT(37+PTN,-100)
                    spMOV(Cname[cflag],EMX-4,EMY-4)
if (cflag == 0){ccount0=0}
                    else
                    {ccount1=0}
                    cflag=cflag+1;if (cflag >= 2){cflag=0}
if (X >= EMX-28){if (X <= EMX+24){if (Y >= EMY-28){if (Y <= EMY+24)
{
                    mycr.visibility="hidden"
                    DOC2c.visibility="hidden"
                    DOCc.visibility="visible"
                    spMOV(Cname[cflag],X,Y)
                    Endflg=1;EFX[tmp1]=-100;cflag=cflag+1;if (cflag >= 2){cflag=0}}}}}}
if (mCount < 50){tmpEMX=X-EMX;tmpEMY=EMY-Y
                        tmpPTN=PTN
if (tmpEMX == 0){tmpEMX=1}
if (tmpEMY == 0){tmpEMY=1}
                            tmpTx=tmpEMY/tmpEMX
if(tmpEMX>0){if(tmpTx> 2.4142){PTN=2;Mvx=Mvx/1.4;Mvy=Mvy+0.7}
else
{if (tmpTx > 0.4142){PTN=3;Mvx=Mvx+0.5;Mvy=Mvy+0.5}
else
{if (tmpTx > -0.4142){PTN=4;Mvx=Mvx+0.7;Mvy=Mvy/1.4}
else
{if (tmpTx > -2.4142){PTN=5;Mvx=Mvx+0.5;Mvy=Mvy-0.5}
else
{PTN=6;Mvx=Mvx/1.4;Mvy=Mvy-0.7}}}}}
else
{if (-tmpTx > 2.4142){PTN=2;Mvx=Mvx/1.4;Mvy=Mvy+0.7}
else
{if (-tmpTx > 0.4142){PTN=1;Mvx=Mvx-0.5;Mvy=Mvy+0.5}
else
{if (-tmpTx > -0.4142){PTN=0;Mvx=Mvx-0.7;Mvy=Mvy/1.4}
```

```
else
{if (-tmpTx > -2.4142){PTN=7;Mvx=Mvx-0.5;Mvy=Mvy-0.5}
else
{PTN=6;Mvx=Mvx/1.4;Mvy=Mvy-0.7}}}}}
if (mCount <= 5){PTN=1;Mvx=Mvx-0.5;Mvy=Mvy+0.5}
if (Mvx >= 5){Mvx=5}
if (Mvy >= 5){Mvy=5}
if (Mvx <=-5){Mvx=-5}
if (Mvy <=-5){Mvy=-5}
EMX=EMX+Mvx-1;EMY=EMY-Mvy
                        spLEFT(37+tmpPTN,-100)
                        spMOV(37+PTN,EMX,EMY)
if (X >= EMX-24){if (X <= EMX+20){if (Y >= EMY-24){if (Y <= EMY+20){
                        spLEFT(37+PTN,-100)
                        spMOV(Cname[cflag],EMX-4,EMY-4)
                        cflag=cflag+1;if (cflag >= 2){cflag=0}
                        mycr.visibility="hidden"
                        DOC2c.visibility="hidden"
                        DOCc.visibility="visible"
                        spMOV(Cname[cflag],X,Y)
                        cflag=cflag+1;if (cflag >= 2){cflag=0}
                        Endflg=1;EFX[tmp1]=-100}}}}}}
// // // // // // // // // // // // // // // // // // // // // // // // // //
function Efmove()
{
for (tmp1=0; tmp1<Fmax; tmp1++){if (EFX[tmp1] != -
100){EFX[tmp1]=EFX[tmp1]+EVX[tmp1];EFY[tmp1]=EFY[tmp1]+EVY[tmp1]
if (EFX[tmp1] < -16){EFX[tmp1]=-100;EVX[tmp1]=0;EVY[tmp1]=0}
if (EFX[tmp1] > 400){EFX[tmp1]=-100;EVX[tmp1]=0;EVY[tmp1]=0}
if (EFY[tmp1] < -16){EFX[tmp1]=-100;EVX[tmp1]=0;EVY[tmp1]=0}
if (EFY[tmp1] > 300){EFX[tmp1]=-100;EVX[tmp1]=0;EVY[tmp1]=0}}
if (EFX[tmp1] >= X-4){if (EFX[tmp1] <= X+26){if (EFY[tmp1] >= Y+8){if (EFY[tmp1] <= Y+24){
                        mycr.visibility="hidden"
                        DOC2c.visibility="hidden"
                        DOCc.visibility="visible"
                        spMOV(Cname[cflag],X,Y)
                        Endflg=1;EFX[tmp1]=-100;cflag=cflag+1;
                        if (cflag >= 2){cflag=0}}}}}
                        spMOV(tmp1+18,EFX[tmp1],EFY[tmp1])}}
// // // // // // // // // // // // // // // // // // // // // // // // // //
function Efire(EPX,EPY)
{
if (Bossflg == 0){tmpE0=tmpE0+1;if (tmpE0 >= 9){tmpE0=0}
if (EFX[tmpE0] == -100){EFX[tmpE0]=EPX;EFY[tmpE0]=EPY
                        tmpEx=X-EPX;tmpEy=Y-EPY
with (Math) {tmpEr=sqrt(tmpEx*tmpEx+tmpEy*tmpEy)
if (tmpEr == 0){tmpEr=1}
        EVX[tmpE0]=floor((tmpEx/tmpEr)*7);EVY[tmpE0]=floor((tmpEy/tmpEr)*7)}}}}
// // // // // // // // // // // // // // // // // // // // // // // // // //
functionloadSCORE() //menghitung score
{if (document.cookie){CUPIDtop = document.cookie.indexOf("JSCUPID", 0)
if(CUPIDtop!=-1)
{tscore=document.cookie.substring(CUPIDtop+9,CUPIDtop+14)}
else
{tscore="0"}}
else
{tscore="0"}
with (Math) {hscore=parseInt(tscore,10)
```

