PENGEMBANGAN PERANGKAT LUNAK PERMAINAN MULTIPLAYER STRATEGI BATTLE SHIP MENGGUNAKAN WIFI

SKRIPSI

Disusun Oleh :

NAMA : Rahadian Mahendra

NIM : 06.12.654

JURUSAN TEKNIK KOMPUTER DAN INFORMATIKA S-1

KONSENTRASI TEKNIK ELEKTRONIKA

FAKULTAS TEKNOLOGI INDUSTRI

INSTITUT TEKNOLOGI NASIONAL MALANG

2011

LEMBAR PERSETUJUAN

PENGEMBANGAN PERANGKAT LUNAK PERMAINAN MULTIPLAYER STRATEGI BATTLE SHIP MENGGUNAKAN WIFI

SKRIPSI

Disusun dan Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh Gelar Sarjana Teknik Komputer dan Informatika Strata Satu (S-1)

Disusun Oleh :

RAHADIAN MAHENDRA
NIM : 06.12.654

Mengetahui

Ketua Jurusan Teknik Elektro S-1

Ir. Yusuf Ismail Nakhoda, MT
NIP.Y.1018800189

Diperiksa dan Disetujui

Dosen Pembimbing I	Dosen Pembimbing II
(M, Ibrahim Ashari, ST, MT)	(Sotyohadi, ST,)
NIP.P 103.0100.358	NIP.P 103.9700.309

JURUSAN TEKNIK KOMPUTER DAN INFORMATIKA S-1
KONSENTRASI TEKNIK ELEKTRONIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2011

PENGEMBANGAN PERANGKAT LUNAK PERMAINAN MULTIPLAYER STRATEGI BATTLE SHIP MENGGUNAKAN WIFI

RAHADIAN MAHENDRA (06.12.654)
Email: dieend99@yahoo.co.id

Konsentrasi Komputer dan Informatika, Jurusan Teknik Elektro
Fakultas Teknologi Industri
Institut Teknologi Nasional Malang
Jln. Raya Karanglo Km 2 Telp (0341) 417636 Fax (0341) 417634 Malang
Email: itn@itn.ac.id

Dosen Pembimbing : 1. Ibrahim Ashari, ST, MT
2. Sotyohadi, ST

Abstrak

Perangkat komputer saat ini tidak hanya digunakan untuk keperluan pekerjaan saja, namun juga telah digunakan sebagai alat hiburan digital. Salah satu hiburan yang dapat dinikmati melalui perangkat komputer dan saat ini sedang digemari oleh anak-anak dan dewasa adalah aplikasi permainan (komputer game) Aplikasi permainan komputer berkembang dengan pesatnya. Diantara aplikasi-aplikasi permainan yang telah ada saat ini, banyak yang rumit alur permainannya sehingga membuat bosan para penggunanya.

Dengan latar belakang tersebut dan terinspirasi dari sebuah buku tentang pemrograman permainan komputer, penulis membuat dan mengembangkan aplikasi permainan komputer sederhana dengan karakter kapal perang yang disebut dengan Battleship yang dapat dijadikan alternatif bagi para penggemar aplikasi permainan komputer dan dapat dipelajari dengan mudah bagi para pemula dibidang pemrograman komputer.Aplikasi permainan kapal perang (Battleship) dibangun menggunakan Microsoft Visual Basic versi 6.0 yang menggunakan Winsock (Windows Socket) sebagai penghubung jaringan dan sebagai pengenal port pada aplikasi computer.

Kata Kunci : Game,Winsock,WiFi

Abstract

Computer devices today are not only used for work purposes only, but also has been used as a means of digital entertainment. One of the entertainment that can be enjoyed via a computer device and is currently favored by children and adults is a game application (computer game). Application of computer games developed rapidly. Among the applications that have the game currently available, many of the complicated plot make the game so bored its users.

With a background and inspiration from a book about programming computer games, authors create and develop applications with a simple computer game character of warship called the Battleship which may be an alternative for fans of computer games and applications can be learned easily to beginners in the field of Applications warship games (Battleship) built using Microsoft Visual Basic 6.0 version that uses the Winsock (Windows Sockets) as the connective tissue and as a port identifier in computer applications.

Keyword : Game,Winsock,WiFi.

KATA PENGANTAR

Puji syukur kehadirat Tuhan Yang Maha Esa yang telah memberikan rahmat dan hidayah-Nya, sehingga dapat diselesaikan skripsi yang berjudul "**PENGEMBANGAN PERANGKAT LUNAK PERMAINAN MULTIPLAYER STRATEGI BATTLE SHIP MENGGUNAKAN WIFI**" ini dengan lancar. Skripsi ini merupakan persyaratan kelulusan Studi pada Jurusan Teknik Elektro S-1 Konsentrasi Teknik Komputer dan Informatika ITN Malang dan untuk mencapai gelar Sarjana Teknik.

Keberhasilan peyelesaian laporan skripsi ini tidak lepas dari dukungan dan bantuan berbagai pihak. Untuk itu penyusun menyampaikan terima kasih kepada :

1. Ir. Soeparno Djiwo, MT, selaku Rektor Institut Teknologi Nasional Malang.
2. Bapak Ir. Sidik Noertjahjono, MT, selaku Dekan Fakultas Teknologi Industri.
3. Bapak Ir. Yusuf Ismail Nakhoda, MT selaku Ketua Jurusan Teknik Elektro S-1.
4. Bapak Dr. Eng. Aryuanto, ST. MT selaku Sekretaris Jurusan Teknik Elektro S-1.
5. Bapak Ir. Yusuf Ismail Nakhoda, MT selaku Dosen Pembimbing.
6. *Thanks to Papa, Mama and brother that I love, who gave the prayer of blessing, spiritual encouragement, and spirit without them I am not who's.*
7. *Thanks God for gifing Apriliya Susanti ini my live, without her I can't as now. because with her great love to me make power for working thesis and I am verry grateful to have her.*
8. Teman – teman GARDA 12 community yang saya sayangi dimana kalian berada.
9. Semua teman angkatan 2006 tanpa terkecuali.
10. Semua yang telah membantu dalam penyelesaian penyusunan skripsi ini.

Penulis telah berusaha semaksimal mungkin dan menyadari sepenuhnya akan keterbatasan pengetahuan dalam menyelesaikan laporan ini. Untuk itu penyusun mengharapkan saran dan kritik yang membangun dari pembaca demi kesempurnaan laporan ini.

KATA PENGANTAR

Puji syukur kehadirat Tuhan Yang Maha Esa yang telah memberikan rahmat dan hidayah-Nya, sehingga dapat diselesaikan skripsi yang berjudul "PENGEMBANGAN PRAKTIKUM LINGKAR PRIMARIKAN MULTIPLATFORM SINATRUL BATTLE SERIE MENGGUNAKAN WIFI" ini dengan lancar. Skripsi ini merupakan syarat untuk kelulusan Siswi pada Jurusan Teknik Elektro S-1 Konsentrasi Teknik Komputer dan Informatika FTK Akademi dan untuk memperoleh gelar Sarjana Teknik.

Kekurtahilan pengetahuan Ilmiah sangat ini tidak lepas dari dukungan dan bantuan berbagai pihak. Untuk itu penyusun mengucapkan terima kasih kepada :

1. Ir. Suyatno Djiwo MT. selaku Rektor Institut Teknologi Nasional Malang.
2. Ibnu Ir. Siti. Nanjatiyono, MT. selaku Dekan Fakultas Teknologi Industri
3. Dipak Ir. Yusuf Ismail Nakhoda, MT selaku Ketua Jurusan Teknik Elektro S-1.
4. Zenal Ir. Lutfi Agustono, ST. MT selaku Sekretaris Jurusan Teknik Elektro S-1.
5. Dipak Ir. Yusuf Ismail Nakhoda, MT selaku Dosen Pembimbing.
6. Tuhan ar Pipa, Ibunda dan Bapaknya tha Ayah, yang ada pengorbanan ar Bimbingan, perhatian, motivasi, dan doa yang tulus dan ikhlas.
7. Tuhan Ibnu dan pihak dosen Sekretaris dan dosen Fakultas dan Jurusan Teknik Elektro dan para pengajar lain yang telah memberikan pengetahuan dan motivasi dan ilmu pengetahuan selama ini.
8. Tuhan – teman GAMMA 12 community yang saya sayangi dimana kalian berada.
9. Semua teman angkatan Zelto ku pa Rekonsul.
10. Semua yang telah membantu dalam penyusunan penyelesaian skripsi ini.

Penulis telah berusaha semaksimal mungkin dan mengadakan seperlunya akan kesempurnaan penyusunan dalam mengidentifikasi tugasan ini. Untuk itu penyusun mengharapkan saran dan kritik yang membangun untuk perbaikan dan kesempurnaan laporan ini.

Harapan penyusun semoga laporan skripsi ini memberikan manfaat bagi perkembangan ilmu pengetahuan dan pembaca.

Malang, Juli 2011

penyusun

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR GAMBAR

DAFTAR TABEL

BAB I
PENDAHULUAN

1.1 Latar Belakang

Sesuai dengan perkembangan zaman yang semakin modern dan canggih, banyak pengguna layanan nirkabel (*WiFi*) menggunakannya untuk mengakses internet atau mensharing data saja. Tetapi sekarang sudah berkembang sebagai sarana penghubung permainan (*Game*) pada computer. Pada saat ini permainan (*Game*) jaringan *Local Area Network* (LAN) atau yang mengharuskan Online sudah banyak bermunculan sehingga banyak menarik minat para pengguna computer untuk mencobanya dan menerapkannya pada jaringan nirkable (*WiFi*).

Salah satu permainan (*Game*) komputer yang cukup menarik adalah permainan (*Game*) strategi *Battle Ship*. Permainan ini menyediakan beberapa buah tempat yang disusun oleh kotak – kotak sebagai medan perang. Kapal – kapal perang dengan ukuran yang berbeda – beda disusun dalam medan perang masing – masing. Letak dari kapal – kapal perang ini tidak terlihat dalam komputer pemain lawan. Setiap pemain berusaha untuk menghancurkan kapal – kapal perang pemain lawan dengan cara meng-klik kotak – kotak yang dianggap sebagai letak dari kapal – kapal perang lawan. Setiap pemain hanya boleh menebak satu kali saja setiap gilirannya. Jika tebakannya tepat maka pemain tersebut mendapat tambahan satu kali tebakan. Pemain yang masih menyisakan kapal perang yang memenangkan permainan (game).

Berdasarkan uraian di atas, penulis bermaksud untuk merancang suatu perangkat lunak permainan strategi *Battle Ship* yang dapat dimainkan pada jaringan. Oleh karena itu, penulis mengambil topik tugas akhir dengan judul "Pengembangan Perangkat Lunak Permainan Strategi *Battle Ship* Menggunakan *WiFi*".

1.2 Perumusan Masalah

Berdasarkan latar belakang pemilihan judul, maka yang menjadi permasalahan adalah bagaimana memainkan permainan strategi *Battle Ship* di tempat yang berbeda dengan menggunakan jaringan *WiFi*.

1.3 Tujuan dan Manfaat Penulisan

Tujuan penyusunan tugas akhir (skripsi) ini adalah untuk merancang suatu perangkat lunak permainan strategi *Battle Ship* pada jaringan *WiFi*. Manfaat dari penyusunan tugas akhir (skripsi) ini yaitu perangkat lunak dapat dijadikan sebagai sarana hiburan yang cukup menarik.

1.4 Pembatasan Masalah

Karena keterbatasan waktu dan pengetahuan penulis, maka ruang lingkup permasalahan dalam merancang perangkat lunak ini antara lain :

1. *Input* dibatasi hanya pada *keyboard* dan *mouse* sedangkan *Output* perangkat lunak pada layar monitor.
2. Jumlah pemain adalah sebanyak 2 orang.
3. Rincian dari masing – masing kapal perang, yaitu :
   - *Battleship* dengan ukuran 4 kotak sebanyak 1 buah.
   - *Cruiser* dengan ukuran 3 kotak sebanyak 2 buah.
   - *Submarine* dengan ukuran 2 kotak sebanyak 3 buah.
   - *Destroyer* dengan ukuran 1 kotak sebanyak 4 buah.
4. Ukuran dari medan perang adalah 10 kotak x 10 kotak.
5. Perangkat lunak menggunakan komponen *Winsock* pada *Microsoft Visual Basic* untuk melakukan koneksi komputer ke jaringan *WiFi*.

1.5 Metodologi Penelitian

Adapun metode penelitian yang digunakan adalah sebagai berikut:

1. Studi literature

   Pengumpulan data yang dilakukan dengan mencari bahan – bahan kepustakaan dan referensi dari berbagai sumber sebagai landasan teori yang ada hubungannya dengan permasalahan yang dijadikan objek penelitian.
2. Analisa kebutuhan system

Data dan informasi yang telah diperoleh akan dianalisa agar didapatkan kerangka global yang bertujuan untuk mendefinisikan kebutuhan system baik hardware maupun software, dimana nantinya akan digunakan sebagai acuan perancangan system.

3. Perancangan system

   Berdasarkan data dan informasi yang telah diperoleh serta analisa kebutuhan untuk membangun system ini, akan dibuat kerangka global yang menggambarkan mekanisme dari system yang akan dibuat dan tahapan ini menerjemahkan hasil perancangan spesifikasi program kedalam baris – baris kode program yang dapat dimegerti oleh computer.

4. Implementasi system

   Tahapan ini merupakan hasil uji coba program dan menampilkan hasil perancangan program yang diperoleh dari tahapan sebelumnya kedalam bentuk table dan printout dari perancangan spesifikasi program.

5. Eksperimen dan evaluasi

   Pada tahap ini, system yang telah selesai dibuat akan diuji coba, yaitu pengujian berdasarkan fungsionalitas program, dan akan dilakukan koreksi dan penyempurnaan program jika diperlukan.

1.6 Sistematika penulisan

Sistematikan penulisan yang diuraikan dalam penyusunan skripsi ini adalah sebagai berikut.

BAB I Pendahuluan

Bab ini berisi tentang Latar Belakang, Rumusan Masalah, Tujuan Penelitian, pembatasan Permasalahan, metode penelitian dan Sistematika Penulisan.

BAB II Landasan Teori

Berisi tentang landasan teori mengenai permasalahan yang berhubungan dengan penelitian yang dilakukan.

BAB III Perancangan dan Desain Aplikasi

Dalam bab ini berisi mengenai analisa kebutuhan sistem baik software maupun hardware yang diperlukan untuk membuat kerangka global yang menggambarkan mekanisme dari sistem yang akan dibuat.

BAB IV Uji Coba Aplikasi

Bab ini berisi tentang pengujian aplikasi, struktur dan tampilan aplikasi.

BAB V Penutup

Pada bab ini berisikan kesimpulan dan saran yang dapat digunakan sebagai pertimbangan untuk pengembangan penulisan selanjutnya.

BAB II

TINJAUAN PUSTAKA

2.1.Permainan (Game)

Menurut Kamus Besar Bahasa Indonesia, permainan merujuk pada pengertian "kelincahan intelektual" (*intellectual playability*). Permainan juga bisa diartikan sebagai arena keputusan dan aksi pemainnya. Ada target-target yang ingin dicapai pemainnya. Namun berbeda lagi pengertiannya, apabila dimasukkan kedalam suatu teori permainan. Teori permainan sendiri memiliki arti yaitu suatu cara belajar yang digunakan dalam menganalisa interaksi antara sejumlah pemain maupun perorangan yang menunjukkan strategi-strategi yang rasional[1]. Teori permainan pertama kali ditemukan oleh sekelompok ahli Matematika pada tahun 1944. Teori itu dikemukakan oleh John von Neumann dan Oscar Morgenstern (1953) yang berisi :

"Permainan terdiri atas sekumpulan peraturan yang membangun situasi bersaing dari dua sampai beberapa orang atau kelompok dengan memilih strategi yang dibangun untuk memaksimalkan kemenangan sendiri ataupun untuk meminimalkan kemenangan lawan. Peraturan-peraturan menentukan kemungkinan tindakan untuk setiap pemain, sejumlah keterangan diterima setiap pemain sebagai kemajuan bermain, dan sejumlah kemenangan atau kekalahan dalam berbagai situasi." [2]

Sedangkan menurut Agustinus Nilwan (diunduh melalui www.scribd.com pada tanggal 15 Juli 2011 dengan judul Pemrograman Animasi dan Game Profesional)[3], permainan komputer dapat dibuat dengan teknik dan metode animasi. Jika ingin mendalami pengunaan animasi haruslah memahami pembuatan game. Atau jika ingin membuat game, maka haruslah memahami teknik dan metode animasi, sebab keduanya saling berkaitan.

Sehingga dapat ditarik sebuah kesimpulan, permainan (*game*) adalah suatu aktivitas terstruktur yang dilakukan oleh seseorang sebagai pemain dengan adanya target – target yang ingin dicapai dan biasanya disepakati pula peraturan ataupun ketentuan yang diberlakukan selama aktivitas tersebut berlangsung sehingga menciptakan kreativitas dan meningkatkan intelektual yang dimiliki oleh pemain itu sendiri.

Adapun karakteristik didalam sebuah permainan (*game*) komputer, ada beberapa hal yang harus diperhatikan, seperti :

BAB II

TINJAUAN PUSTAKA

2.1 Permainan (Game)

Menurut Kamus Besar Bahasa Indonesia, permainan merujuk pada pengertian "kelincahan intelektual" (intellectual alertness). Permainan juga dikaitkan sebagai arena kepuasan dan aksi permainan. Ada tujuan-tujuan yang ingin dicapai pemainnya. Namun berbeda lagi pengertiannya apabila dikaitkan kedalam suatu teori permainan. Teori permainan adalah matematika yang suatu cara belajar yang digunakan dalam mengunakan matematika sejumlah permain maupun pengambilan yang menunjukan strategi-strategi yang rasional[11]. Teori permainan pertama kali diemukan oleh seleompok ahli Matematika pada tahun 1944. Teori ini dikemukakan oleh John von Neumann dan Oscar Morgenstern (1953) yang berisi :

"Permainan terdiri atas sekumpulan peraturan yang menghasilkan situasi bersaing dari dua sampai beberapa orang atau kelompok dengan memilih strategi yang dibangun untuk meraih simulasi kemenangan sesuai dengan untuk memaksimalkan kemenangan lawan. Peraturan-peraturan menentukan kemungkinan tindakan untuk setiap pemain, sejumlah keterangan diterima setiap pemain sebagai kemajuan bermain, dan sejumlah kemenangan atau kekalahan dalam berbagai situasi."[12]

Sedangkan menurut Agustinus Nilwan (diunduh melalui www.scribd.com pada tanggal 15 Juli 2011 dengan judul Permainan Animasi dan Game Profesional)[13], permainan komputer dapat dibuat dengan teknik dan metode animasi. Ada jenis mendalami pengunaan animasi bersifat memalami pembuatan game. Atau ada jenis mendalami game, maka bersifat mendalami teknik dan metode animasi, sebab kedunya saling berkaitan.

Sehingga dapat diartik sebuah kesimpulan permainan (game) adalah suatu aktivitas tersendiri yang dilakukan oleh seseorang sebagai pemain dengan adanya target – target yang ingin dicapai dan bisanya disepakati pula peraturan aturan tertentu yang dibentukkan selama aktivitas tersebut berlangsung sehingga mendapatkan kesenangan dan mengejakan tindakan yang dimiliki oleh pemain itu sendiri.

Adapun karakteristik dikalam sebuah permainan (game) komputer ada beberapa hal yang harus diperhatikan, seperti :

a. Peralatan

Alat yang dibutuhkan didalam sebuah permainan. Misal : bola, kartu, papan, atau sebuah komputer.

b. Peraturan

Peraturan digunakan untuk menentukan giliran pemain, hak dan keharusan masing – masing pemain, dan tujuan permainan.

c. Keahlian (Skill), Strategi dan Keberuntungan

Permainan dengan keahlian (*skill*) adalah pemain menggunakan beberapa hal yang diahlikan seperti dengan kekuatan fisik, misal : gulat, menembak dan kekuatann mental seperti catur. Namun sebuah permainan tidak hanya menggunakan keahlian tetapi juga strategi yang digunakan. Strategi adalah suatu cara permainan yang digunakan sendiri oleh pemain untuk mematikan lawan. Strategi biasanya lebih mengarah kepada kemampuan otak kanan seseorang. Dan yang terakhur, apabila pemain telah menjalankan keahlian dan juga strategi, factor terakhir yang mempengaruhi dari sebuah permainan adalah factor keberuntungan (*luck factor*).

d. *Single Player Game* (Pemain satu orang) dam *Double Player* (lebih dari satu pemain)

Jika pemain tunggal, pemain harus bermain dengan keahlian, berpacu dengan waktu dan keberuntungan sedangkan pemain *double*, pemain diharuskan untuk menggunakan suatu strategi dan kekompakan sesama pemain, untuk mencapai tujuan tertentu atau sebaliknya pemain harus berlomba dengan pemain lainnya untuk mencapai sesuatu tujuan.

2.2.Animasi

Animasi merupakan salah satu bagian grafika komputer yang menyajikan tampilan-tampilan atraktif yang juga merupakan sekumpulan gambar yang ditampilkan secara berurutan dengan cepat untuk mensimulasi gerakan yang hidup [4]. Pemanfaatan animasi dapat ditujukan untuk simulasi, menarik perhatian pemakai komputer pada bagian tertentu dari layar, memvisualisasikan cara kerja suatu alat atau menampilkan keluaran program dengan gambar-gambar yang

a. Peralatan

Alat yang dibutuhkan didalam sebuah permainan. Misal : bola, kartu, papan, atau sebuah komputer.

b. Peraturan

Peraturan digunakan untuk membatasi gerakan pemain, baik dari kebarusan masing – masing pemain, dan tujuan permainan

c. Kecakilan (Skill), Strategi dan Keberuntungan

Permainan dengan kecakilan (skill) adalah permain menggunakan beberapa hal yang dibutuhkan seperti dengan kekuatan fisik, misal : gulat, menembak dan kekuatan mental seperti catur. Namun sebuah permainan tidak hanya menggunakan kecakilan atau juga strategi yang digunakan. Strategi adalah suatu cara permainan yang digunakan sebelum oleh pemain untuk mematikan lawan. Strategi biasanya lebih mengarah kepada kemampuan otak kanan sesorang. Dan yang terakhir, apabila permain telah menjalankan kecakilan dan juga strategi, faktor terakhir yang mempengaruhi dari sebuah permainan adalah faktor keberuntungan (luck factor).

d. Single Player Game (Permain satu orang) dan Double Player (lebih dari satu permain)

Jika permain tunggal, permain harus bermain dengan kecakilan, berpacu dengan waktu dan keberuntungan sedangkan permain ganda, permain dituntutkan untuk menggunakan suatu strategi dan kekompakan sesama permain, untuk mencapai tujuan tertentu dan sebelumnya permain harus berlomba dengan permain lainnya untuk mencapai sasaran tujuan.

2.2.Animasi

Animasi merupakan salah satu bagian grafika komputer yang menyajikan tampilan-tampilan analit yang juga merupakan sekumpulan gambar yang ditampilkan secara berurutan dengan cepat untuk menimbulkan gerakan yang hidup [4]. Pemanfaatan animasi dapat ditujukan untuk simulasi, menarik perhatian pemakai komputer pada bagian tertentu dari layar, memvisualisasikan cara kerja suatu alat atau menampilkan keluaran program dengan gambar-gambar yang

menarik dibanding dengan sederetan angka, serta tidak ketinggalan untuk program-program permainan.

Pada dasarnya, animasi adalah transformasi objek yang dimana semua titik pada sembarang objek akan diubah sesuai dengan aturan tertentu, sementara sistem koordinatnya tetap. Implementasi pada animasi dapat dikerjakan secara interaktif maupun non interaktif. Dibandingkan animasi non interaktif, animasi interaktif memberikan tampilan yang lebih menarik dan dinamis. Pada animasi interaktif, pergerakan objek mengikuti perintah yang diberikan oleh pemakai lewat perangkat interaktif. Sedangkan animasi non interaktif, pergerakan objek hanya dikendalikan dari prosedur yang ada di dalam sebuah program[5]. Untuk animasi interaktif kebanyakan digunakan untuk program-program permainan, sedangkan animasi non interaktif kebanyakan untuk melakukan simulasi objek.

Animasi yang bagus dihasilkan dari gambar yang cukup banyak agar gambar yang dihasilkan akan tampak gerakan yang berkesan halus. Dalam hal ini, maka gambar-gambar tersebut haruslah berpindah posisi sekecil mungkin agar pada perubahan atau pergantian gambar terlihat lebih menarik dan bagus. Selain itu diperlukan juga kecepatan tertentu untuk tampilan gambar yang akan dibuat dalam animasi. Hal ini tergantung pada jumlah gambar yang diberikan. Kecepatan yang dimaksud yaitu begitu satu gambar ditampilkan maka akan berganti gambar berikutnya dengan kecepatan tertentu. Makin cepat pergantian antara satu gambar dengan gambar berikutnya maka akan menghasilkan gerakan gambar yang semakin halus.

2.3. Metode Grafik [6]

Yaitu metode penyelesaian permainan dengan menggunakan grafik. Metode grafik ini dapat digunakan untuk menyelesaikan kasus permainan di antaranya adalah sebagai berikut:

a. Matriks berukuran 2 x n

Matriks pembayaran dari permainan berukuran 2 x n adalah

	Pemain P2	
		Y1 y2 ... yn

		j i	1 2 … n
Pemain P1	X1	1	a11 a12 … a1n
	X2 = 1 – x1	2	a21 a22 … a2n

Tabel 2.1 Matriks Pembayaran Permainan Berukuran 2 x n

Dengan $\sum_{i=1}^{2} x_i = 1$ dan $\sum_{j=1}^{n} y_1 = 1$, xi ≥ 0 dan y1 ≥ 1 untuk setiap i, j. Pembayaran harapan yang berkaitan dengan strategi murni pemain P2.

Strategi murni pemain 2	Pembayaran harapan pemain 1
1	a11.x1 + a21.(1 – x1) = (a11 – a21).x1 + a21
2	a12.x1 + a22.(1 – x1) = (a12 – a22).x1 + a22
n	a1n.x1 + a2n.(1 – x1) = (a1n – a2n).x1 + a2n

Tabel 2.2 Rata – Rata Pembayaran

Tabel di atas menunjukkan bahwa pembayaran harapan (rata-rata pembayaran) bagi pemain P1 bervariasi secara linear dengan x1. Berdasarkan kriteria minimax untuk pemain P1 harus memilih nilai x1 yang akan memaksimalkan pembayaran harapan (rata-rata pembayaran) minimumnya (prinsip maximin). Hal ini dapat dilakukan dengan cara menggambarkan garisgaris lurus di atas sebagai fungsi dari x1. Sumbu vertikal menunjukkan pembayaran harapan (rata-rata pembayaran) dan sumbu horizontal menunjukkan variasi dari x1 (0 ≤ x1 ≤ 1). Dalam grafik ini dicari titik maximinnya.

Contoh:

Diberikan matriks pembayaran sebagai berikut. Akan dicari strategi optimum untuk P1 dan P2.

		P_2		
P_1	j i	y1	y 2	y3
	x1 x2 = 1-x1	-1 5	1 3	3 -3

Tabel 2.3 Matriks Pembayaran

Penyelesaian

x1 = probabilitas pemain 1 memainkan strategi kesatu

x2 = probabilitas pemain 2 memainkan strategi kedua

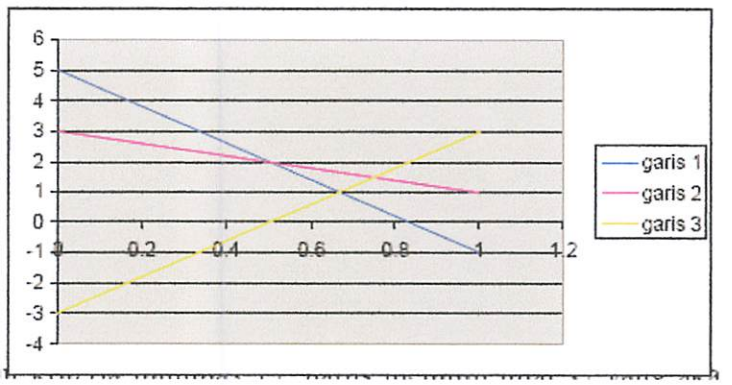
yj = probabilitas pemain 2 memainkan strategi ke-j

maka pembayaran harapan bagi pemain P1 yang berkaitan dengan strategi murni P2 adalah

Strategi murni P2	Pembayaran harapan P1
1	$-x_1 + 5(1-x_1) = -6.x_1 + 5$
2	$x_1 + 3(1- x_1) = -2. x_1 + 3$
3	$3. x_1 - 3(1-x1) = 6. x_1 - 3$

Tabel 2.4 Pembayaran Harapan P1

Ketiga garis lurus fungsi dari x1 tersebut dapat digambarkan pada grafik



Gambar 2.1 Grafik Pembayaran Harapan P1

Menurut kriteria minimax P1, harus memilih nilai x1 yang akan **memaksimalkan** pembayaran harapan minimumnya yaitu

v* = max(x1) { min (-6x1 + 5, -2x1 + 3 , 6x1 - 3)}

karena hanya garis (1) dan (3) yang melalui titik maximin maka

v* = max(x1) { min (-6 x1 + 5, 6 x1 - 3)}

dari sini nilai optimum x1 _ titik potong garis (1) dengan garis (3)

-6 x1 + 5 = 6 x1 – 3 _ 12 x1 = 8

x1 = x1* = 2/3

x2* = 1- x1* = 1/3

jadi strategi campuran optimum P1 X* = [2/3, 1/3]

Nilai permainan yang diperoleh

v*= -6x1* + 5 = -6. 2/3 + 5 = 1 atau v* = 6x1* – 3 = 1

selanjutnya akan dihitung strategi optimum pemain P2 . Nilai yang optimum bagi pemain

pembayaran P2 dapat diperoleh dari nilai pembayaran harapan permainan, yaitu:

$$v^* = \sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij} x_2^* y_j^*$$

sehingga

y1*(-6x1* + 5) + y2* (-2x1* + 3) + y3* (6x1*-3) = v*

y1* + 5/3 y2* + y3* = 1

y1* + y3* = 1

(y2* = 0 karena tidak melalui titik maximin dimana v* > 1)

Jadi strategi **kedua** pemain P2 **tidak** dimainkan, sehingga matriks pembayarannya menjadi

	P1		
P2	j i	y1	y3=1-y1
	x_1	-1	3
	x_2	5	-3

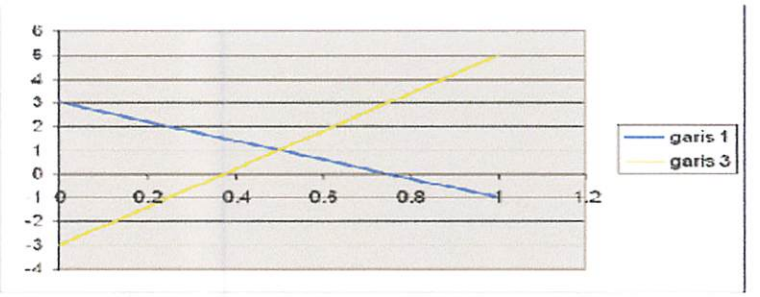
Tabel 2.5 Matriks Pembayaran

maka pembayaran harapan bagi pemain P2 yang berkaitan dengan strategi murni P1 adalah

Strategi murni P1	Pembayaran harapan P2
1	-y1 – 3(1-y1) = 3 - 4y1
2	5y1 - 3(1-y1) = 8x1 - 3

Tabel 2.6 Pembayaran Harapan P2

Kedua garis lurus fungsi dari y1 tersebut dapat digambarkan pada grafik



Gambar 2.2 Grafik Pembayaran Harapan P2

Karena P2 menginginkan untuk meminimumkan kekalahan yang maksimum maka pemain P2 harus memilih nilai y1 yang akan **meminimumkan** pembayaran harapan yang maksimum, yaitu:

$v^* = \min(y1)\ \{\max(3 - 4y1,\ 8y1 - 3)\}$

karena kedua garis (1) dan (2) melalui titik minimax maka nilai optimum y1* adalah titik potong

kedua garis tersebut, diperoleh

$y1 = y1^* = ½$

karena $y1 + y3 = 1$ maka $y3^* = 1 - y1^* = ½$

jadi strategi optimum P2 $y^* = [\ ½\ ,\ 0,\ ½]$, dan nilai permainan $v^* = 1$

b. Matriks berukuran m x 2

Matriks pembayaran dari permainan berukuran 2 x n adalah

			Pemain P2	
			y1	y2 = 1 – y1
		j i	1	2
Pemain P1	X1	1	a11	a12
	X2	2	a21	a22
	. . xm	. . M	. . am1	. . am2

Tabel 2.7 Matriks Pembayaran Berukuran 2 x n

Pembayaran harapan yang berkaitan dengan strategi murni pemain P2

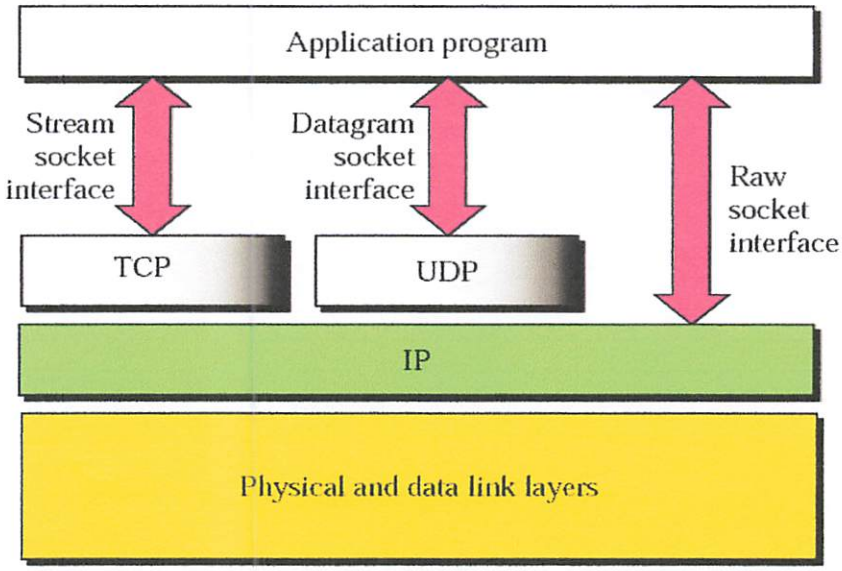
Strategi murni pemain 1	Pembayaran harapan pemain 2 (P2)
1	(a11 – a21)y1 – a12
2	(a12 – a22)y1 – a22
.	.
.	.
.	.
M	(am1 – am2)y1 + am2

Tabel 2.8 Pembayaran harapan P2

Tabel di atas menunjukkan bahwa pembayaran harapan (rata-rata pembayaran) bagi pemain P2 bervariasi secara linear dengan y1. Berdasarkan kriteria minimax untuk pemain P2 harus memilih nilai y1 yang akan meminimumkan pembayaran harapan (rata-rata pembayaran) maksimumnya (prinsip minimax). Hal ini dapat dilakukan dengan cara menggambarkan garisgaris lurus di atas sebagai fungsi dari y1. Sumbu vertikal menunjukkan pembayaran harapan (rata-rata pembayaran) dan sumbu horizontal menunjukkan variasi dari y_1 ($0 \leq y_1 \leq 1$). Dalam grafik ini dicari titik minimaxnya.

2.4. WINSOCK (*WINdows SOCKet*)

WINSOCK (*WINdows SOCKet*) merupakan program API (*Application Programming Interface*)[7]. Winsock API berupa sekumpulan fungsi-fungsi yang dapat digunakan untuk antarmuka program aplikasi dengan suatu Socket. Dua diantara program Socket adalah: *Socket Barkeley* (dari *Barkeley Software Distribution of UNIX*) dan WINSOCK (dari *Microsoft Windows*). Winsock memiliki beberapa fungsi tambahan dari Socket Barkeley yang digunakan untuk mendukung Sistem Operasi Windows yang menggunakan konsep arsitektur *Massage Driven* –nya[7].



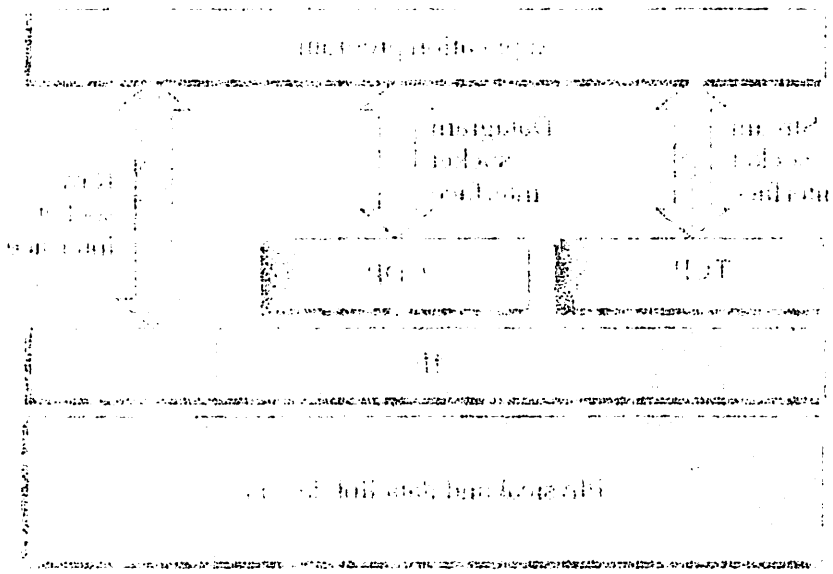
Gambar 2.3 Tipe Socket

Pada dasarnya winsock merupakan sebuah interface yang menghubungkan antara TCP/IP dengan aplikasi berbasis Windows. TCP/IP biasa dijuluki sebagai "bahasa untuk internet", dimana julukan ini diambil karena internet menggunakan sistem yang bergantung pada TCP/IP, untuk dapat saling berkomunikasi.

Winsock merupakan sistem penghubung yang memiliki ekstensi berupa *Dynamic Link Library* (DLL),dan berjalan dibawah sistem operasi *Windows 3x*, Windows untuk *workgroups, Windows NT*, dan *Windows* generasi selanjutnya. Winsock bekerja sebagai sebuah layer yang berada diantara aplikasi winsock dengan stack TCP/IP sebagai perantara komunikasi data, lalu stack TCP/IP meneruskannya ke jaringan atau ke *internet*.

Pada aplikasi *client/server*, soket digunakan dalam implementasi program sisi client/ sisi server. Saat client mengirim request , soket pengiriman ada pada sesi client, sementara soket penerimaan ada pada sisi server. Pada saat server mengirimkan response, soket pengiriman ada pada sisi server sementara soket penerimaan ada pada sisi client.