```
            h5=floor(hscore/100)
            h4=floor((hscore-h5*10)/100)
            h3=floor(hscore-h5*10-h4*10)/100))
            h2=floor(hscore-h5*10-h4*10-h3*10)/100))
            h1=hscore-h5*10-h4*10-h3*10-h2*10}
            L2c.top=-16,h5
            L3c.top=-16,h4
            L4c.top=-16,h3
            L5c.top=-16,h2
            L6c.top=-16,h1}
functionkeyDown(DnEvents)
{if (systm != "C"){k=DnEvents.which}
else
{k=window.event.keyCode}
  if (k == 50){IY=1}
  if (k == 56){DY=1}
  if (k == 54){IX=1}
  if (k == 52){DX=1}
  if (k == 98 ){IY=1}
  if (k == 104){DY=1}
  if (k == 102){IX=1}
  if (k == 100){DX=1}
  if (k == 75){IY=1}
  if (k == 73){DY=1}
  if (k == 76){IX=1}
  if (k == 74){DX=1}
  if (k == 107){IY=1}
  if (k == 105){DY=1}
  if (k == 108){IX=1}
  if (k == 106){DX=1}
  if (k == 49){SDX=1;SIY=1}
  if (k == 51){SIX=1;SIY=1}
  if (k == 55){SDX=1;SDY=1}
  if (k == 57){SIX=1;SDY=1}
  if (k == 32){ff=1}
  if (k == 90){ff=1}
  if (k == 122){ff=1}
  if (k == 0){ff=1}
  if (k == 48){if (bc == 0){bf=1}}
  if (k == 96){if (bc == 0){bf=1}}
  if (k == 13){if (bc == 0){bf=1}}
  if (k == 88){if (bc == 0){bf=1}}
  if (k == 120){if (bc == 0){bf=1}}
  if (k == 83){if (Endflg == 1){if (sf == 1){Restart()}}}
  if (k == 115){if (Endflg == 1){if (sf == 1){Restart()}}}
  if (k == 69){if (Endflg == 1){Flimit=3;EASY()}}
  if (k == 101){if (Endflg == 1){Flimit=3;EASY()}}
  if (k == 78){if (Endflg == 1){Flimit=9;NORMAL()}}
  if (k == 110){if (Endflg == 1){Flimit=9;NORMAL()}}
  if (k == 81){QuitPlay()}
  if (k == 113){QuitPlay()}}
// // // // // // // // // // // // // // // // // // // // // // // //
functionkeyUp(UpEvents)
{if (systm != "C"){k=UpEvents.which}
else
{k=window.event.keyCode}
  if (k == 50){IY=0}
  if (k == 56){DY=0}
```