Sebuah soket dilengkapi alamat, yang terdiri atas alamat IP tujuan dan nomor port. Nomor port merupakan bilangan bulat yang digunakan untuk membedakan layanan-layanan yang berjalan pada komputer server yang sama. Pengguna layanan menggunakan nomor port ini untuk menghubungi server. Nomor-nomor port yang dipergunakan baik itu dalam TCP maupun UDP merupakan jenis port statis dan port dinamis. Apapun implementasi dari TCP/IP (contohnya pabrikan produsennya) yang dipergunakan, aplikasi-aplikasi yang mengurutkan penomoran port yang akan



Gambar 2.2 Tipe Socket

Pada dasarnya *winsock* merupakan sebuah *interface* yang menghubungkan antara TCP/IP dengan aplikasi berbasis *Windows*. TCP/IP bisa dipakai sebagai "bahasa untuk Internet", dimana jaringan lokal maupun Internet menggunakan sistem yang bergantung pada TCP/IP untuk dapat saling berkomunikasi.

Winsock merupakan sistem penghubung yang memiliki ekstensi berupa *Dynamic Link Library* (DLL) dan berjalan dibawah sistem operasi *Windows*. Jika *Windows* untuk workgroups, *Windows* NT, dan *Windows* versi sebelumnya, *Winsock* bekerja sebagai sebuah layer yang berada diantara aplikasi *winsock* dengan stack TCP/IP sebagai penerima kemudahan dan lalu stack TCP/IP menerjemahkannya ke jaringan atau ke *Internet*.

Pada aplikasi client/server, soket digunakan dalam implementasi program sisi client/ sisi server. Saat client mengirim request, soket yang dimiliki oleh pada sisi client, sementara soket pendengaran ada pada sisi server. Pada saat server mengirimkan response, soket pendengaran ada pada sisi server sementara soket percakapan ada pada sisi client.

Sebuah soket diidentifikasi dengan yang terdiri atas alamat IP tujuan dan nomor port. Nomor port merupakan bilangan bulat yang digunakan untuk membedakan layanan-layanan yang berjalan pada komputer server yang sama. Penggunaan layanan menggunakan nomor port ini untuk menghubungi server. Nomor-nomor port yang dipergunakan baik itu dalam TCP maupun UDP merupakan jenis port yang dikenal umum. Apapun implementasi dari TCP/IP memiliki pembatasan perbedaan yang dipergunakan, aplikasi-aplikasi yang mengunakan protocol port yang dan

dipergunakannya pasti akan selalu sama, yang biasa diketahui sebagai "ketetapan penomoran global" atau yang telah kita kenal sebagai nomor port. Yang telah ditetapkan oleh "*Central Authority*" untuk teknologi komputasi. Perbedaan dari port statis dan port dinamis ialah pada port statis bersifat port yang dipesan atau telah ditentukan terlebih dahulu untuk penyesuaian pemakaian dan aktifitas yang akan dilakukan melalui port-port tertentu tersebut, sedangkan pada port dinamis sifatnya dapat dipergunakan oleh berbagai komputer pada jaringan untuk melaksanakan aktifitas komunikasi lain [8].

2.5.Wireless LAN

Sejarah kemunculan WLAN dimulai pada tahun 1997, sebuah lembaga *independen* IEEE membuat spesifikasi/standar WLAN yang pertama yang diberi kode 802.11. peralatan yang sesuai standar 802.11 dapat bekerja pada frekuensi 2,4 GHz dengan kecepatan transfer data (*troughput*) maksimal 2 Mbps. WLAN atau Wireless Local Area Network adalah suatu jaringan komputer bersifat lokal yang memanfaatkan gelombang radio sebagai media transmisi data. Informasi data elektronik ditransfer dari satu komputer ke komputer lain melalui gelombang radio.WLAN sebagai jaringan *Wi-Fi* atau *WaveLAN* atau LAN *nirkabel* atau jaringan *wireless* atau *wireless*[9].

Komunikasi tanpa kabel / nirkabel (wireless) telah menjadi kebutuhan dasar atau gaya hidup masyarakat informasi. LAN nirkabel yang lebih dikenal dengan jaringan Wi-Fi menjadi teknologi alternatif dan relatif lebih mudah untuk diimplementasikan di lingkungan kerja (SOHO/Small Office Home Office), seperti perkantoran, laboratorium komputer, dan sebagainya[9]. Instalasi perangkat jaringan Wi-Fi lebih fleksibel karena tidak membutuhkan penghubung kabel antar komputer. Komputer dengan Wi-Fi Device dapat saling terhubung yang hanya membutuhkan ruang atau space dengan syarat jarak jangkauan dibatasi kekuatan pancaran sinyal radio dari masing-masing komputer.

2.5.1. LAN (Local Area Network)

Local area network adalah jaringan lokal yang dikenal pada area tertutup misalnya dalam satu gedung atau dalam satu ruangan[]. Jaringan ini juga disebut juga dengan jaringan privat. LAN biasanya digunakan untuk jaringan kecil yang menggunakan resource bersama – sama seperti pengunaan printer secara bersama, penggunaan media penyimpanan secara bersama.

dipergunakannya pasti akan sedikit sama, yang bisa dikerahui sebagai "keterapan pemrosesan global" atau yang lebih lain kerap sebagai nomor port. Yang telah diterapkan oleh "Control Manufacturer" untuk teknologi komputasi. Perbedaan dari port statis dan port dinamis ialah pada port statis berisiko port yang tipecan atau telah ditentukan terlebih dahulu untuk pengosisian permasalan dan aktifitas yang akan dilakukan melalui port port tertentu tersebut, sedangkan pada port dinamis adalah dapat dipergunakan oleh berbagai komputer pada jaringan untuk melaksanakan aktifitas komunikasi lain [26].

2.5.Wireless LAN

Sejarah kemunculan WLAN dimulai pada tahun 1997, sebuah lembaga pedoman IEEE membuat spesifikasistandar WLAN yang pertama yang diberi kode 802.11, peralatan yang sesuai standar 802.11 dapat bekerja pada frekuensi 2,4 GHz dengan kecepatan transfer data (throughput) maksimal 2 Mbps. WLAN atau Wireless Local Area Network adalah suatu jaringan komputer berskala lokal yang memanfaatkan gelombang radio sebagai media transmisi data. Informasi dan elektronik ditransfer dari satu komputer ke komputer lain melalui gelombang radio.WLAN sebagai jaringan Wi-Fi atau Hiperlan atau LAN nirkabel atau jaringan wireless atau wireless[27].

Komunikasi tanpa kabel / nirkabel (wireless) telah menjadi kebutuhan dasar atau gaya hidup masyarakat informasi. LAN nirkabel yang lebih dikenal dengan jaringan Wi-Fi merupakan teknologi alternatif dan relatif lebih mudah untuk diimplementasikan di lingkungan kerja (SOHO(Small Office Home Office), seperti perkantoran, laboratorium komputer, dan sebagainya[28]. Instalasi pemasangan jaringan Wi-Fi lebih fleksibel karena tidak membutuhkan pengikatan kabel antar komputer. Komputer dengan Wi-Fi Device dapat saling terhubung yang bahwa membutuhkan ruang atau space dengan syarat jarak jangkauan dibatasi kekuatan pancaran sinyal radio dari masing-masing komputer.

2.5.1. LAN (Local Area Network)

Local area network adalah jaringan lokal yang dikenal pada area tertentu misalnya dalam satu gedung atau dalam satu ruangan[]. Jaringan ini juga dikenal juga dengan jaringan privat. LAN biasanya digunakan untuk jaringan kerja yang menggunakan resource bersama – sama seperti pengunaan printer secara bersama, pengunaan media penyimpanan secara bersama.

Adapun tabel yang menjelaskan daerah cakupan masing – masing seperti yang telah dijelaskan diatas, sebagai berikut :

Jarak / Cakupan (meter)	Contoh	Jenis
10 – 100	Ruangan	LAN
100 – 1000	Gedung	LAN
1000 – 10.000	Kampus	LAN

Tabel 2.9. Jaringan Komputer berdasarkan Area

2.5.2. Teknologi jaringan Wi-Fi

Wi-Fi atau Wireles Fidelity adalah salah satu standar Wireless Networking tanpa kabel, hanya dengan komponen yang dapat terkoneksi ke jaringan[10]. Teknologi Wi-Fi memiliki standar yang ditetapkan oleh sebuah institusi international yang bernama Institute of Electrical and Electronic Engineers (IEEE), yang secara umum sebagai berikut:

1. Standar IEEE 802.11a yaitu WiFi dengan frekuensi 5 Ghz yang memiliki kecepatan 54 Mbps dan jangkauan jaringan 300 m.
2. Standar IEEE 802.11b yaitu WiFi dengan frekuensi 2,4 Ghz yang memiliki kecepatan 11 Mbps dan jangkauan jaringan 100 m.
3. Standar IEEE 802.11g yaitu WiFi dengan frekuensi 2,4 Ghz yang memiliki kecepatan 54 Mbps dan jangkauan jaringan 300 m.

Teknologi Wi-Fi yang akan diimplementasikan adalah standar IEEE 802.11g karena standar tersebut lebih cepat untuk proses transfer data dengan jangkauan jaringan yang lebih jauh.

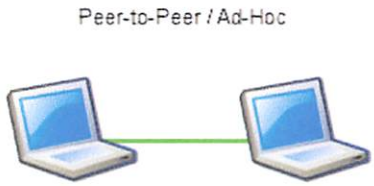
2.5.3. Tipe Jaringan Wi-Fi

Seperti halnya Ethernet-LAN (jaringan dengan kabel), jaringan Wi-Fi juga dikonfigurasikan ke dalam dua jenis jaringan:

1. Jaringan perr to perr / Ad Hoc Wireless LAN

Komputer dapat saling berhubungan berdasarkan nama SSID (Service Set Identifier). SSID adalah nama identitas kopmputer yang memiliki

komponen nirkabel, pada jaringan perr to perr/ ad – hoc salah satu berfungsi sebagai server dan yang lain sebagai client.

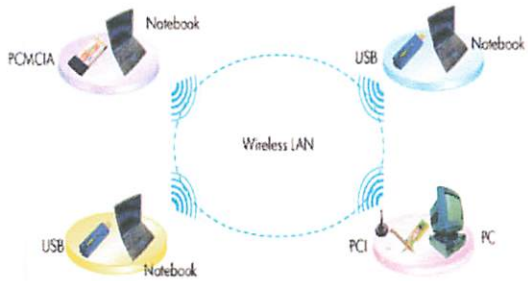


Server Client

2.4. Gambar Jaringan Perr to Perr / Ad Hoc

2. Jaringan Server Based / Wireless Infrastructure

Sistem Infrastruktur membutuhkan sebuah komponen khusus yang berfungsi sebagai Access Point.



Gambar 2.5. Jaringan Server Based / Wireless Infrastructure

2.6. IP Address

IP Address merupakan pengenal yang digunakan umtuk memberi alamat pada tiap-tiap komputer dalam jaringan. Format IP address adalah bilangan 32 bit yang tiap 8 bitnya dipisahkan oleh tanda titik. Adapun format IP Address dapat berupa bentuk 'biner' (xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx dengan x merupakan bilangan biner). Atau dengan bentuk empat bilangan desimal yang masing-masing dipisahkan oleh titik bentuk ini dikenal dengan 'dotted decimal' (xxx.xxx.xxx.xxx adapun xxx merupakan nilai dari satu oktet/delapan bit)[11].

Sebelumnya dikenal cara-cara pembagian IP Address, dimana IP address (yang berjumlah sekitar 4 milyar) dibagi kedalam lima kelas yakni:

Kelas A

Format : 0nnnnnnn.hhhhhhhh.hhhhhhhh.hhhhhhhh

Bit pertama : 0

Panjang NetID : 8 bit

Panjang HostID : 24 bit

Byte pertama : 0-127

Jumlah : 126 Kelas A (0 dan 127 dicadangkan)

Range IP : 1.xxx.xxx.xxx sampai 126.xxx.xxx.xxx

Jumlah IP : 16.777.214 IP Address pada setiap Kelas A

Dekripsi : Diberikan untuk jaringan dengan jumlah host yang besar

Kelas B

Format : 10nnnnnn.nnnnnnnn.hhhhhhhh.hhhhhhhh

Bit pertama : 10

Panjang NetID : 16 bit

Panjang HostID : 16 bit

Byte pertama : 128-191

Jumlah : 16.384 Kelas B

Range IP : 128.0.xxx.xxx sampai 191.155.xxx.xxx

Jumlah IP : 65.532 IP Address pada setiap Kelas B

Deskripsi : Dialokasikan untuk jaringan besar dan sedang

Kelas C

Format : 110nnnnn.nnnnnnnn.nnnnnnnn.hhhhhhhh

Bit pertama : 110

Panjang NetID : 24 bit

Panjang HostID : 8 bit

Byte pertama : 192-223

Jumlah : 2.097.152 Kelas C

Range IP : 1.xxx.xxx.xxx sampai 223.255.255.xxx

Jumlah IP : 254 IP Address pada setiap Kelas C

Deskripsi : Digunakan untuk jaringan berukuran kecil

Kelas D

Format : 1110mmmm.mmmmmmm. mmmmmmm. mmmmmmm

Bit pertama : 1110

Bit multicast : 28 bit

Byte inisial : 224-247

Deskripsi : Kelas D digunakan untuk keperluan IP multicasting (RFC 1112)

Kelas E

Format : 1111rrrr.rrrrrrrr. rrrrrrrr. rrrrrrrr

Bit pertama : 1111

Bit cadangan : 28 bit

Byte inisial : 248-255

Deskripsi : Kelas E dicadangkan untuk keperluan eksperimental.

Saat ini dikenal juga cara pengalokasian IP Address dalam notasi Classless Inter Domain Routing (**CIDR**) (network/mask). Istilah lain yang digunakan untuk menyebut bagian IP address yang menunjuk suatu jaringan secara lebih spesifik yakni: ***Network Prefix***. Biasanya dalam menuliskan network prefix suatu kelas IP Address digunakan tanda garis miring (*Slash*) "/", diikuti dengan angka yang menunjukan panjang network prefix ini dalam bit.

Misalnya, ketika menuliskan network kelas A dengan alokasi IP 12.xxx.xxx.xxx, network prefixnya dituliskan sebagai 12/8. Angka 8 menunjukan notasi CIDR yang merupakan jumlah bit yang digunakan oleh network prefix, yang berarti netmask-nya 255.0.0.0 dengan jumlah maksimum host pada jaringan sebanyak 1x6.777.214 node. Contoh lain untuk menunjukan suatu network kelas B 167.205.xxx.xxx digunakan: 167.205/18. Angka 18 merupakan notasi CIDR, yang berarti netmask yang digunakan pada jaringan ini adalah 255.255.192.0 dengan jumlah maksimum host pada jaringan sebanyak 16.382 node.

2.7. Format Citra

Dalam sistem operasi Windows, mendukung beberapa jenis format gambar. Aplikasi pengolahan grafis juga menyediakan fasilitas untuk membuka *file – file* dengan format yang berbeda. Adapun beberapa jenis format gambar yang didukung oleh sistem operasi Windows dan aplikasi grafis seperti : JPEG, WMF, BMP dan GIF[12].

2.7.1. Format Citra JPEG

JPEG (*Joint Photographic Experts Group*) adalah teknik kompresi grafik *high color bit – mapped* dan juga standar *file* yang dikembangkan oleh grup *joint photographic experts* yang menggunakan kombinasi DCT

Byte mulai : 224–247

Deskripsi : Kelas D digunakan untuk kepentingan IP multicasting (RFC 1112)

Kelas E

Format : 1111 nnnn.nnnnn.nnnnn.nnnnn

Bit pertama : 1111

Bit cadangan : 28 bit

Byte mulai : 248–255

Deskripsi : Kelas E dicadangkan untuk keperluan eksperimental.

Saat ini dikenal juga cara pengalokasian IP Address dalam model Classless Inter Domain Routing (CIDR) (*www.rfc.net*). Inilah lain yang digunakan untuk mengatur bagian IP address yang membuat suatu jaringan seperti lebih spesifik yakni *network prefix*. Bagian *network prefix* dalam mendefinisikan *network* prefix suatu kelas IP Address digunakan untuk pada mirip (*class*) "C" dikan dengan angka yang membuatkan panjang *network* prefix ini dalam bit.

Misalnya, kelas mendefinisikan *network* kelas A dengan alamat IP 1.xxx.xxx.xxx, *network* prefixnya dinyatakan sebagai 127/8. Angka 8 menunjukan notasi CIDR yang merupakan jumlah bit yang digunakan oleh *network* prefix yang berarti ketemuk-nya 255.0.0.0 dengan jumlah maksimum host pada jaringan sebanyak 126.777.214 node. Contoh lain untuk *network* kelas B 167.203.xxx.xxx digunakan 167.203/18. Angka 18 merupakan notasi CIDR yang berarti network yang digunakan pada jaringan ini adalah 255.255.192.0 dengan jumlah maksimum host pada jaringan sebanyak 16.382 node.

2.7. Format Citra

Dalam sistem operasi *Windows* membuka & beberapa jenis format gambar. Aplikasi pengolahan grafis juga mengadakan fasilitas untuk membuka /& menyimpan format yang berbeda. Adapun beberapa jenis format gambar yang didukung oleh sistem operasi *Windows* dan aplikasi grafis seperti : JPEG, *Web*, BMP dan GIF[3].

2.7.1. Format Citra JPEG

JPEG (*Joint Photographic Experts Group*) adalah teknik kompresi grafik *high color bit* – mempersempit jarak antara file yang dikembangkan oleh grup *joint photographic experts* yang menggunakan kombinasi DCT

(*Discrete Cosine Transform*) dan pengkodean huffman untuk mengkompresi suatu *file* citra.

JPEG adalah suatu algoritma kompresi yang bersifat *lossy* dimana kualitas citranya kurang bagus. *Lossy compression* adalah metode memperkecil ukuran *file* citra dengan cara membuang beberapa data, hal ini menyebabkan adanya sedikit penurunan kualitas citra. JPEG merupakan teknik dan standar universal untuk kompresi dan dekompresi citra tidak bergerak untuk digunakan pada kamera digital dan sistem pencitraan menggunakan komputer yang dikembangkan oleh *Joint photographic experts group.*

2.7.2. Format Citra WMF

WMF (*Windows Meta File*) adalah suatu format *file* grafik standar Windows yang biasanya digunakan pada *clipboard* dan representasi sebuah grafik. Format *file* ini menyimpan semua komponen dari suatu grafik beserta dengan atribut-atributnya seperti garis (*lines*), warna (*colors*), pola (*patterns*), teks (*text*) dan atribut teks (*text attributes*). Adapun jenis format *file* WMF yang telah ditingkatkan dari segi kualitas grafiknya yaitu EMF (*Enchange Meta File*).

2.7.3. Format Citra BMP

Format *file* BMP merupakan format *file* gambar standar untuk komputer – komputer yang menjalankan sistem operasi Windows. Format *file* ini dikembangkan oleh pihak Microsoft untuk menyimpan *file* gambar (*bitmap*) dan memungkinkan Windows untuk menampilkan kembali gambar tersebut. Struktur dari *file* BMP terdiri dari *Bitmap file header* dan *Bitmap info header*. Struktur *Bitmap file header* menyimpan informasi mengenai dimensi dan format warna dari *device independent* bitmap (DIV).

Jadi dapat disimpulkan bahwa *Bitmapfileheader* memberi informasi mengenai file dan *Bitmapinfoheader* memberikan informasi mengenai gambar. Tabel warna yang didefinisikan sebagai *array* dan struktur RGBQUAD dan sisanya adalah data gambar. Format ini mendukung resolusi dan monokrom hingga *true color* (16,7 juta warna).

2.7.4. Format Citra GIF

GIF (*Graphics Interchange Format*) merupakan format gambar standar yang menyediakan sejumlah kemampuan seperti ketajaman gambar

(Discrete Cosine Transform) dan pengkodean huffman untuk mengkompresi suatu file citra.