```
   if (k == 54){IX=0}
   if (k == 52){DX=0}
   if (k == 98 ){IY=0}
   if (k == 104){DY=0}
   if (k == 102){IX=0}
   if (k == 100){DX=0}
   if (k == 75){IY=0}
   if (k == 73){DY=0}
   if (k == 76){IX=0}
   if (k == 74){DX=0}
   if (k == 107){IY=0}
   if (k == 105){DY=0}
   if (k == 108){IX=0}
   if (k == 106){DX=0}
   if (k == 49){SDX=0;SIY=0}
   if (k == 51){SIX=0;SIY=0}
   if (k == 55){SDX=0;SDY=0}
   if (k == 57){SIX=0;SDY=0}}
// // // // // // // // // // // // // // // // // // // // // //
function spINI(spNAME,spX,spY,spW,spH,spIMG)
{
if (systm!="C")
{document.layers["W1"].document.write("<LAYER NAME="'+spNAME+'" LEFT="+spX+"
TOP="+spY+"><IMG WIDTH="+spW+" HEIGHT="+spH+" SRC="'+spIMG+'"></LAYER>")}
else
{with(document){write("<IMG ID="+spNAME+" WIDTH="+spW+" HEIGHT="+spH+"
SRC="'+spIMG+'" STYLE='position:absolute'>")
images[spCOUNT+14].style.posLeft = spX
images[spCOUNT+14].style.posTop =  spY}}
spCOUNT=spCOUNT+1}
// // // // // // // // // // // // // // // // // // // // // //
functionspMOV(spNO,spX,spY)
{
if (systm!="C"){W1c.layers[spNO].moveTo(spX,spY)}
else
{document.images[spNO+14].style.posLeft = spX
document.images[spNO+14].style.posTop =  spY}}
// // // // // // // // // // // // // // // // // // // // // //
functionspLEFT(spNO,spX)
{
if(systm!="C"){W1c.layers[spNO].left=spX}
else
{document.images[spNO+14].style.posLeft = spX}}
document.onkeydown = keyDown
document.onkeyup = keyUp
if (systm!="C"){document.captureEvents(Event.KEYDOWN|Event.KEYUP)}
-->
</SCRIPT>
</HEAD>
<!-- //////////////////////////////////// -->
<BODY BGCOLOR="#00003F" TEXT="#FFFFFF" ONLOAD="onLD()">
<SCRIPT LANGUAGE="JavaScript">

<!--  Begin
var MSIE=navigator.userAgent.indexOf("MSIE");
var NETS=navigator.userAgent.indexOf("Netscape");
var OPER=navigator.userAgent.indexOf("Opera");
if((MSIE>-1) || (OPER>-1)) {
```

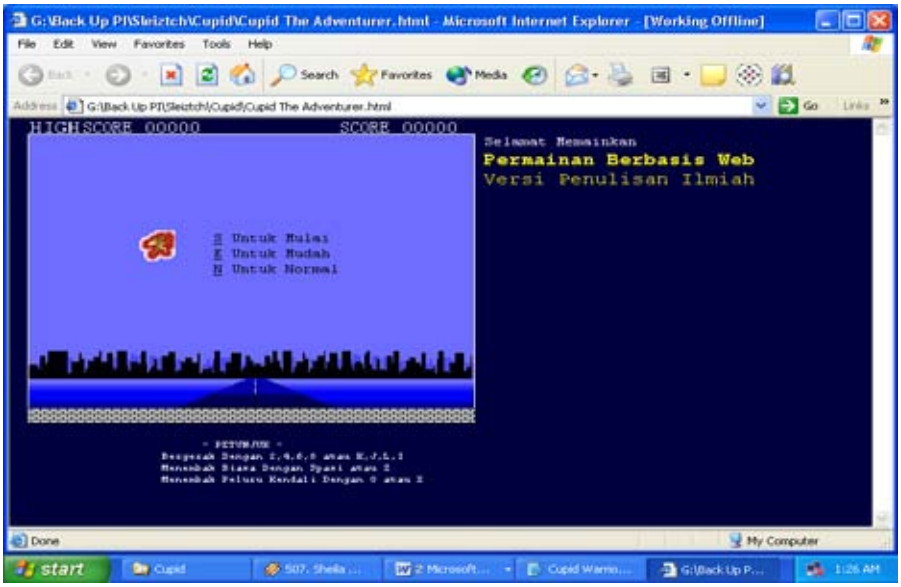
```
document.write("<BGSOUND SRC=sound.mid LOOP=INFINITE>");
}
else
{document.write("<EMBED SRC=sound.mid AUTOSTART=TRUE ");
document.write("HIDDEN=true VOLUME=100 LOOP=TRUE>");}
// End -->
</SCRIPT>

<DIV ID="BG"></DIV>

<DIV STYLE='position:absolute; left:16; top:0'><IMG WIDTH=50 HEIGHT=16 SRC="highc.gif"></DIV>

<DIV STYLE='position:absolute; left:60; top:0'><IMG WIDTH=50 HEIGHT=16 SRC="score.gif"></DIV>

<DIV ID="L2I" STYLE='position:absolute; left:120; top:0'><IMG WIDTH=16 HEIGHT=160 SRC="number.gif"></DIV>

<DIV ID="L3I" STYLE='position:absolute; left:130; top:0'><IMG WIDTH=16 HEIGHT=160 SRC="number.gif"></DIV>

<DIV ID="L4I" STYLE='position:absolute; left:140; top:0'><IMG WIDTH=16 HEIGHT=160 SRC="number.gif"></DIV>

<DIV ID="L5I" STYLE='position:absolute; left:150; top:0'><IMG WIDTH=16 HEIGHT=160 SRC="number.gif"></DIV>

<DIV ID="L6I" STYLE='position:absolute; left:160; top:0'><IMG WIDTH=16 HEIGHT=160 SRC="number.gif"></DIV>

<DIV STYLE='position:absolute; left:290; top:0'><IMG WIDTH=50 HEIGHT=16 SRC="score.gif"></DIV>

<DIV ID="L7I" STYLE='position:absolute; left:350; top:0'><IMG WIDTH=16 HEIGHT=160 SRC="number.gif"></DIV>

<DIV ID="L8I" STYLE='position:absolute; left:360; top:0'><IMG WIDTH=16 HEIGHT=160 SRC="number.gif"></DIV>

<DIV ID="L9I" STYLE='position:absolute; left:370; top:0'><IMG WIDTH=16 HEIGHT=160 SRC="number.gif"></DIV>

<DIV ID="L10I" STYLE='position:absolute; left:380; top:0'><IMG WIDTH=16 HEIGHT=160 SRC="number.gif"></DIV>

<DIV ID="L11I" STYLE='position:absolute; left:390; top:0'><IMG WIDTH=16 HEIGHT=160 SRC="number.gif"></DIV>

<DIV STYLE='position:absolute; left:16; top:16'><IMG WIDTH=400 HEIGHT=300 SRC="skybg.png"></DIV>

<DIV ID="TITL" STYLE='position:absolute; left:424; top:16'><PRE>Selamat Datang<FONTCOLOR="#FFFF00"SIZE="+2"><BR><B>
Cupid Warrior</B><BR>Versi 1</FONT></PRE></DIV>