JPEG adalah suatu algoritma kompresi yang bersifat *lossy* dimana kualitas citanya *kurang bagus*. *Lossy compression* adalah metode mempercecil ukuran file citra dengan cara membuang beberapa data, hal ini mengakibatkan adanya sedikit perubahan kualitas citra. JPEG merupakan teknik dan standar universal untuk kompresi dan dekompresi citra tidak bergerak untuk digunakan pada kamera digital dan sistem pencitraan menggunakan komputer yang dikembangkan oleh *Joint photographic experts group*.

2.7.2. Format Citra WMF

WMF (*Windows Meta File*) adalah suatu format file grafik standar *Windows* yang bisanya digunakan pada *clipboard* dan representasi sebuah grafik. Format file ini mengimpan semua komponen dari suatu grafik beserta dengan atribut-atributnya seperti garis (*lines*), warna (*colors*), pola (*patterns*), teks (*text*) dan aturan teks (*text attributes*). Adapun jenis format file WMF yang telah ditingkatkan dari segi kualitas grafiknya yaitu EMF (*Enhanced Meta File*).

2.7.3. Format Citra BMP

Format file BMP merupakan format file gambar standar untuk komputer – komputer yang menjalankan sistem operasi *Windows*. Format file ini dikembangkan oleh pihak *Microsoft* untuk mengimpan file gambar (*bitmap*) dan mencangkinkan *Windows* untuk menampilkan kembali gambar tersebut. Struktur dari file BMP terdiri dari *bitmap file header* dan *bitmap info header*. Struktur *bitmap file header* mengimpan informasi mengenai dimensi dan format warna dari *device independent bitmap* (DIB).

Jadi dapat disimpulkan bahwa *bitmapfileheader* membori informasi mengenai file dan *bitmapinfoheader* membentukan informasi mengenai gambar. Tabel warna yang didefinisikan sebagai array dan struktur RGBQUAD dan sisanya adalah data gambar. Format ini mendukung resolusi dan monokrom hingga tiga ratus (16.7 juta warna).

2.7.4. Format Citra GIF

GIF (*Graphics Interchange Format*) merupakan format gambar standar yang mengunakan spesifikasi kemampuan seperti kesiapan gambar

(*sharp*) resolusi gambar (*resolution*) yang baik serta ukuran *file* gambar yang relatif lebih kecil. Bagus tidaknya resolusi gambar GIF yang dihasilkan juga tergantung pada perangkat keras (*Graphics Hardware*) yang digunakan. Perangkat keras tersebut harus dapat menghasilkan resolusi warna gambar yang tajam dengan *pixel* warna yang baik.

Pixel merupakan elemen yang terkecil dalam sebuah gambar. *Pixel* berhubungan dengan titik (*dot*) dalam layar grafik. Misalnya sebuah gambar memiliki ukuran sebesar 320 x 320. Hal ini mempunyai arti bahwa gambar tersebut mempunyai jumlah *pixel* sebanyak 320 x 320. Semakin banyak *pixel* yang terdapat dalam objek gambar, maka semakin halus gambar tersebut karena jarak antar *pixel* semakin rapat dan semakin besar ukuran *file* tersebut. Begitu juga sebaliknya, dimana semakin sedikit jumlah *pixel* yang terdapat pada objek gambar, maka tampilan gambar tersebut terlihat kasar karena jarak antar *pixel* renggang dan ukuran *file* yang dihasilkan lebih kecil.

Format GIF ditujukan untuk mendukung format gambar saat ini dan format gambar yang akan datang. Format GIF mempunyai ekstensi .gif yang didukung oleh aplikasi grafis, misalnya *Microsoft Paint*, *Corel Draw*, *Corel PhotoPaint*, *Adobe PageMaker* dan sebagainya.

2.8. Visual Basic 6.0

Visual Basic adalah salah satu pemrograman bahasa computer. Bahasa pemrograman adalah perintah-perintah yang dimengerti oleh computer untuk melakukan tugas-tugas tertentu.

Bahasa pemrograman Visual Basic, yang dikembangkan oleh Microsoft sejak tahun 1991, merupakan pengembangan dari pendahulunya, yaitu bahasa pemrograman BASIC (*Beginner's All-purpose Symbolic Instruction Code*) yang dikembangkan pada era 1950-an[13].

Visual Basic merupakan salah satu development tool, yaitu alat bantu untuk membuat berbagai macam program computer dan mudah untuk dipelajari sehingga dapat membuat program dengan aplikasi GUI (Graphical User Interface) atau program yang memungkinkan pemakai computer berkomunikasi dengan computer tersebut dengan menggunakan modus grafik atau gambar[14].

(dpi) resolusi gambar (*resolution*) yang baik serta ukuran file gambar yang relatif lebih kecil. Bagus tidaknya resolusi gambar GIF yang dihasilkan juga tergantung pada pemakaian warna (*palette Warna*) yang digunakan. Pemakaian warna tersebut harus dapat mendukung resolusi warna gambar yang tajam dengan pixel warna yang baik.

Pixel merupakan elemen yang terkecil dalam sebuah gambar. Pixel berhubungan dengan titik (dot) dalam layar grafik. Misalnya sebuah gambar memiliki ukuran sebesar 320 x 320. Hal ini mempunyai arti bahwa gambar tersebut mempunyai jumlah pixel sebanyak 320 x 320. Semakin banyak pixel yang terdapat dalam objek gambar maka semakin baik gambar tersebut karena jarak antar pixel semakin rapat dan semakin besar ukuran file tersebut. Begitu juga sebaliknya, dimana semakin sedikit jumlah pixel yang terdapat pada objek gambar maka tampilan gambar tersebut terlihat kasar karena jarak antar pixel renggang dan ukuran file yang dihasilkan lebih kecil.

Format GIF dirancang untuk memudahkan format gambar saat ini dan bentuk gambar yang akan datang. Format GIF mempunyai ekstensi .gif yang dikenal oleh aplikasi grafis misalnya *Microsoft Paint*, *Corel Draw*, *Corel PhotoPaint*, *Adobe Photoshop* dan sebagainya.

2.6. Visual Basic 6.0

Visual Basic adalah salah satu pemrograman bahasa komputer. Bahasa pemrograman adalah perintah-perintah yang dimengerti oleh komputer untuk melakukan tugas-tugas tertentu.

Bahasa pemrograman Visual Basic yang dikembangkan oleh Microsoft sejak tahun 1991, merupakan pengembangan dari pendahulunya yaitu bahasa pemrograman BASIC (*Beginner's All-purpose Symbolic Instruction Code*) yang diperkenalkan pada era 1950-an[7].

Visual Basic merupakan salah satu development tool yang dapat untuk membuat berbagai macam program komputer dan mudah untuk dipelajari sehingga dapat membuat program dengan aplikasi GUI (*Graphical User Interface*) atau program yang memungkinkan pemakai komputer berkomunikasi dengan komputer tersebut dengan menggunakan media grafik atau gambar[7].

Microsoft Visual Basic 6.0 menyediakan fasilitas yang memungkinkan penggunanya untuk menyusun sebuah program dengan memasang objek-objek grafis dalam sebuah form.Visual Basic 6.0 berawal dari bahasa pemrograman BASIC (Beginners All-purpose Symbolic Instruction Code).

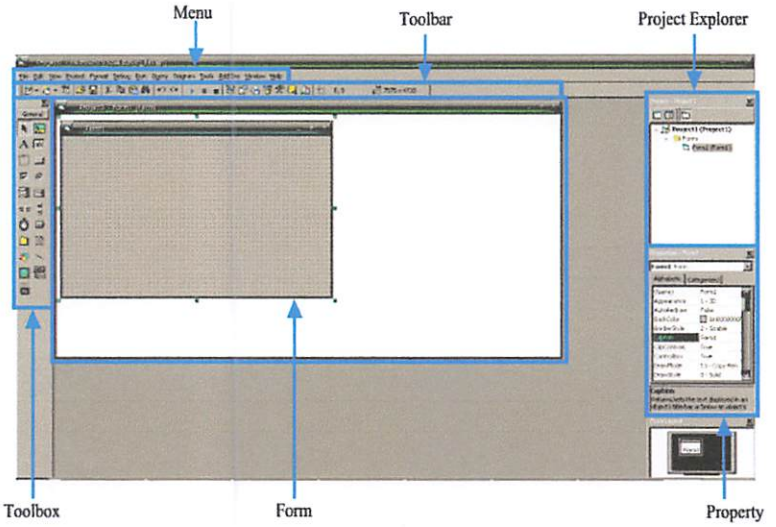
Tahun 1980-an system operasi DOS cukup popular di kalangan pemakai computer karena didalamnya disertakan bahasa BASIC yang dikenal dengan QBASIC (QuickBasic). System tersebut sekarang sudah jarang digunakan sekarang sebagai penyempurnanya pada era Windows, Microsoft menciptakan Visual basic yang terus mengalami penyempurnaan hingga Visual Basic 6.0.

Visual Basic merupakan bahasa pemrograman yang sangat mudah dipelajari, dengan teknik pemrograman visual yang memungkinkan penggunanya untuk berkreasi lebih baik dalam menghasilkan suatu program aplikasi. Ini terlihat dari dasar pembuatan dalam visual basic adalah FORM, dimana pengguna dapat mengatur tampilan form kemudian dijalankan dalam script yang sangat mudah[14].

Ledakan pemakaian Visual Basic ditandai dengan kemampuan Visual Basic untuk dapat berinteraksi dengan aplikasi lain di dalam sistem operasi Windows dengan komponen ActiveX Control. Dengan komponen ini memungkinkan penguna untuk memanggil dan menggunakan semua model data yang ada di dalam sistem operasi windows. Hal ini juga ditunjang dengan teknik pemrograman di dalam Visual Basic yang mengadopsi dua macam jenis pemrograman yaitu Pemrograman Visual dan *Object Oriented Programming* (OOP)[15].

2.8.1. Antar Muka Visual Basic 0.6

Antar muka Visual Basic 6.0, berisi menu, toolbar, toolbox, form, project explorer dan property seperti terlihat pada gambar berikut:

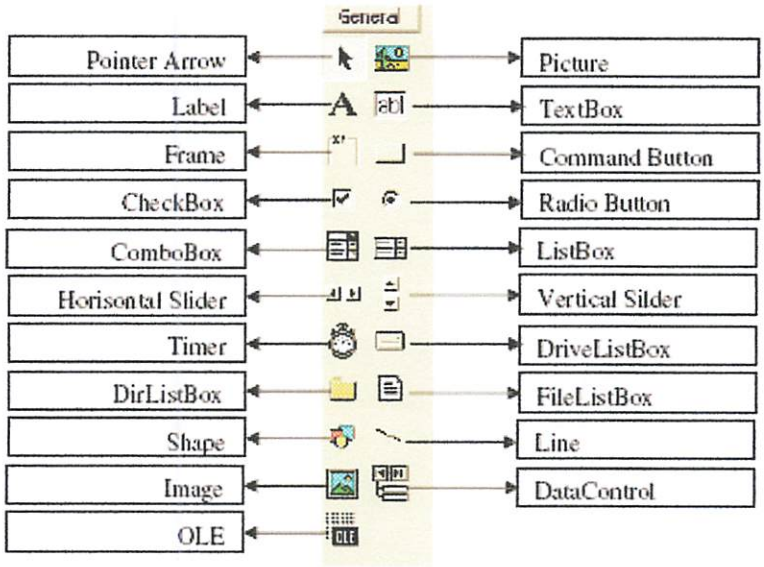


Gambar 2.6. Antar Muka Visual Basic 0.6

Pembuatan program aplikasi menggunakan Visual Basic dilakukan dengan membuat tampilan aplikasi pada form, kemudian diberi script program di dalam komponen-komponen yang diperlukan. Form disusun oleh komponen-komponen yang berada di [Toolbox], dan setiap komponen yang dipakai harus diatur propertinya lewat jendela [Property].

Menu pada dasarnya adalah operasional standar di dalam sistem operasi windows, seperti membuat form baru, membuat project baru, membuka project dan menyimpan project. Di samping itu terdapat fasilitas-fasilitas pemakaian visual basic pada menu. Untuk lebih jelasnya Visual Basic menyediakan bantuan yang sangat lengkap dan detail dalam MSDN.

Toolbox berisi komponen-komponen yang bisa digunakan oleh suatu project aktif, artinya isi komponen dalam toolbox sangat tergantung pada jenis project yang dibangun. Komponen standar dalam toolbox dapat dilihat pada gambar berikut ini.



Gambar 2.7. Toolbox pada Visual Basic 0.6

2.8.2.Konsep Dasar Pemrograman

Konsep dasar pemrograman Visual Basic 6.0, adalah pembuatan form dengan mengikuti aturan pemrograman **Property, Metode** dan **Event**. Hal ini berarti:

1. Property

   Setiap komponen di dalam pemrograman Visual Basic dapat diatur propertinya sesuai dengan kebutuhan aplikasi. Property yang tidak boleh dilupakan pada setiap komponen adalah "**Name**", yang berarti nama variable (komponen) yang akan digunakan dalam scripting. Properti "Name" ini hanya bisa diatur melalui jendela Property, sedangkan nilai peroperti yang lain bias diatur melalui script seperti :

   Command1.Caption="Play"
   Text1.Text="Visual Basic"
   Label1.Visible=False
   Timer1.Enable=True

2. Metode

   Bahwa jalannya program dapat diatur sesuai aplikasi dengan menggunakan metode pemrograman yang diatur sebagai aksi dari setiap komponen. Metode inilah tempat untuk mengekpresikan logika pemrograman dari pembuatan suatu prgram aplikasi.

3. Event

Setiap komponen dapat beraksi melalui event, seperti event click pada command button yang tertulis dalam layar script Command1_Click, atau event Mouse Down pada picture yang tertulis dengan Picture1_MouseDown. Pengaturan event dalam setiap komponen yang akan menjalankan semua metode yang dibuat.

BAB III
PERANCANGAN DAN ANALISA SISTEM

3.1. Perancangan

Perancangan perangkat lunak permainan strategi *Battle Ship* pada jaringan ini melalui beberapa tahapan proses yaitu :

1. **Perancangan Gambar Kapal.**

   Kapal yang digunakan dalam perangkat lunak ini terdiri dari :

   a. *Battleship* dengan ukuran 4 kotak sebanyak 1 buah.
   b. *Cruiser* dengan ukuran 3 kotak sebanyak 2 buah.
   c. *Submarine* dengan ukuran 2 kotak sebanyak 3 buah.
   d. Destroyer dengan ukuran 1 kotak sebanyak 4 buah.

   Gambar kapal yang digunakan sebenarnya merupakan penggabungan dari beberapa bagian kapal yang telah dipecahkan sebelumnya. Sebuah gambar kapal dipecahkan menjadi 3 bagian, yaitu bagian depan, tengah dan belakang kapal. Bagian-bagian kapal tersebut digabungkan kembali sesuai dengan ukuran dari kapal dengan perincian sebagai berikut :

   a. *Battleship* merupakan penggabungan dari 1 buah bagian depan kapal, 2 buah bagian tengah kapal dan 1 buah bagian belakang kapal.
   b. *Cruiser* merupakan penggabungan dari 1 buah bagian depan kapal, 1 buah bagian tengah kapal dan 1 buah bagian belakang kapal.
   c. *Submarine* merupakan penggabungan dari 1 buah bagian depan kapal dan 1 buah bagian belakang kapal.

   Khusus untuk gambar kapal *Destroyer* yang memiliki panjang 1 kotak saja dirancang dengan menggunakan gambar tersendiri dan bukan merupakan gabungan dari bagian-bagian kapal tersebut.

   Gambar-gambar di atas juga dirancang secara vertikal dengan menggunakan metoda yang sama.

2. **Perancangan Animasi dan Suara.**

   Gambar kapal terbakar, simbol tembakan tepat sasaran dan tembakan gagal dicari pada sumber-sumber di *internet*, kemudian di-*edit* dan disimpan dengan menggunakan aplikasi *Adobe Photoshop*. Efek animasi yang digunakan

BAB III

PERANCANGAN DAN ANALISA SISTEM

3.1. Perancangan

Perancangan peralatan untuk permainan sepatu roda pada jembatan ini meliputi beberapa tahapan proses yaitu :

1. Perancangan Gambar Kapal.

Kapal yang digunakan dalam percobaan untuk ini terdiri dari :

a. Monoship dengan ukuran 4 buah sebanyak 1 buah.
b. Twiner dengan ukuran 3 buah sebanyak 2 buah.
c. Submarine dengan ukuran 2 buah sebanyak 3 buah.
d. Catamaran dengan ukuran 1 buah sebanyak 4 buah.

Gambar kapal yang digunakan sebelumnya merupakan penggabungan dari beberapa bagian kapal yang telah dipersiapkan sebelumnya. Sebuah gambar kapal dipecahkan menjadi 3 bagian, yaitu bagian depan, tengah dan belakang kapal. Bagian-bagian kapal tersebut digabungkan kembali sesuai dengan ukuran dari kapal dengan perincian sebagai berikut :

a. Monoship menggunakan penggabungan dari 1 buah bagian depan kapal, 2 buah bagian tengah kapal dan 1 buah bagian belakang kapal.
b. Twiner menggunakan penggabungan dari 1 buah bagian depan kapal, 1 buah bagian tengah kapal dan 1 buah bagian belakang kapal.
c. Submarine menggunakan penggabungan dari 1 buah bagian depan kapal dan 1 buah bagian belakang kapal.

Khusus untuk gambar kapal Catamaran yang mempunyai penjang 1 buah saja dimasang dengan menggunakan gambar tersendiri dan tidak menggunakan gabungan dari bagian-bagian kapal tersebut.

Gambar-gambar di atas juga dimasang secara vertikal dengan menggunakan mesin yang sama.

2. Perancangan Animasi dan Sensor.

Gambar kapal tersebut sinkron tombolan kapal sesuai dan tombolan gagal diuji pada gambar-gambar di atasnya. Kemudian di-edit dan disimpan dengan menggunakan aplikasi Adobe Premiere. Efek animasi yang digunakan

merupakan proses pergantian gambar. Sedangkan suara ledakan dirancang dengan menggunakan aplikasi *Recorder*.

3. **Perancangan *Interface* untuk Menyusun Kapal..**

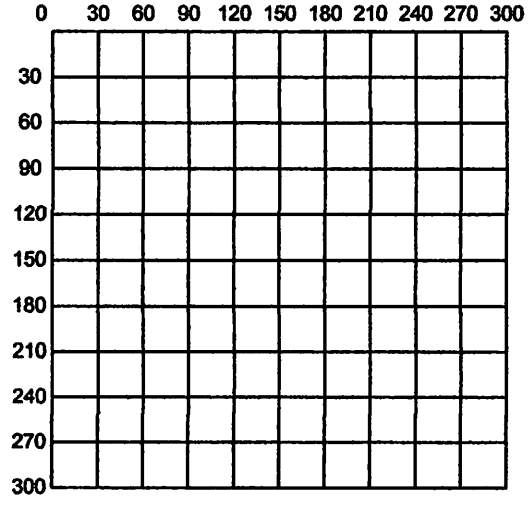
   Interface untuk menyusun kapal dirancang dengan ukuran 10 kotak x 10 kotak. 1 kotak diwakili oleh sebuah *picturebox* dengan gambar air, sehingga *interface* ini merupakan penggabungan dari *picturebox* yang diatur posisinya.