<DIV ID="KEYS" STYLE='position:absolute; left:16; top:320'><PRE><FONT SIZE="-1">
```

- PETUNJUK -
Bergerak Dengan 2,4,6,8 atau K,J,L,I
Menembak Biasa Dengan Spasi atau Z
Menembak Peluru Kendali Dengan 0 atau X
</FONT></PRE></DIV>

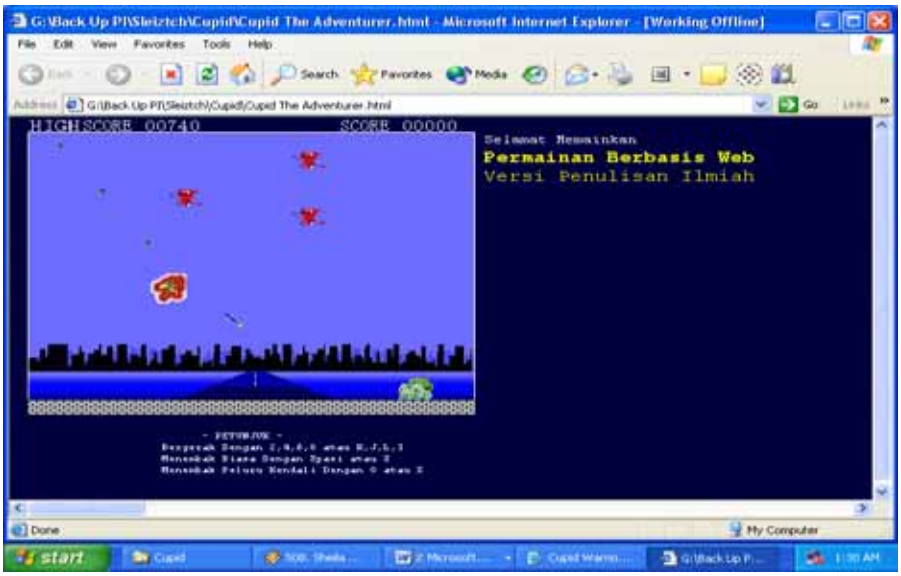
```
<DIVID="W1"STYLE='position:absolute;left:16; top:16; clip:rect(0,400,300,0)'>
<SCRIPT LANGUAGE="JavaScript">
<!--
  spINI("Mychr",100,100,32,32,"f16c.gif")
  spINI("MyF01",-100,0,32,8,"fire0c.gif")
  spINI("MyF02",-100,0,32,8,"fire0c.gif")
  spINI("MyF03",-100,0,32,8,"fire0c.gif")
  spINI("MyF04",-100,0,32,8,"fire0c.gif")
  spINI("MyF05",-100,0,32,8,"fire0c.gif")
  spINI("MyF06",-100,0,32,8,"fire0c.gif")
  spINI("MyF07",-100,0,32,8,"fire0c.gif")
  spINI("MyF08",-100,0,32,8,"fire0c.gif")
  spINI("MyF09",-100,0,32,8,"fire0c.gif")
  spINI("MyF10",-100,0,32,8,"fire0c.gif")
  spINI("MyF11",-100,0,32,8,"fire0c.gif")
  spINI("MyF12",-100,0,32,8,"fire0c.gif")
  spINI("MyF13",-100,0,32,8,"fire0c.gif")
  spINI("boss2",-180,120,50,50,"boss2c.gif")
  spINI("boss1",-230,170,80,50,"boss1c.gif")
  spINI("bossR",-230,170,80,50,"boss1rc.gif")
  spINI("boss0",-200,220,128,64,"boss0c.gif")
  spINI("EnF1",-100,0,8,8,"fire1c.gif")
  spINI("EnF2",-100,0,8,8,"fire1c.gif")
  spINI("EnF3",-100,0,8,8,"fire1c.gif")
  spINI("EnF4",-100,0,8,8,"fire1c.gif")
  spINI("EnF5",-100,0,8,8,"fire1c.gif")
  spINI("EnF6",-100,0,8,8,"fire1c.gif")
  spINI("EnF7",-100,0,8,8,"fire1c.gif")
  spINI("EnF8",-100,0,8,8,"fire1c.gif")
  spINI("EnF9",-100,0,8,8,"fire1c.gif")
  spINI("M231",-100,0,32,19,"m23c.gif")
  spINI("M232",-100,0,32,19,"m23c.gif")
  spINI("M233",-100,0,32,19,"m23c.gif")
  spINI("cob1",-100,0,32,19,"cobrac.gif")
  spINI("cob2",-100,0,32,19,"cobrac.gif")
  spINI("cob3",-100,0,32,19,"cobrac.gif")
  spINI("shi1",-100,0,32,24,"shilkac.gif")
  spINI("shi2",-100,0,32,24,"shilkac.gif")
  spINI("shi3",-100,0,32,24,"shilkac.gif")
  spINI("sa81",-100,0,32,24,"sa8c.gif")
  spINI("msl0",-100,0,24,24,"msl0c.gif")
  spINI("msl1",-100,0,24,24,"msl1c.gif")
  spINI("msl2",-100,0,24,24,"msl2c.gif")
  spINI("msl3",-100,0,24,24,"msl3c.gif")
  spINI("msl4",-100,0,24,24,"msl4c.gif")
  spINI("msl5",-100,0,24,24,"msl5c.gif")
  spINI("msl6",-100,0,24,24,"msl6c.gif")
  spINI("msl7",-100,0,24,24,"msl7c.gif")
  spINI("bom0",-100,0,32,8,"bom0c.gif")
  spINI("bom1",-100,0,32,8,"bom1c.gif")
  spINI("miss",-100,0,16,16,"clashc.gif")
  spINI("clash",-100,0,32,32,"clashc.gif")
```

```
   spINI("clash1",-100,0,32,32,"clashc.gif")
   spINI("clash2",-100,0,64,64,"clash2c.gif")
-->
</SCRIPT>
<!--<DIV ID="MTN" STYLE='position:absolute; left:0; top:200'><table
Border="0"Cellpadding="0"Cellspacing="0"><TD><IMG SRC="BKG.png"></TD><TD><IMG
SRC="BKG.png"></TD><TD><IMG SRC="BKG.png"></TD><TD><IMG
SRC="BKG.png"></TD><TD><IMG SRC="BKG.png"></TD><TD><IMG
SRC="BKG.png"></TD><TD><IMG SRC="BKG.png"></TD><TD><IMG
SRC="BKG.png"></TD><TD><IMG SRC="BKG.png"></TD><TD><IMG
SRC="BKG.png"></TD><TD><IMG SRC="BKG.png"></TD><TD><IMG
SRC="BKG.png"></TD><TD><IMG SRC="BKG.png"></TD></TABLE></DIV>-->

<DIV ID="GND" STYLE='position:absolute; left:0; top:284'><table Border="0" cellspacing="0"
Cellpadding="0"><TD>
  <IMG SRC="block.gif" width="16" height="16"></TD><TD>
  <IMG SRC="block.gif" width="16" height="16"></TD><TD>
  <IMG SRC="block.gif" width="16" height="16"></TD><TD>
  <IMG SRC="block.gif" width="16" height="16"></TD><TD>
  <IMG SRC="block.gif" width="16" height="16"></TD><TD>
  <IMG SRC="block.gif" width="16" height="16"></TD><TD>
  <IMG SRC="block.gif" width="16" height="16"></TD><TD>
  <IMG SRC="block.gif" width="16" height="16"></TD><TD>
  <IMG SRC="block.gif" width="16" height="16"></TD><TD>
  <IMG SRC="block.gif" width="16" height="16"></TD><TD>
  <IMG SRC="block.gif" width="16" height="16"></TD><TD>
  <IMG SRC="block.gif" width="16" height="16"></TD><TD>
  <IMG SRC="block.gif" width="16" height="16"></TD><TD>
  <IMG SRC="block.gif" width="16" height="16"></TD><TD>
  <IMG SRC="block.gif" width="16" height="16"></TD><TD>
  <IMG SRC="block.gif" width="16" height="16"></TD><TD>
  <IMG SRC="block.gif" width="16" height="16"></TD><TD>
  <IMG SRC="block.gif" width="16" height="16"></TD><TD>
  <IMG SRC="block.gif" width="16" height="16"></TD><TD>
  <IMG SRC="block.gif" width="16" height="16"></TD><TD>
  <IMG SRC="block.gif" width="16" height="16"></TD><TD>
  <IMG SRC="block.gif" width="16" height="16"></TD><TD>
  <IMG SRC="block.gif" width="16" height="16"></TD><TD>
  <IMG SRC="block.gif" width="16" height="16"></TD><TD>
  <IMG SRC="block.gif" width="16" height="16"></TD><TD>
  <IMG SRC="block.gif" width="16" height="16"></TD></TABLE></DIV>

<DIV ID="doc" STYLE='position:absolute; left:150; top:100; visibility: hidden'>
<PRE><FONT COLOR="#000000">Press <U>S</U> to Start<BR>

<U>E</U> for Easy<BR>  <U>N</U> for Normal</FONT></PRE></DIV>
<DIV ID="doc2" STYLE='position:absolute; left:0; top:150; visibility: hidden'>

</PRE></DIV><DIV ID="EASY" STYLE='position:absolute; left:2; top:2; visibility: hidden'>EASY
MODE</DIV></BODY></HTML>
```

LAMPIRAN 3
OUTPUT PROGRAM



Tampilan Awal



Tampilan Proses Permainan