4. **Pengaturan Koneksi Jaringan.**

   Microsoft Winsock berfungsi untuk menghubungkan komputer yang satu dengan yang lainnya dan saling bertukar data dengan menggunakan dua *protocol*, yaitu *User Datagram Protocol (UDP)* dan *Transmission Control Protocol (TCP)*. Perangkat lunak permainan strategi *Battleship* pada jaringan ini menggunakan *protocol* TCP. Prinsip kerjanya adalah salah satu komputer akan menentukan apakah bertindak sebagai *Client* atau *Server*. Komputer yang bertindak sebagai *server* akan dijalankan dahulu dan menunggu komputer yang bertindak sebagai *client* untuk melakukan koneksi. Setelah koneksi berhasil dilakukan, maka kedua komputer itu dapat mengadakan interaksi satu sama lain.

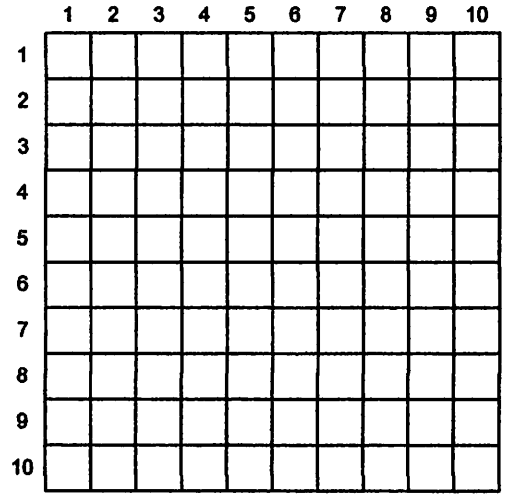
5. **Perancangan Medan Perang.**

   Medan perang dirancang dengan ukuran 10 kotak x 10 kotak. Kotak-kotak ini digambarkan dengan dibatasi oleh garis-garis. Jarak garis-garis tersebut sebanyak 30 *twips* baik secara horizontal maupun vertikal. Sketsa dari gambar medan perang tersebut dapat dilihat pada gambar berikut ini :



Gambar 3.1 Sketsa gambar dari medan perang

Kotak-kotak tersebut akan disimpan dalam bentuk *array* zone2(x,y) dengan cara penomoran kotak tersebut adalah sebagai berikut :



Gambar 3.2 Penomoran kotak pada medan perang

6. **Penentuan Posisi Kapal**

   Kapal yang diletakkan akan ditentukan posisinya dan disimpan ke dalam sebuah variabel *Array* yang memiliki dua buah properti yaitu X untuk menyimpan posisi kolom dari kapal, Y untuk menyimpan posisi baris dari kapal. Proses perhitungan posisi x dan y sebagai berikut :

   X = Posisi_Kursor_X div 30

   Y = Posisi_Kursor_Y div 30

7. **Pengecekan Posisi Tembakan dan Kapal.**

   Jika posisi (X,Y) yang diklik sama dengan salah satu dari posisi kapal yang disimpan dalam *array* tersebut maka berarti kapal tertembak dan posisi bagian kapal yang tertembak tersebut ditandai. Posisi koordinat yang ditembak juga ditandai. Proses perhitungan posisi x dan y tersebut sama dengan proses perhitungan pada penentuan posisi kapal di atas.

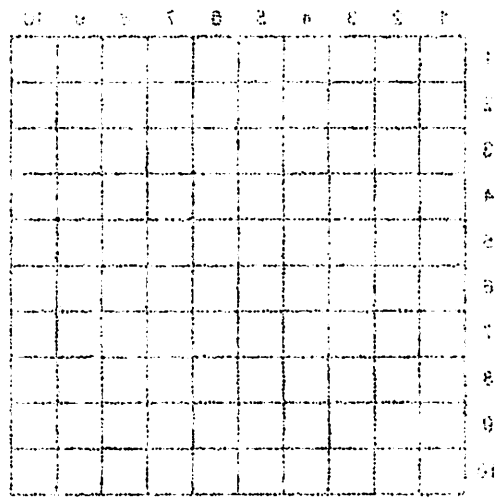
8. **Penentuan Pemenang.**

   Jika jumlah bagian kapal yang ditandai sama dengan 20 maka berarti semua kapal telah tertembak, maka pemain lawan dinyatakan menang.

9. **Perhitungan *Score*.**

   Jika sebuah kapal telah ditenggelamkan maka *score* pemain akan ditambahkan. Besar *score* yang didapatkan dihitung berdasarkan ukuran kapal. Ukuran 1 kotak bernilai 10 point. Jadi, menenggelamkan sebuah kapal berukuran 3 kotak, maka pemain akan mendapatkan nilai 30 point.

Kotak-kotak tersebut akan dibangun dalam bentuk array zona(x,y) dengan cara pemrograman kotak tersebut adalah sebagai berikut :



Gambar 5.2 Pemrograman kotak pada modus permainan

6. **Penentuan Posisi Kapal**

Kapal yang dilewatkan akan ditentukan posisinya dan dibimpan ke dalam sebuah variabel array yang mempunyai dua buah properti yaitu X untuk menyimpan posisi kolom dari kapal, Y untuk menyimpan posisi baris dari kapal. Proses perhitungan posisi x dan y sebagai berikut :

X = Posisi_Kursor_X div 30

Y = Posisi_Kursor_Y div 30

7. **Pengecekan Posisi Tembakan dan Kapal.**

Jika posisi (X,Y) yang diklik sama dengan salah satu dari posisi kapal yang dibimpan dalam array tersebut maka berarti kapal tersebut dan posisi bagian kapal yang tertembak tersebut ditandai. Posisi koordinat yang ditandai juga ditandai. Proses perhitungan posisi x dan y tersebut sama dengan proses perhitungan pada penentuan posisi kapal di atas.

8. **Penentuan Pemenang.**

Jika jumlah bagian kapal yang ditandai sama dengan 20 maka berarti semua kapal telah tertembak, maka permain lawan dinyatakan menang.

9. **Perhitungan Score.**

Jika sebuah kapal telah dilengkapkan maka score pemain akan ditambahkan. Hasil score yang didapatkan dikurang banderasekan skorsu kapal. Ukuran 1 kotak bernilai 10 point, tiap mengapalkan sebuah kapal berukuran 3 kotak, maka pemain akan mendapatkan nilai 30 point.

3.2 Flowchart

Flowchart adalah penggambaran secara grafik dari langkah-langkah dan urut-urutan prosedur dari suatu program. Flowchart menolong analis dan programmer untuk memecahkan masalah kedalam segmen-segmen yang lebih kecil dan menolong dalam menganalisis alternatif-alternatif lain dalam pengoperasian.

Flowchart biasanya mempermudah penyelesaian suatu masalah khususnya masalah yang perlu dipelajari dan dievaluasi lebih lanjut.

3.2.1. Flowchart Sistem

Flowchart Sistem merupakan bagan yang menunjukkan alur kerja atau apa yang sedang dikerjakan di dalam sistem secara keseluruhan dan menjelaskan urutan dari prosedur-prosedur yang ada di dalam sistem. Dengan kata lain, flowchart ini merupakan deskripsi secara grafik dari urutan prosedur-prosedur yang terkombinasi yang membentuk suatu sistem.

Flowchart Sistem terdiri dari data yang mengalir melalui sistem dan proses yang mentransformasikan data itu. Data dan proses dalam flowchart sistem dapat digambarkan secara *online* (dihubungkan langsung dengan komputer) atau *offline* (tidak dihubungkan langsung dengan komputer, misalnya mesin tik, *cash register* atau kalkulator).

3.2.2. Simbol – Simbol Flowchart

Simbol-simbol flowchart yang biasanya dipakai adalah simbol-simbol flowchart standar yang dikeluarkan oleh ANSI dan ISO.

SIMBOL	ARTI	CONTOH
Input / Output	Merepresentasikan Input data atau Output data yang diproses atau Informasi.	Baca Jam & Tarif Upah
Proses	Mempresentasikan operasi	Hitung Upah Kotor
Penghubung	Keluar ke atau masuk dari bagian lain flowchart khususnya halaman yang sama	3 Keluar 3 Masuk
Anak Panah	Merepresentasikan alur kerja	Hitung Upah Kotor 3
Penjelasan	Digunakan untuk komentar tambahan	Urutkan Sebelum Pembayaran Berdasarkan No. Pelanggan

SIMBOL	ARTI	CONTOH
Keputusan	Keputusan dalam program	Apakah A < B = > <, >, =
Predefined Process	Rincian operasi berada di tempat lain	Hitung Akar Pangkat Dua
Preparation	Pemberian harga awal	SW = 1
Terminal Points	Awal / akhir flowchart	Start
Punched card	Input / outuput yang menggunakan kartu berlubang	Kartu Absen

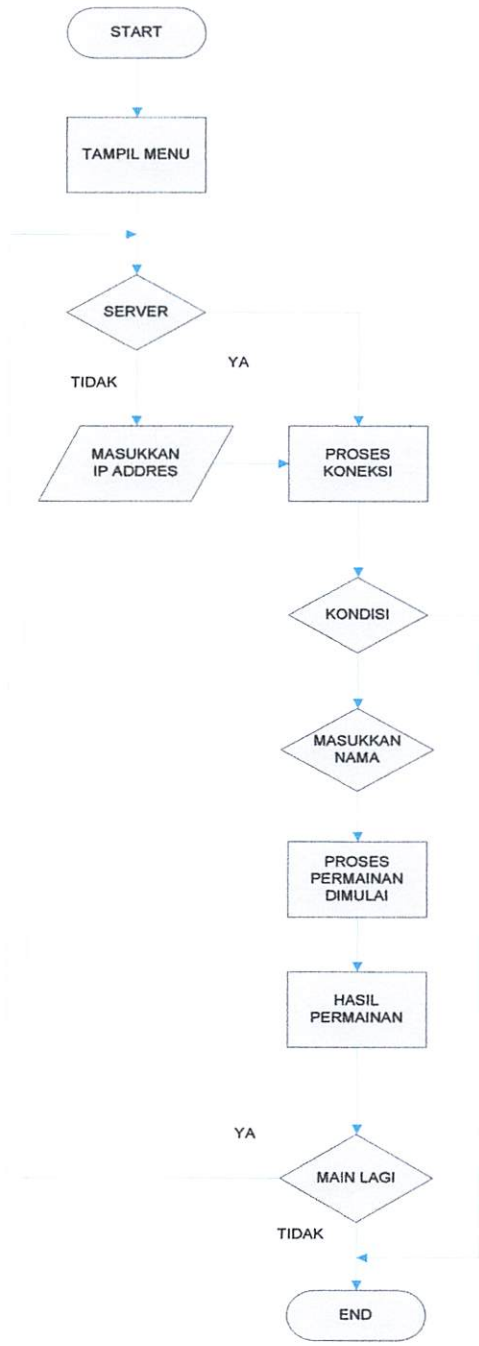
SIMBOL	ARTI	CONTOH
Dokumen	I/O dalam format yang dicetak	Cetak Slip Upah
Magnetic Tape	I/O yang menggunakan pita magnetik	File Upah
Magnetic Disk	I/O yang menggunakan disk magnetik	Update File Pegawai
Magnetic Drum	I/O yang menggunakan drum magnetik	File Pegawai

SIMBOL	ARTI	CONTOH
On-line Storage	I/O yang menggunakan penyimpanan akses langsung	File Pelanggan
Punched Tape	I/O yang menggunakan pita kertas berlubang	File Pelanggan
Manual Input	Input yang dimasukkan secara manual dari keyboard	Masukkan Tarif Upah
Display	Output yang ditampilkan pada terminal	Pesan Kesalahan
Manual Operation	Operasi Manual	Ketik Pesanan Penjualan

SIMBOL	ARTI	CONTOH
Communication Link	Transmisi data melalui channel komunikasi, seperti telepon	Komputer Terminal
Off-line Storage	Penyimpanan yang tidak dapat diakses oleh komputer secara langsung	

Tabel 3.1 Simbol – Simbol Flowchart

Berikut ini adalah Flowchart proses kinerja sistem Perangkat lunak permainan (*game*) strategi B*attle ship* seperti pada gambar 3.1 dibawah ini :



Gambar 3.1 Flowchart Permainan (Game) Strategi Multiplayer Battle Ship

3.3 Analisa Sistem

Aplikasi ini dibangun dengan menggunakan bahasa pemrograman Visual Basic 6.0, dan di *compile* dalam bentuk '.exe' agar lebih mudah dalam pengimplementasiannya. Berikut ini adalah algoritma perancangan perangkat lunak permainan strategi '*Battle Ship*' pada jaringan *WiFi*.

3.3.1. Algoritma

Algoritma perancangan perangkat lunak permainan strategi '*Battle Ship*' pada jaringan dibagi menjadi 6 bagian yaitu,

1. Algoritma Pengaturan Koneksi Antara 2 (Dua) Pemain.
2. Algoritma Tembak Kapal Lawan.
3. Algoritma Ubah Giliran Pemain.
4. Algoritma Pengecekan Permainan Berakhir.

3.3.2. Algoritma Pengaturan Koneksi Antara 2 (Dua) Pemain

Algoritma ini menjadi inti dari proses untuk menghubungkan komunikasi antara dua komputer pada sebuah jaringan lokal. Penulis menggunakan komponen *visual basic* '*winsock*' untuk mengirim dan menerima data antar komputer.

Prosedur kerja algoritma ini adalah komputer yang berperan sebagai *server*, menjalankan *program* terlebih dahulu. Ini berarti *winsock* pada komputer tersebut berperan sebagai *server*. *Winsock server* akan melakukan prosedur *listening*, yaitu aksi mendengarkan permintaan koneksi dari komputer lain. Algoritmanya adalah sebagai berikut,

{*Tutup winsock untuk menjaga kemungkinan winsock sedang terbuka*}

```
Winsock.Close
```

{*Local Port Default yang dipakai = 1234*}

```
Winsock.LocalPort = 1234
```

{*Winsock server melakukan aksi listening*}

```
Winsock.Listen
```

Selanjutnya *winsock server* menunggu koneksi dari *winsock client*. *Winsock client* pada komputer lain melakukan koneksi ke *winsock server* dengan meng-*input ip-address* komputer *server* yang dituju. Algoritmanya adalah sebagai berikut,

3.3 Analisa Sistem

Aplikasi ini dibangun dengan menggunakan bahasa pemrograman Visual Basic 6.0, dan di compile dalam bentuk "exe" agar lebih mudah dalam penginstalannya. Berikut ini adalah algoritma permainan perangkat lunak permainan strategi "Battle Ship" pada jaringan WAN.

3.3.1. Algoritma

Algoritma permainan perangkat lunak permainan strategi "Battle Ship" pada jaringan dibagi menjadi 6 bagian yaitu.

1. Algoritma Pengaturan Koneksi Antara 2 (Dua) Komputer
2. Algoritma Tembak Kapal Lawan.
3. Algoritma Lihat Posisi Permainan.
4. Algoritma Pengecekan Permainan Berakhir.

3.3.2. Algoritma Pengaturan Koneksi Antara 2 (Dua) Pemain

Algoritma ini menjadi inti dari proses untuk mengintegrasikan komunikasi antara dua komputer pada sebuah jaringan lokal. Pemain menggunakan komputer server dan client "winsock" untuk mengirim dan menerima data antar komputer.

Proses kerja algoritma ini adalah komputer yang berperan sebagai server mendengarkan pengiriman terlebih dahulu. Ini berarti winsock pada komputer tersebut berperan sebagai server. Winsock server akan melakukan proses listening, yaitu ada mendengarkan permintaan koneksi dari komputer lain. Algoritmanya adalah sebagai berikut.

```
{Tunggu winsock untuk menyiapkan kemungkinan winsock sedang terkoneksi}
Winsock.Close
{Local Port (Nomor port yang digunakan - 1234)}
Winsock.LocalPort = 1234
{Winsock server mendengarkan coba koneksi}
Winsock.Listen
```

Selanjutnya winsock server menunggu koneksi dari winsock client. Winsock client pada komputer lain melakukan koneksi ke winsock server dengan menggunakan nomor-address komputer server yang dituju. Algoritmanya adalah sebagai berikut.

{Tutup winsock untuk menjaga kemungkinan winsock sedang terbuka}

```
Winsock.Close
```

{Koneksi ke server. Local Port Default yang dipakai = 1234}

```
Winsock.Connect <ip-address server>, 1234
```

Apabila *winsock server* yang dituju oleh *winsock client* aktif, maka *winsock server* akan menangkap permintaan koneksi dari *winsock client*, dan menerima koneksi tersebut. Algoritmanya adalah sebagai berikut,

{Event 'ConnectionRequest' pada komponen winsock server akan aktif apabila ada permintaan koneksi dari winsock client}

```
Private Sub Winsock_ConnectionRequest(ByVal requestID As
Long)
```

{Tutup winsock untuk menjaga kemungkinan winsock sedang terbuka}

```
Winsock.Close
```

{Terima koneksi dari client}

```
Winsock.Accept requestID
End Sub
```

3.3.3. Algoritma Tembak Kapal Lawan

Pada saat kita berusaha menembak kapal pada papan permainan lawan, terdapat dua kemungkinan yaitu, kapal lawan berhasil ditembak atau tembakan kita gagal. Ketika kapal lawan berhasil ditembak, maka *program* menjalankan animasi ledakan pada kapal dan animasi api yang berkobar pada bagian kapal yang tertembak. Apabila tembakan gagal, maka *program* akan menjalankan animasi gelombang air pada posisi yang ditembak. Algoritmanya adalah sebagai berikut,

{Tembak pada posisi x, y}

```
cX = Left(msg, InStr(1, msg, ",") - 1)
cY = Mid(msg, InStr(1, msg, ",") + 1)
```

{Koordinat Temporary}

```
tempX = cX
tempY = cY
```

{Jika terkena tembakan}

```
If zone1(cX, cY) > 0 And zone1(cX, cY) < 5 Then
```

```
{Tunggu sebuah untuk menerima kemungkinan sebanyak satunya berhasil}
Winsock.Close
{Koneksi ke server. Local Port dibiarkan yang dipakai = 1234}
Winsock.Connect "ip-address server", 1234
```

Apabila sebuah server yang dituju oleh sebuah client aktif, maka sebuah server akan memunggu permintaan koneksi dari sebuah client, dan menerima koneksi tersebut. Algoritmanya adalah sebagai berikut.

```
{Event 'ConnectionRequest' pada komponen sebuah server dan client/peminta
ada permintaan koneksi dari sebuah client}
Private Sub Winsock_ConnectionRequest(ByVal requestID As
Long)
{Tunggu sebuah untuk menerima kemungkinan sebanyak satunya
berhasil}
Winsock.Close
{Terima koneksi dari client}
Winsock.Accept requestID
End Sub
```

3.3.3. Algoritma Tombak Kapal Laut

Pada saat kita bermain monembak kapal pada permainan lawan, terdapat dua kemungkinan yaitu, kapal lawan berhasil ditembak atau tembakan kita gagal. Kedua kapal lawan berhasil ditembak, maka pemain menjalankan animasi ledakan pada kapal dan animasi api yang berkobar pada bagian kapal yang tertembak. Apabila tembakan gagal, maka pemain akan menjalankan animasi gelombang air pada posisi yang ditembak. Algoritmanya adalah sebagai berikut.

```
{Tembak pada posisi x, y}
x = Int(Rnd * Int(Text1.Text) + 1)
y = Int(Rnd * Int(Text2.Text) + 1)

{Koordinat Tampung}
temp1 = x
temp2 = y

{Jika terkena tembakan}
If Kapal(x, y) = 1 Then Kapal(x, y) = 2
```

```
zone1(cX, cY) = zone1(cX, cY) + 4

{Cari head_x kapal}
Do
    tempX = tempX - 1
    If tempX < 0 Then
        Exit Do
    End If
Loop Until zone1(tempX, cY) = 0
headX = tempX + 1

{Cari head_y kapal}
Do
    tempY = tempY - 1
    If tempY < 0 Then
        Exit Do
    End If
Loop Until zone1(cX, tempY) = 0
headY = tempY + 1

{Cari kapal dengan head_x dan head_y}
For i = 1 To 10
    If ships(i).headX = headX And ships(i).headY = headY
Then
        {Apakah kapal sudah tenggelam}
        r = ships(i).cripple()

        {R = True, kapal tenggelam}
        If r Then

            {Cek apakah permainan sudah berakhir}
            If isOver() Then
                {Send data ke lawan - menang}
                Winsock.SendData "Result:Win"
            Else
                {Send data ke lawan – kapal tenggelam}
                Winsock.SendData "Result:Dead"
            End If
        Else
```

```
                {Send data ke lawan – kapal tertembak}
                Winsock.SendData "Result:Wound"
            End If

            zone1flame(curFlame).frame = 0
            Randomize
            zone1flame(curFlame).typ = Int(Rnd() * 3) + 1
            zone1flame(curFlame).x = cX
            zone1flame(curFlame).y = cY
            getBackImage cX, cY, zone1flame(curFlame).backImage,
_
        zone1flame(curFlame).backImageMask
            curFlame = curFlame + 1
            Exit For
        End If
    Next i

Else {Tembakan gagal / tidak kena}

    {Gambar air pada daerah yang tidak terkena tembakan}
    BitBlt field1.hDC, cX * 30, cY * 30, tile.ScaleWidth, _
tile.ScaleHeight, waterMiss.hDC, 0, 0, vbSrcCopy

    {Send data ke lawan – bahwa tembakan gagal}
    Winsock.SendData "Result:Miss"
End If

{Ganti giliran pemain}
changeTurn
```

3.3.4. Algoritma Ubah Giliran Pemain

Algoritma ini berfungsi untuk mengubah giliran bermain pemain. Pemain tidak dapat menembak jika bukan gilirannya bermain. Algoritmanya adalah sebagai berikut,

```
'Prosedur giliran
Sub changeTurn()

{Ubah giliran}
user1 = Not user1
user2 = Not user2
```

```
field2.Enabled = Not field2.Enabled

'user1 = true?
If user1 Then
    {Giliran pemain bermain}
    lblState.Caption = " Giliran Anda !"

    {Hidupkan timer the turnTaggle untuk berkedip}
    turnTaggle.Enabled = True
Else
    {Giliran lawan bermain}
    lblState.Caption = "Giliran " & user & " !"

    {Matikan timer the turnTaggle untuk berkedip}
    turnTaggle.Enabled = False
    lblState.BackColor = &HF0C4A6

End If
End Sub
```

3.3.5.Algoritma Pengecekan Permainan Berakhir

Permainan berakhir apabila kapal dari salah satu pemain telah tertembak semuanya. Algoritmanya adalah sebagai berikut,

```
'Apakah permainan sudah berakhir
Function isOver() As Boolean

Dim i As Integer

isOver = True

{Looping – cek apakah semua kapal telah tenggelam}
For i = 1 To 10
    If Not ships(i).isDead Then
        isOver = False
        Exit Function
    End If
Next i
End Function
```

BAB IV

IMPLEMENTASI DAN PENGUJIAN SISTEM

4.1. Implementasi Sistem

Pada bab ini akan ditunjukan tentang pengujian aplikasi yang telah diibuat, ini semua bertujuan untuk mengetahui sejauh mana keberhasilan aplikasi ini bila diterapkan dalam sistem.

Pengujian ini juga untuk mencari letak keberhasilan program dijalankan dengan menggunakan winsock dan mengukur jarak pada koneksi wifi dalam aplikasi.

Berikut ini adalah Pengujian Winsock sebagai penghubung jaringan pada Perangkat lunak permainan (*game*) *Battle ship.*

Test Koneksi	Panggilan Dari Server Yang Terkoneksi	
	Client 1	Client 2
Client yang terkoneksi	✓	
Migabite Persecond (Mbps)	54 Mbps	54 Mbps

Tabel 4.1 Pengujian Koneksi Pada Winsock

Berikut ini adalah Pengujian jarak ukur sejauh mana proses koneksi WiFi pada Perangkat lunak permainan (*game*) *Battle ship* masih berjalan.

NO	Jarak Jaringan WiFi	Jaringan Dapat Terkoneksi WiFi		Megabite Per Second (Mbps)
		YA	TIDAK	
1	1 – 5 Meter	✓		54
2	5 – 10 Meter	✓		48

3	10 – 15 Meter	✓		11
4	ANTAR RUANG 5 meter	✓		54
5	ANTAR RUANG 10 meter	✓		48
6	ANTAR RUANG 15 meter	✓		11
7	ANTAR GEDUNG		✓	—
8	1 LANTAI	✓		9,5
9	2 LANTAI		✓	—
10	3 LANTAI		✓	—

Tabel 4.2. Pengujian Koneksi aplikasi Game Battle Ship pada jaringan WiFi

4.2. Pengujian Sistem

Pengujian sistem dalam perangkat lunak permainan (game) battle ship ini mencakup spesifikasi kebutuhan perangkat keras (*hardware*) dan spesifikasi perangkat lunak (*software*).

4.2.1. Spesifikasi Perangkat Keras dan Perangkat Lunak

Program ini dijalankan dengan menggunakan perangkat keras (*hardware*) yang direkomendasikan sebagai berikut :

1. *Prosesor Intel Pentium IV 2.4 GHz.*
2. *Memory 128 MB.*
3. *Harddisk 40 GB, dengan freespace 200 MB.*
4. *VGA card 32 MB.*
5. *Monitor dengan resolusi 800 X 600 pixel.*
6. *Keyboard dan Mouse*
7. *Wireless LAN device Card* – 54 MBps dan adanya jaringan WiFi.

Adapun perangkat lunak (*software*) yang direkomendasikan untuk menjalankan aplikasi ini adalah lingkungan sistem operasi NT/2000/XP/VISTA/WINDOWS 7.

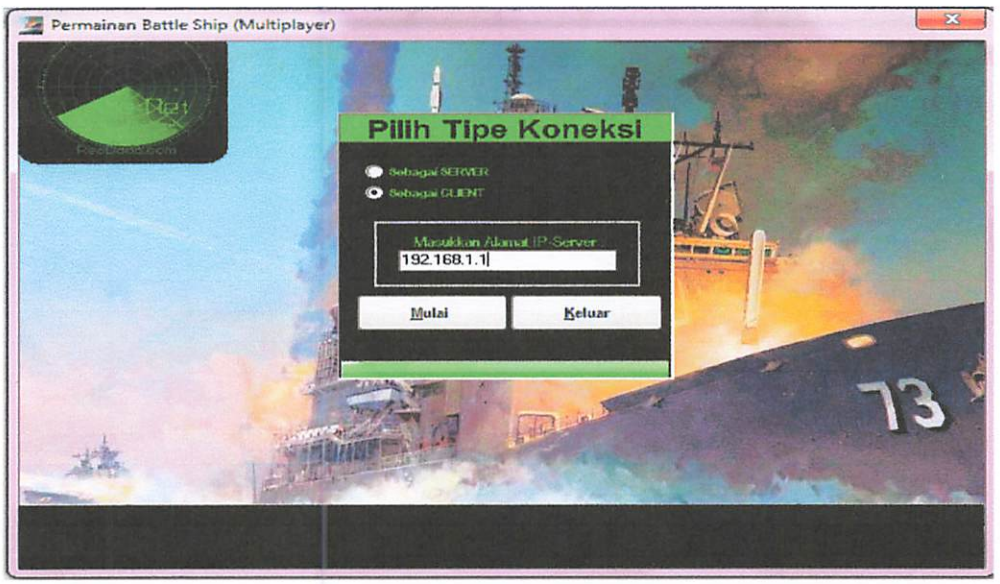
4.2.2. Tampilan Output

Klik pada *file* '*Permainan Battle-Ship (Multiplayer).exe*' untuk menjalankan perangkat lunak. Setelah itu akan muncul tampilan *Splash Screen*. Klik pada *form* atau tekan '*Enter*' untuk masuk ke dalam *program*.

Gambar 4.1 Tampilan *Form Splash Screen*

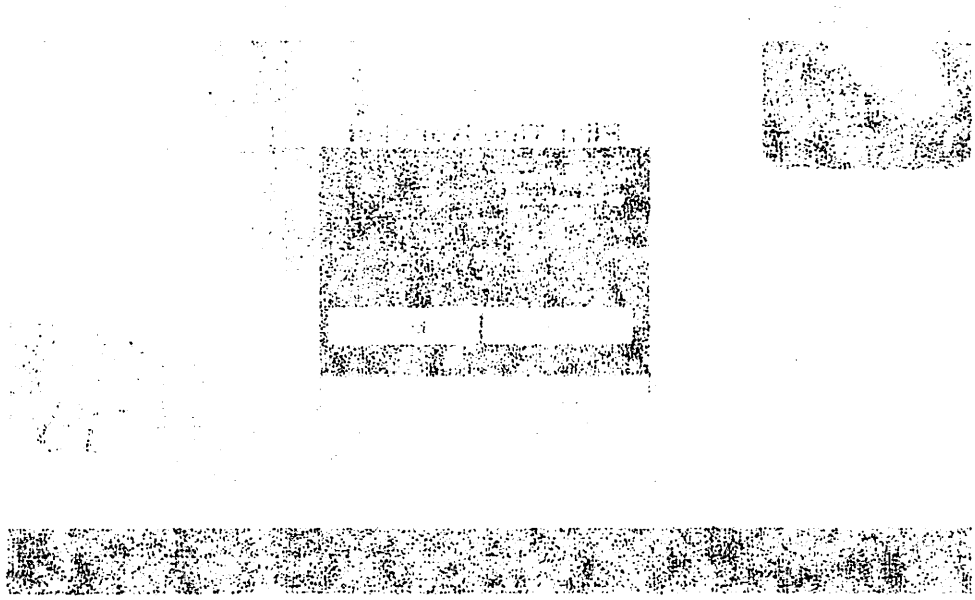
Setelah itu, akan muncul tampilan '*Setting* Koneksi'. Apabila anda memilih koneksi sebagai *server*, maka anda harus menunggu koneksi dari pemain *client*. Apabila anda memilih koneksi sebagai *client*, maka anda harus memasukkan *ip-address* komputer *server* yang dituju.

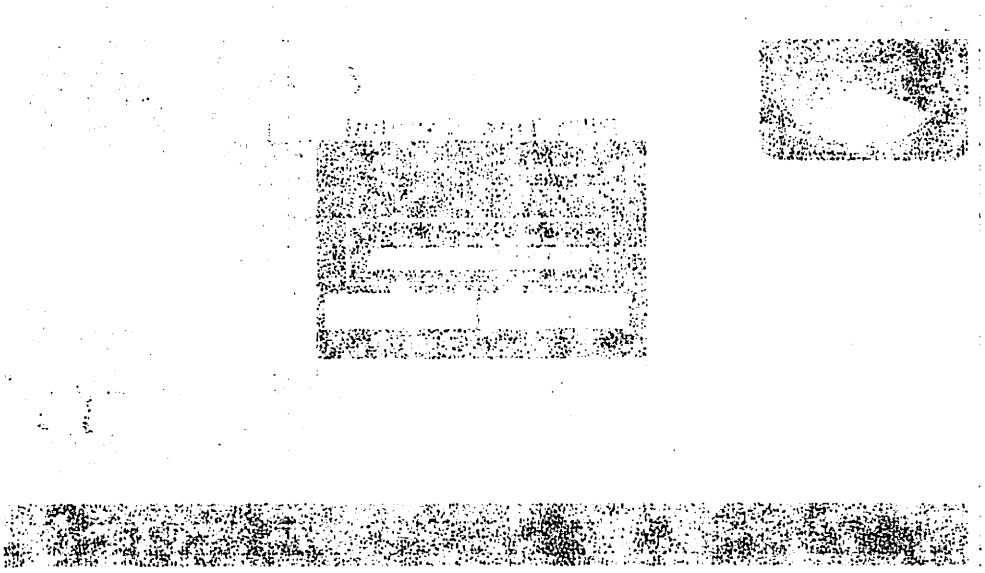
Gambar 4.2 Tampilan *Setting* Koneksi pada Server

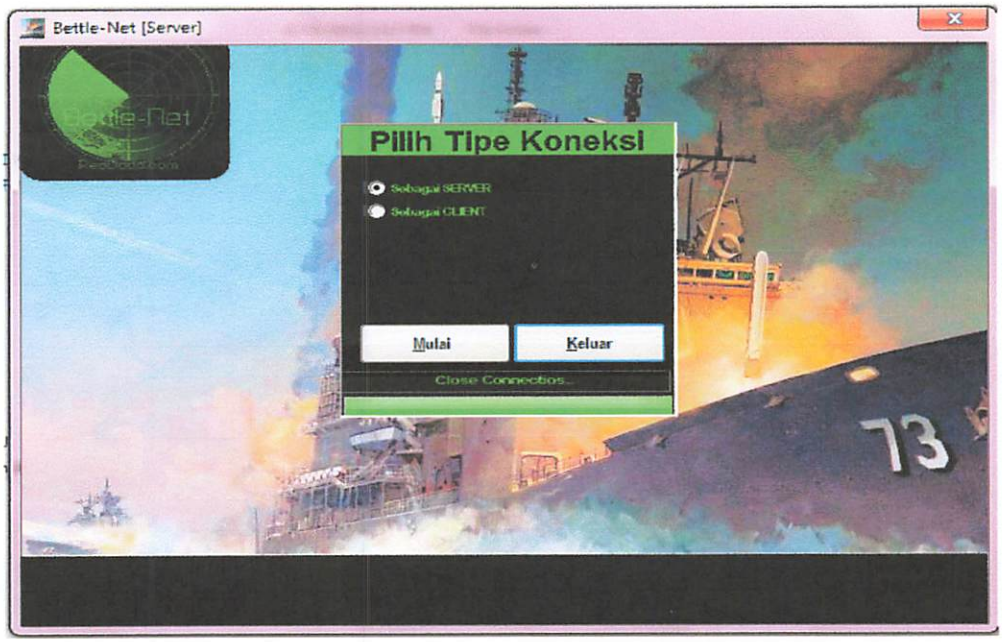


Gambar 4.3 Tampilan *Setting* Koneksi pada client

Apabila anda memilih setting client dan lawan anda sama – sama memilih client ataupun sama – sama server maka akan muncul tampilan pesan seperti gambar dibawah ini.

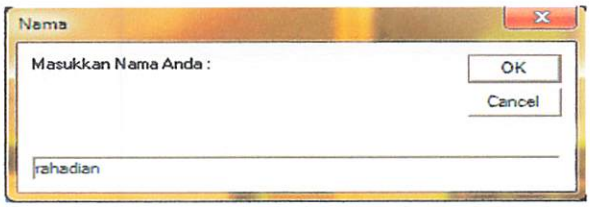






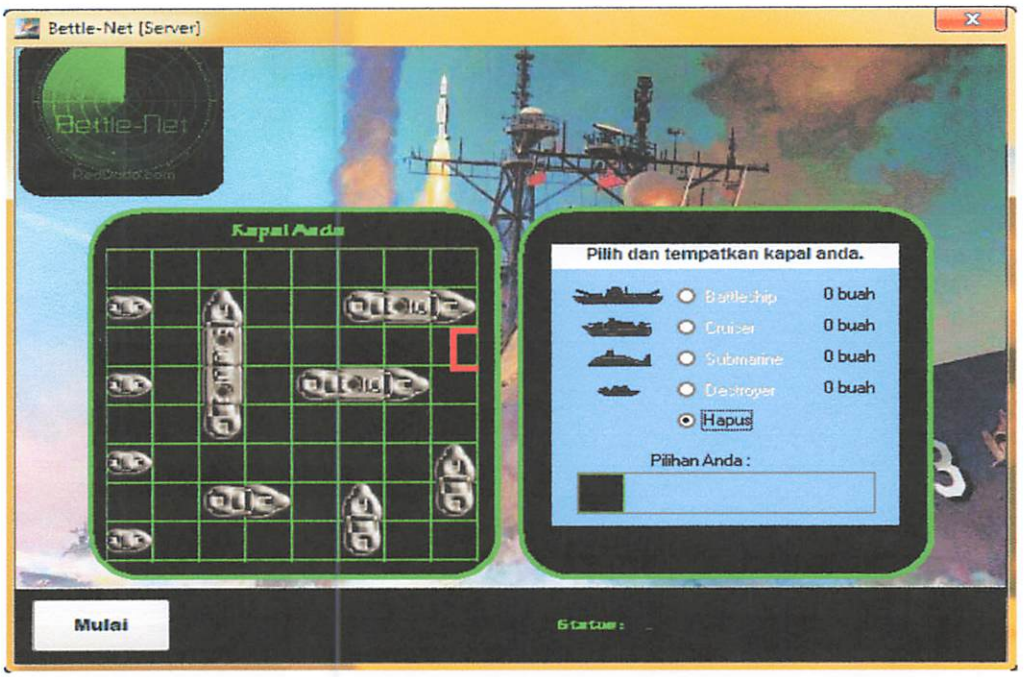
Gambar 4.4 Tampilan jaringan tidak terkoneksi

Apabila koneksi *winsock* telah terhubung, maka akan muncul tampilan *input form* untuk meng-*input* nama pemain.



Gambar 4.5 Tampilan *Input Form* Nama

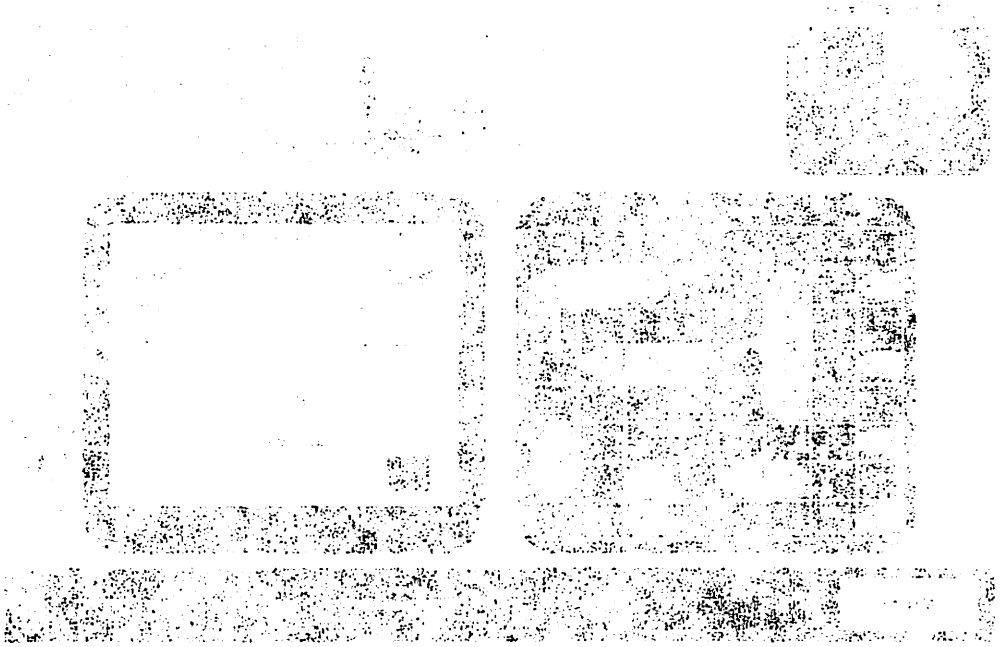
.Setelah *input* nama pemain, maka akan muncul tampilan 'Pengaturan Letak Kapal'. Arah letak kapal dapat diatur secara horizontal atau vertikal. Klik kanan *mouse* pada daerah penempatan kapal untuk mengubah arah horizontal menjadi vertikal, dan sebaliknya.



Gambar 4.6 Tampilan Pengaturan Letak Kapal

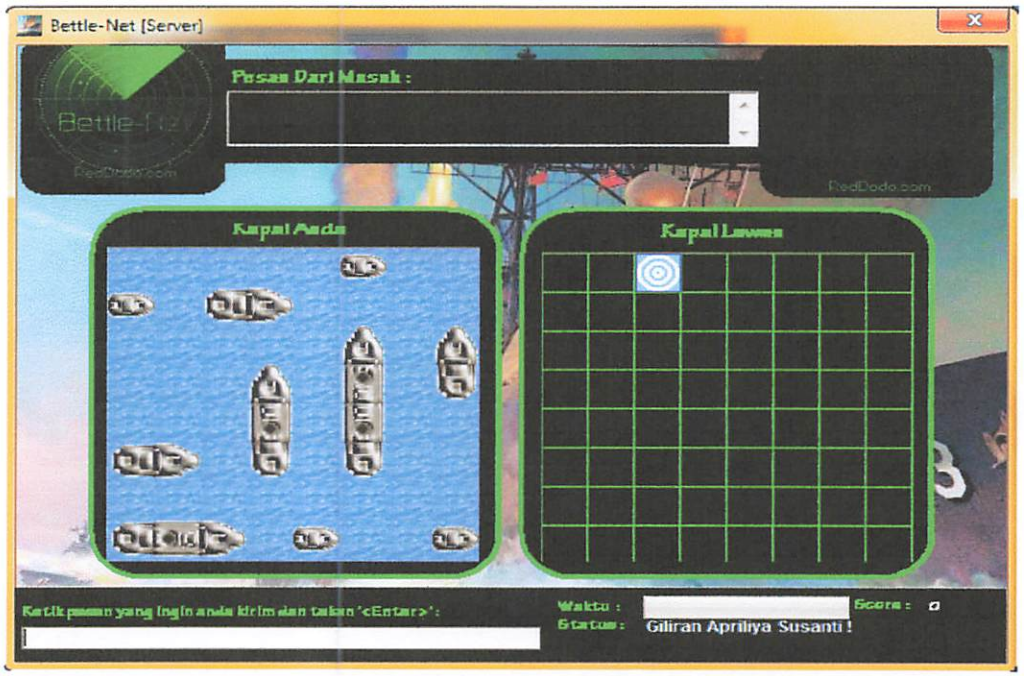
Permainan akan dimulai setelah kedua pemain klik tombol 'Mulai'. Selanjutnya, muncul tampilan medan perang.

Gambar 4.7 Tampilan Medan Perang

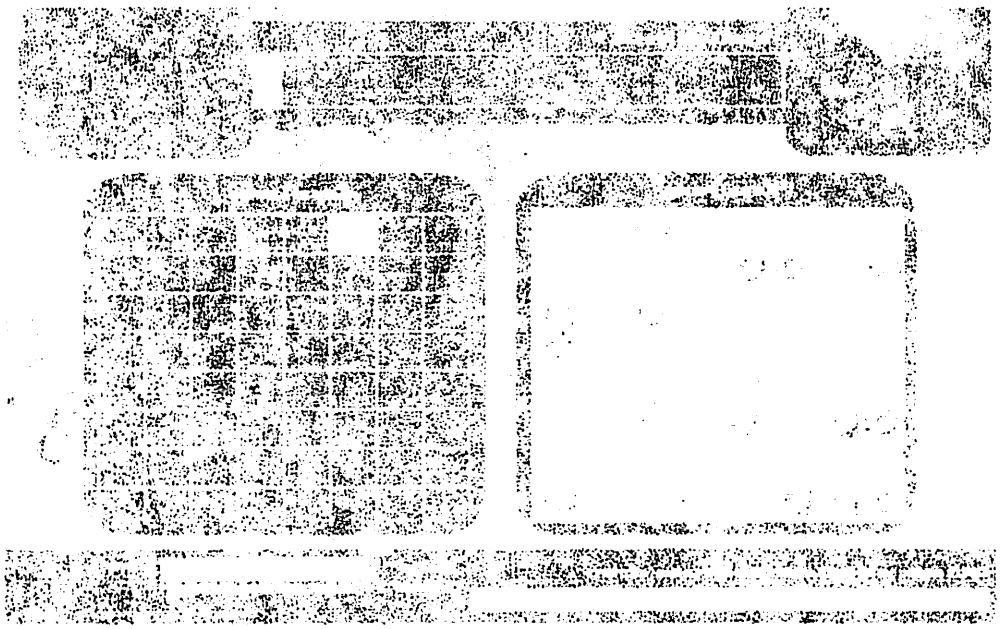


Daerah kotak sebelah kiri merupakan letak kapal anda, sedangkan daerah kotak sebelah kanan merupakan letak kapal lawan. Pada giliran anda, klik kiri pada kotak sebelah kanan untuk menembak kapal lawan.

Apabila tembakan gagal (tidak mengenai kapal lawan), maka akan muncul animasi gelombang air pada daerah kotak sebelah kanan.

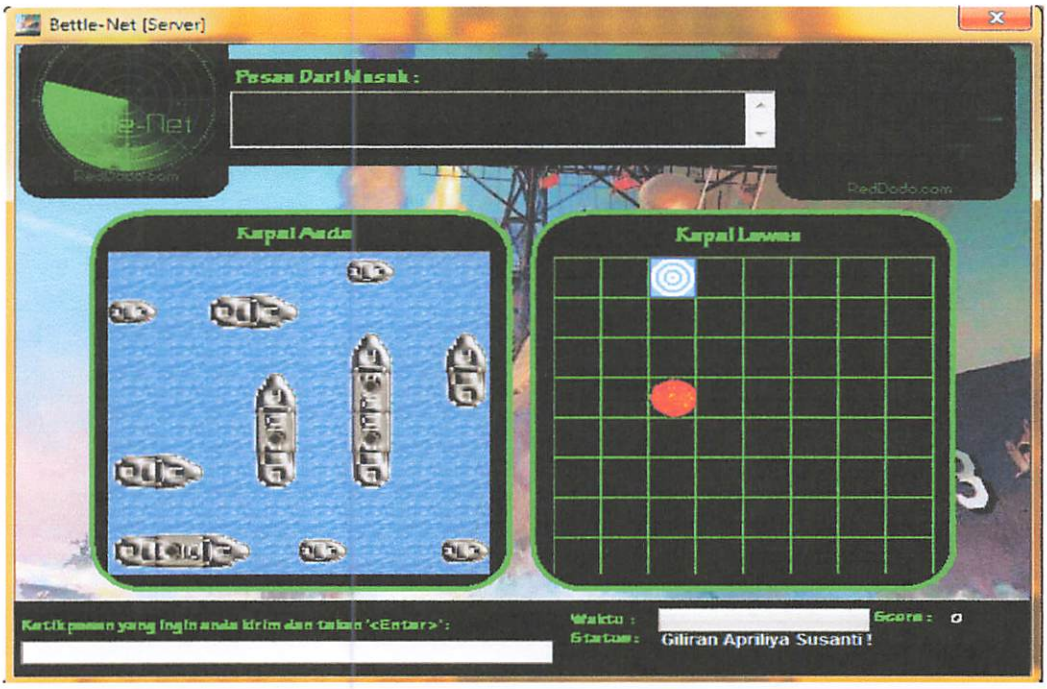


Gambar 4.8 Tampilan Tembakan Gagal Pada Komputer-1 (Pemain yang menembak)

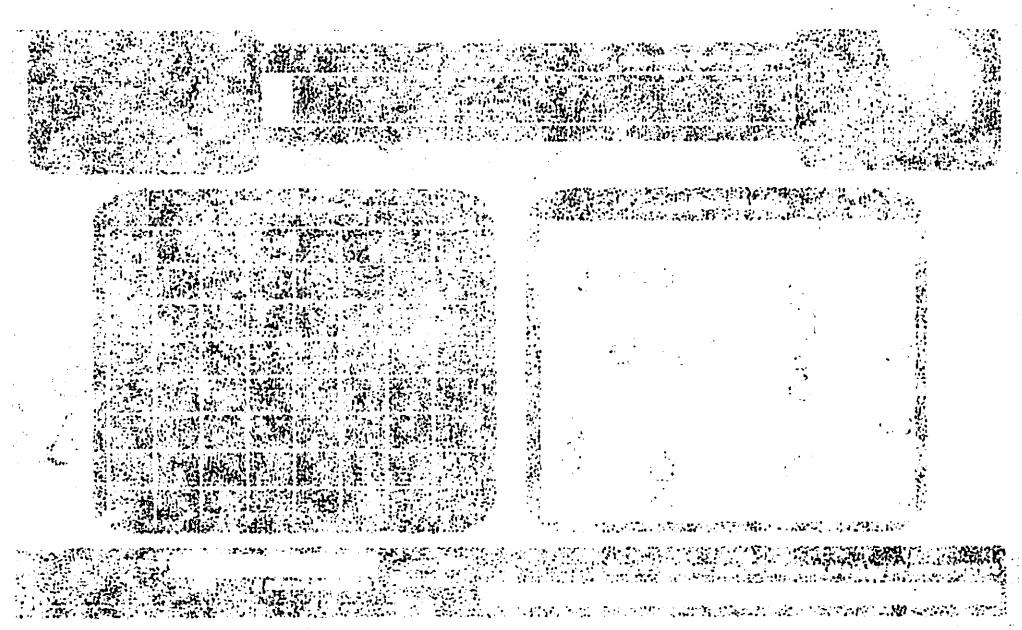


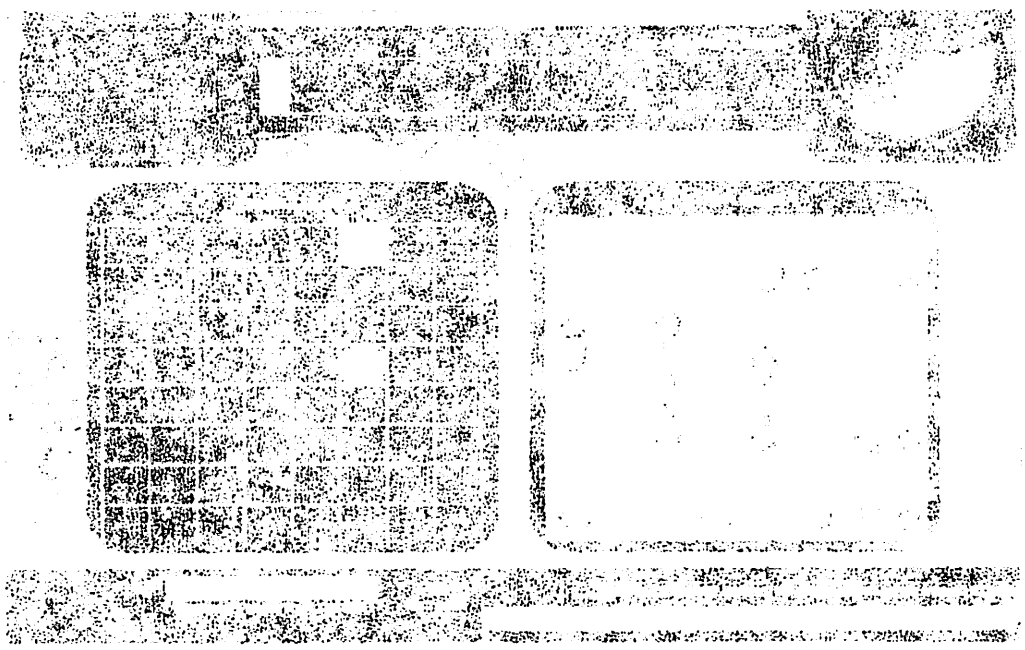
Gambar 4.9 Tampilan Tembakan Gagal Pada Komputer-2 (Pemain yang ditembak)

Apabila tembakan berhasil (mengenai kapal lawan), maka akan muncul animasi ledakan pada bagian yang tertembak.

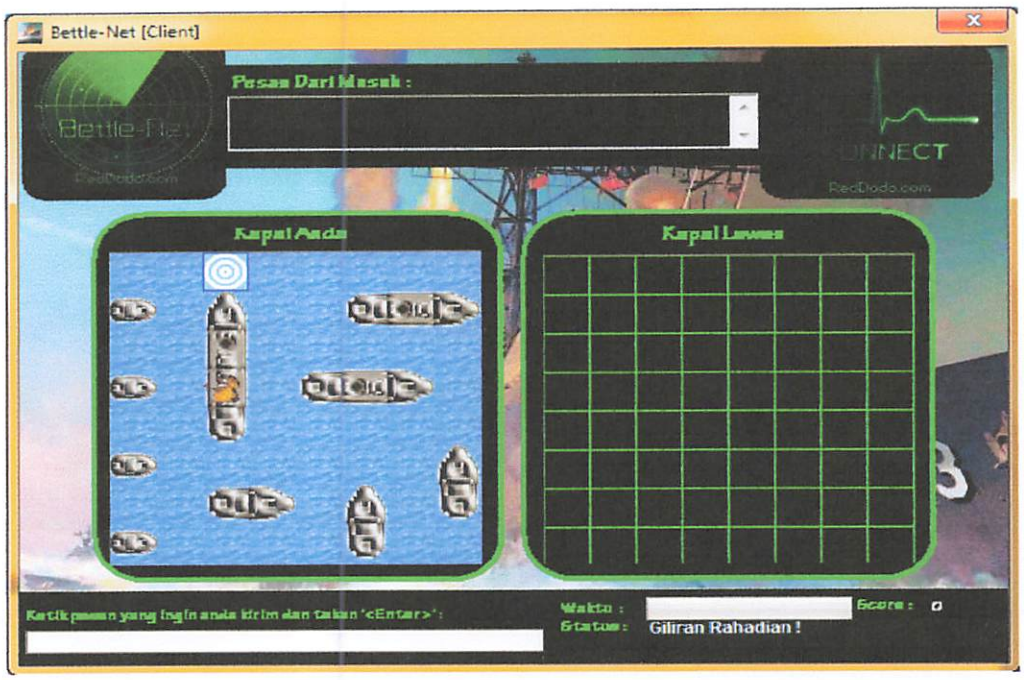


Gambar 4.10 Tampilan Tembakan Berhasil Pada Komputer-1 (Pemain yang menembak)

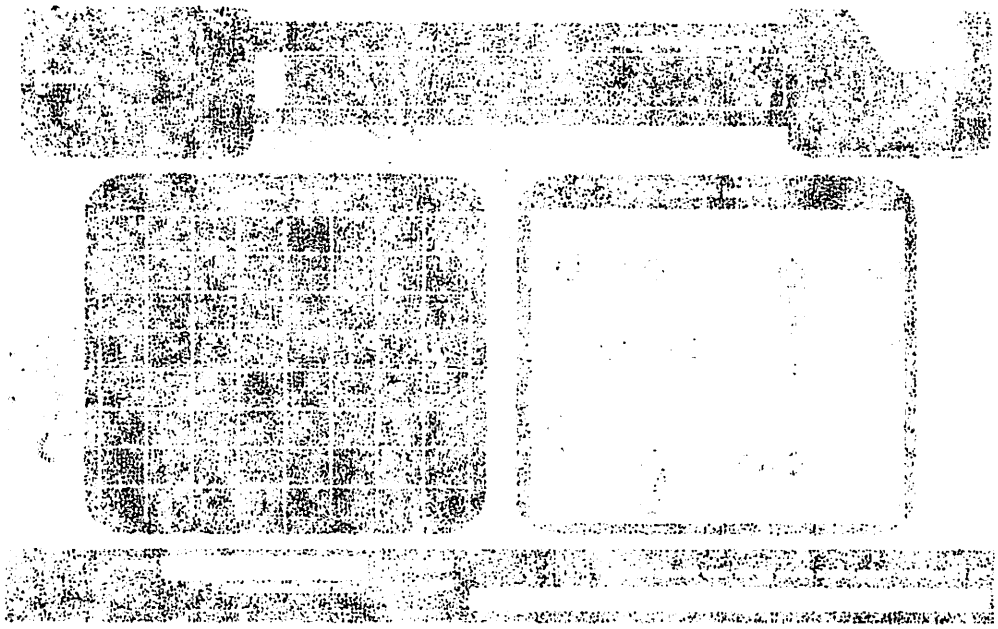




Apabila tembakan lawan berhasil, maka akan muncul animasi api pada bagian kapal yang tertembak pada daerah kotak sebelah kiri.



Gambar 4.11 Tampilan Tembakan Berhasil Pada Komputer-2 (Pemain yang ditembak)



BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Setelah menyelesaikan perancangan perangkat lunak permainan strategi *Battle Ship* yang dapat dimainkan pada jaringan, penulis menarik kesimpulan sebagai berikut :

1. Perangkat lunak permainan (*game*) *battle ship* ini memungkinkan pemakai (*user*) komputer untuk memainkan permainan (*game*) *battle ship* tanpa harus berhadapan secara langsung.
2. Perangkat lunak permainan (*game*) *battle ship* ini dapat dimainkan oleh dua orang yang terkoneksi dalam jaringan .
3. Pemrograman *Visual Basic* 6.0 mempunyai keterbatasan pada *WinSock* (*Windows Socket*) yang hanya dapat mengkoneksikan jaringan secara *Perr to Perr.*

5.2 Saran

Penulis memberikan beberapa saran sebagai berikut :

1. Perangkat lunak dapat dikembangkan untuk *user* yang lebih banyak (lebih dari 2 orang).
2. Perangkat lunak dapat ditambahkan konsep *Artificial Intelligence* (AI) sehingga dapat dimainkan dengan komputer.
3. Sebagai penyempurnaan perangkat lunak permainan (*game*) strategi *Battle Ship* untuk lebih dari 2 orang penulis menyarankan menggunakan pemrograman *Borland Delphi 7.*

DAFTAR PUSTAKA

[1] Suharian Ramadi, 2008. **4 *Game* Asah Otak dengan *Visual Basic 6*.** Palembang : MAXICOM.

[2] J. Von Neumann and O. Morgenstern, Theory of Games and Economic Behavior (3d ed. 1953).

[3]http://www.scribd.com/doc/22887860/Konsep-Dasar-Game-Animasi, 23 Juli 2011, 10.39 WIB.

[4]http://agesvisual.wordpress.com/2008/01/18/animasi-dalam-teori/, 23 Juli 2011, 10.39 WIB.

[5] http://www.scribd.com/doc/33214027/Animation, 23 Juli 2011, 10.39 WIB.

[6] http://www. ririez.blog.uns.ac.id, 29 Juli 2011, 13.14 WIT.

[7]http://jurnal.upnyk.ac.id/index.php/telematika/article/view/9/4 (winsock UDP),29 Juli 2011, 13.14 WIT.

[8]http://www.rimbaraya.net/ebook/view.php?file (winsock tcp), 29 Juli 2011, 13.14 WIT.

[9] Iwan Sofana, 2008. **Membuat Jaringan Komputer mudah membuat jaringan computer (*Wire & Wireless*) untuk pengguna *windows* dan *linux*.** Bandung : INFORMATIKA.

[10] Tri Kuntoro Priyambodo dan Dodi Heriadi, 2005. **Jaringan *Wi-Fi* Teori dan Implementasi**. Yogyakarta : CV ANDI OFFSET.

[11]http://directory.umm.ac.id/Data%20Elmu/doc/IP_ADDRESS_Versi_4.doc (ip address), 29 Juli 2011, 13.14 WIT.

[12] Harrington S, 1983. ***Computer Graphics A Programming*** *Approach*. Japan : International Student Edition Mc Graw-Hill International Book Company.

[13] Ario Suryokusumo, 2001. ***Microsoft Visual Basic 6.0.*** Jakarta : PT. Elex Media Komputindo.

[14] Djoko Pramono, 2002. **Mudah menguasai *Visual Basic 6*.** Jakarta : PT. Elex Media Komputindo,.

[15] Hadi, Rahadian, 2001. **Pemrograman *Microsoft Visual Basic***, Jakarta : PT. Elex Media Komputindo.

[16] WAHANA KOMPUTER, 2002. **Tips dan Trik Pemrograman *Visual Basic 6.0*.** Semarang : WAHANA KOMPUTER.

[17] www.jaycar.com.au/images_uploaded/networks.pdf. pada hari Sabtu tanggal 23 Juli 2011 pada jam 10.39 WIB.

L A M P I R A N

PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAM TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK
PT. BNI (PERSERO) MALANG
BANK NIAGA MALANG
Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

BERITA ACARA UJIAN SKRIPSI

FAKULTAS TEKNOLOGI INDUSTRI

Nama : Rahadian Mahendra
NIM : 06.12.654
Jurusan : Teknik Elektro S-1
Konsentrasi : Teknik Komputer dan Informatika
Judul Skripsi : PENGEMBANGAN PERANGKAT LUNAK PERMAINAN MULTIPLAYER STRATEGI BATTLE SHIP MENGGUNAKAN WIFI

Dipertahankan dihadapan Majelis Penguji Skripsi Jenjang Strata Satu (S-1) pada :
Hari : Sabtu
Tanggal : 18 Agustus 2011
Dengan Nilai : 76,8 (B+)

Panitia Ujian Skripsi,

Ketua Majelis Penguji

Ir. Yusuf Ismail Nakhoda, MT
NIP.Y.1018800189

Sekretaris Majelis Penguji

Dr. Eng. Aryuanto Soetedjo, ST. MT
NIP.Y.1030800417

Anggota Penguji,

Penguji Pertama

I Komang Somawirata, ST, MT
NIP.P.1030100361

Penguji Kedua

Ahmad Faisol, ST
NIP.P.1031000431

INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO S-1
Jl. Karanglo KM.2 Malan

FORMULIR PERBAIKAN SKRIPSI

Dalam pelaksanaan ujian skripsi jenjang Strata Satu (S-1) Jurusan Teknik Elektro Konsentrasi Teknik Komputer dan Informatika, maka perlu adanya perbaikan skripsi untuk mahasiswa :

NAMA : RAHADIAN MAHENDRA
NIM : 06.12.654
JURUSAN : Teknik Elektro S-1
KONSENTRASI : Teknik Komputer dan Informatika
MASA BIMBINGAN : 4 Juli 2011 s/d 4 Januari 2012
JUDUL : **PENGEMBANGAN PERANGKAT LUNAK PERMAINAN MULTIPLAYER STRATEGI BATTLE SHIP MENGGUNAKAN WIFI**

Tanggal	Uraian	Paraf
Penguji I 18 - 08 - 2011		
Penguji II 18 - 08 - 2011	Jumlah pemilihan kapal tidak perlu dibatasi yang penting jumlah kapal maksimal 10 buah	

Mengetahui,

Dosen Pembimbing I

Ibrahim Ashari, ST. MT
NIP.Y.1010100358

Dosen Pembimbing II

Sotyohadi, ST
NIP.Y.1039700309

Disetujui,

Dosen Penguji I

I Komang Somawirata, ST. MT
NIP.Y.1030100361

Dosen Penguji II

Ahmad Faisol, ST
NIP.P.1031000431

INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO

Formulir Perbaikan Ujian Skripsi

Dalam pelaksanaan Ujian Skripsi Janjang Strata 1 Jurusan Teknik Elektro Konsentrasi T. Energi Listrik / T. Elektronika / T.Infokom, maka perlu adanya perbaikan skripsi untuk mahasiswa :

NAMA : Rahadian Mahendra
NIM : 0612654
Perbaikan meliputi :

- Jumlah pemilihan kapal tdk perlu dibatasi yang penting jml. kapal max.10

Malang, 18-08-2011

(A. Pauzi)

INSTITUT TEKNOLOGI NASIONAL
Jl. Raya Karanglo, KM 2
MALANG

FORMULIR BIMBINGAN SKRIPSI

Nama : Rahadian Mahendra
Nim : 06.12.654
Masa Bimbingan : 7 Juni 2011 s/d 7 Desember 2011
Judul Skripsi : Pengembangan Perangkat Lunak Permainan Strategi Battle Ship Menggunakan Wifi

No	Tanggal	Uraian	Paraf Pembimbing
1	8 Juli	Bab I	
2	18 Juli	acc Bab I	
3	27 Juli	acc Bab II	
4	4 Agustus	acc Bab III	
5	5 Agustus	acc Bab IV	
6	6 Agustus	acc Bab V, Makalah	
7			
8			
9			
10			

Malang,
Dosen Pembimbing I

Ibrahim Ashari, ST, MT
NIP.Y.1010100358

Form S-4b

INSTITUT TEKNOLOGI NASIONAL
Jl. Raya Karanglo, KM 2
MALANG

FORMULIR BIMBINGAN SKRIPSI

Nama : Rahadian Mahendra
Nim : 06.12.654
Masa Bimbingan : 7 Juni 2011 s/d 7 Desember 2011
Judul Skripsi : Pengembangan Perangkat Lunak Permainan Strategi Battle Ship Menggunakan Wifi

No	Tanggal	Uraian	Paraf Pembimbing
1	26 Juli	Demo Program	
2	27 Juli	Bab I, Bab II	
3	29 Juli	Revisi Bab II	
4	4 Agustus	Revisi Program, Bab III, Bab IV, Bab V	
5	6 Agustus	Acc Bab I, Bab II, Bab IV, Bab V, Makalah	
6			
7			
8			
9			
10			

Malang,
Dosen Pembimbing II

Sotyohadi, ST
NIP.P.1039700309

Form S-4b

Institut Teknologi Nasional Malang
Fakultas Teknologi Industri
Jurusan Teknik Elektro S-1

BERITA ACARA SEMINAR PROPOSAL SKRIPSI
JURUSAN TEKNIK ELEKTRO S-1

Konsentrasi : Teknik Energi Listrik/Teknik Elektronika/ Teknik Komputer & Informatika*)

1.	Nama Mahasiswa: RAHADIAN MAHENDRA			Nim: 0612654
2.	Keterangan	Tanggal	Waktu	Tempat
	Pelaksanaan	4 Juli 2011		Ruang:
3.	Spesifikasi Judul (berilah tanda silang)**)			
	a. Sistem Tenaga Elektrik		e. Elektronika & Komponen	
	b. Energi & Konversi Energi		f. Elektronika Digital & Komputer	
	c. Tegangan Tinggi & Pengukuran		g. Elektronika Komunikasi	
	d. Sistem Kendali Industri		h. lainnya	
4.	Judul Proposal yang diseminarkan Mahasiswa	Pengembangan Perangkat Lunak Permainan Strategi Battle Ship Menggunakan WiFi		
5.	Perubahan Judul yang diusulkan oleh Kelompok Dosen Keahlian			
6.	Catatan:			
7.	Catatan:			
	Persetujuan Judul Skripsi			
	Disetujui, Dosen Keahlian I	Disetujui, Dosen Keahlian II	Disetujui, Dosen Keahlian III	
	Mengetahui, Ketua Jurusan. Ir. Yusuf Ismail Nakhoda,MT NIP. Y. 1018800189	Disetujui, Calon Dosen Pembimbing ybs		
		Pembimbing I	Pembimbing II	

Perhatian:
1. Keterangan: *) Coret yang tidak perlu
   **) dilingkari a, b, c, ...atau g sesuai bidang keahlian

Form S-3c

PERMOHONAN PERSETUJUAN SKRIPSI

Yang bertanda tangan dibawah ini :

Nama : Rahadian Mahendra
NIM : 0612654
Semester : IX
Fakultas : Teknologi Industri
Jurusan : Teknik Elektro S-1
Konsentrasi : ~~TEKNIK ELEKTRONIKA~~
~~TEKNIK ENERGI LISTRIK~~
TEKNIK KOMPUTER DAN INFORMATIKA
~~TEKNIK KOMPUTER~~
~~TEKNIK TELEKOMUNIKASI~~
Alamat : Jl. Tasikmadu 99 Malang

Dengan ini kami mengajukan permohonan untuk mendapatkan persetujuan untuk membuat ***SKRIPSI Tingkat Sarjana.*** Untuk melengkapi permohonan tersebut, bersama kami lampirkan persyaratan-persyaratan yang harus dipenuhi.
Adapun persyaratan-persyaratan pengambilan ***SKRIPSI*** adalah sebagai berikut :

1. Telah melaksanakan semua praktikum sesuai dengan konsentrasinya (...✓...)
2. Telah lulus dan menyerahkan Laporan Praktek Kerja (...✓...)
3. Telah lulus seluruh mata kuliah keahlian (MKB) sesuai konsentrasinya (...✓...)
4. Telah menempuh mata kuliah ≥ 134 sks dengan IPK ≥ 2 dan tidak ada nilai E (...✓...)
5. Telah mengikuti secara aktif kegiatan seminar skripsi yang diadakan Jurusan (.....)
6. Memenuhi persyaratan administrasi (.....)

Demikian permohonan ini untuk mendapatkan penyelesaian lebih lanjut dan atas perhatiannya kami ucapkan terima kasih.

Telah diteliti kebenaran data tersebut diatas
Recording Teknik Elektro

(Ayu Lantayani)

Malang, ...201
Pemohon

(Rahadian. M)

Disetujui
Ketua Jurusan Teknik Elektro

Ir. Yusuf Ismail Nakhoda, MT
NIP. Y. 1018800189

Mengetahui
Dosen Wali

(.....)

Catatan :
Bagi mahasiswa yang telah memenuhi persyaratan mengambil SKRIPSI agar membuat proposal dan mendapat persetujuan dari Ketua Jurusan/Sekretaris Jurusan T. Elektro S-1

1. IPK 4,25 / 3,07
2. (3,0)
3. Praktikum

Form. S-1a

LAMPIRAN

Penggambar kapal

```
Public headX As Integer    'head_x kapal

Public headY As Integer    'head_y kapal

Public length As Integer   'panjang kapal

Public direction As Integer 'Arah kapal (hor = vbKeyRight / vert = vbKeyDown)

Dim status() As Integer    'Kena tembak atau tidak

Sub drawShip()

'Gambar kapal

Select Case length

Case 1

  BitBlt frmMain.field1.hDC, headX * 30, headY * 30, frmMain.tile.ScaleWidth,
frmMain.tile.ScaleHeight, frmMain.shipSingleHorMask.hDC, 0, 0, vbSrcAnd

  BitBlt frmMain.field1.hDC, headX * 30, headY * 30, frmMain.tile.ScaleWidth,
frmMain.tile.ScaleHeight, frmMain.shipSingleHor.hDC, 0, 0, vbSrcPaint

Case 2

  If direction = vbKeyRight Then

    BitBlt frmMain.field1.hDC, headX * 30, headY * 30, frmMain.tile.ScaleWidth,
frmMain.tile.ScaleHeight, frmMain.shipLeftEndMask.hDC, 0, 0, vbSrcAnd

    BitBlt frmMain.field1.hDC, headX * 30, headY * 30, frmMain.tile.ScaleWidth,
frmMain.tile.ScaleHeight, frmMain.shipLeftEnd.hDC, 0, 0, vbSrcPaint

    BitBlt frmMain.field1.hDC, (headX + 1) * 30, headY * 30, frmMain.tile.ScaleWidth,
frmMain.tile.ScaleHeight, frmMain.shipRightEndMask.hDC, 0, 0, vbSrcAnd

    BitBlt frmMain.field1.hDC, (headX + 1) * 30, headY * 30, frmMain.tile.ScaleWidth,
frmMain.tile.ScaleHeight, frmMain.shipRightEnd.hDC, 0, 0, vbSrcPaint

  ElseIf direction = vbKeyDown Then

    BitBlt frmMain.field1.hDC, headX * 30, headY * 30, frmMain.tile.ScaleWidth,
frmMain.tile.ScaleHeight, frmMain.shipUpEndMask.hDC, 0, 0, vbSrcAnd

    BitBlt frmMain.field1.hDC, headX * 30, headY * 30, frmMain.tile.ScaleWidth,
frmMain.tile.ScaleHeight, frmMain.shipUpEnd.hDC, 0, 0, vbSrcPaint

    BitBlt frmMain.field1.hDC, headX * 30, (headY + 1) * 30, frmMain.tile.ScaleWidth,
frmMain.tile.ScaleHeight, frmMain.shipDownEndMask.hDC, 0, 0, vbSrcAnd
```

```
    BitBlt frmMain.field1.hDC, headX * 30, (headY + 1) * 30, frmMain.tile.ScaleWidth,
frmMain.tile.ScaleHeight, frmMain.shipDownEnd.hDC, 0, 0, vbSrcPaint

  End If

Case 3

  If direction = vbKeyRight Then

    BitBlt frmMain.field1.hDC, headX * 30, headY * 30, frmMain.tile.ScaleWidth,
frmMain.tile.ScaleHeight, frmMain.shipLeftEndMask.hDC, 0, 0, vbSrcAnd

    BitBlt frmMain.field1.hDC, headX * 30, headY * 30, frmMain.tile.ScaleWidth,
frmMain.tile.ScaleHeight, frmMain.shipLeftEnd.hDC, 0, 0, vbSrcPaint

    BitBlt frmMain.field1.hDC, (headX + 1) * 30, headY * 30, frmMain.tile.ScaleWidth,
frmMain.tile.ScaleHeight, frmMain.shipMid1Mask.hDC, 0, 0, vbSrcAnd

    BitBlt frmMain.field1.hDC, (headX + 1) * 30, headY * 30, frmMain.tile.ScaleWidth,
frmMain.tile.ScaleHeight, frmMain.shipMid1.hDC, 0, 0, vbSrcPaint

    BitBlt frmMain.field1.hDC, (headX + 2) * 30, headY * 30, frmMain.tile.ScaleWidth,
frmMain.tile.ScaleHeight, frmMain.shipRightEndMask.hDC, 0, 0, vbSrcAnd

    BitBlt frmMain.field1.hDC, (headX + 2) * 30, headY * 30, frmMain.tile.ScaleWidth,
frmMain.tile.ScaleHeight, frmMain.shipRightEnd.hDC, 0, 0, vbSrcPaint

  ElseIf direction = vbKeyDown Then

    BitBlt frmMain.field1.hDC, headX * 30, headY * 30, frmMain.tile.ScaleWidth,
frmMain.tile.ScaleHeight, frmMain.shipUpEndMask.hDC, 0, 0, vbSrcAnd

    BitBlt frmMain.field1.hDC, headX * 30, headY * 30, frmMain.tile.ScaleWidth,
frmMain.tile.ScaleHeight, frmMain.shipUpEnd.hDC, 0, 0, vbSrcPaint

    BitBlt frmMain.field1.hDC, headX * 30, (headY + 1) * 30, frmMain.tile.ScaleWidth,
frmMain.tile.ScaleHeight, frmMain.shipMid1DownMask.hDC, 0, 0, vbSrcAnd

    BitBlt frmMain.field1.hDC, headX * 30, (headY + 1) * 30, frmMain.tile.ScaleWidth,
frmMain.tile.ScaleHeight, frmMain.shipMid1Down.hDC, 0, 0, vbSrcPaint

    BitBlt frmMain.field1.hDC, headX * 30, (headY + 2) * 30, frmMain.tile.ScaleWidth,
frmMain.tile.ScaleHeight, frmMain.shipDownEndMask.hDC, 0, 0, vbSrcAnd

    BitBlt frmMain.field1.hDC, headX * 30, (headY + 2) * 30, frmMain.tile.ScaleWidth,
frmMain.tile.ScaleHeight, frmMain.shipDownEnd.hDC, 0, 0, vbSrcPaint

  End If

Case 4

  If direction = vbKeyRight Then

    BitBlt frmMain.field1.hDC, headX * 30, headY * 30, frmMain.tile.ScaleWidth,
frmMain.tile.ScaleHeight, frmMain.shipLeftEndMask.hDC, 0, 0, vbSrcAnd

    BitBlt frmMain.field1.hDC, headX * 30, headY * 30, frmMain.tile.ScaleWidth,
frmMain.tile.ScaleHeight, frmMain.shipLeftEnd.hDC, 0, 0, vbSrcPaint
```

```
    BitBlt frmMain.field1.hDC, (headX + 1) * 30, headY * 30, frmMain.tile.ScaleWidth,
frmMain.tile.ScaleHeight, frmMain.shipMid1Mask.hDC, 0, 0, vbSrcAnd

    BitBlt frmMain.field1.hDC, (headX + 1) * 30, headY * 30, frmMain.tile.ScaleWidth,
frmMain.tile.ScaleHeight, frmMain.shipMid1.hDC, 0, 0, vbSrcPaint

    BitBlt frmMain.field1.hDC, (headX + 2) * 30, headY * 30, frmMain.tile.ScaleWidth,
frmMain.tile.ScaleHeight, frmMain.shipMid2Mask.hDC, 0, 0, vbSrcAnd

    BitBlt frmMain.field1.hDC, (headX + 2) * 30, headY * 30, frmMain.tile.ScaleWidth,
frmMain.tile.ScaleHeight, frmMain.shipMid2.hDC, 0, 0, vbSrcPaint

    BitBlt frmMain.field1.hDC, (headX + 3) * 30, headY * 30, frmMain.tile.ScaleWidth,
frmMain.tile.ScaleHeight, frmMain.shipRightEndMask.hDC, 0, 0, vbSrcAnd

    BitBlt frmMain.field1.hDC, (headX + 3) * 30, headY * 30, frmMain.tile.ScaleWidth,
frmMain.tile.ScaleHeight, frmMain.shipRightEnd.hDC, 0, 0, vbSrcPaint

  ElseIf direction = vbKeyDown Then

    BitBlt frmMain.field1.hDC, headX * 30, headY * 30, frmMain.tile.ScaleWidth,
frmMain.tile.ScaleHeight, frmMain.shipUpEndMask.hDC, 0, 0, vbSrcAnd

    BitBlt frmMain.field1.hDC, headX * 30, headY * 30, frmMain.tile.ScaleWidth,
frmMain.tile.ScaleHeight, frmMain.shipUpEnd.hDC, 0, 0, vbSrcPaint

    BitBlt frmMain.field1.hDC, headX * 30, (headY + 1) * 30, frmMain.tile.ScaleWidth,
frmMain.tile.ScaleHeight, frmMain.shipMid2DownMask.hDC, 0, 0, vbSrcAnd

    BitBlt frmMain.field1.hDC, headX * 30, (headY + 1) * 30, frmMain.tile.ScaleWidth,
frmMain.tile.ScaleHeight, frmMain.shipMid2Down.hDC, 0, 0, vbSrcPaint

    BitBlt frmMain.field1.hDC, headX * 30, (headY + 2) * 30, frmMain.tile.ScaleWidth,
frmMain.tile.ScaleHeight, frmMain.shipMid1DownMask.hDC, 0, 0, vbSrcAnd

    BitBlt frmMain.field1.hDC, headX * 30, (headY + 2) * 30, frmMain.tile.ScaleWidth,
frmMain.tile.ScaleHeight, frmMain.shipMid1Down.hDC, 0, 0, vbSrcPaint

    BitBlt frmMain.field1.hDC, headX * 30, (headY + 3) * 30, frmMain.tile.ScaleWidth,
frmMain.tile.ScaleHeight, frmMain.shipDownEndMask.hDC, 0, 0, vbSrcAnd

    BitBlt frmMain.field1.hDC, headX * 30, (headY + 3) * 30, frmMain.tile.ScaleWidth,
frmMain.tile.ScaleHeight, frmMain.shipDownEnd.hDC, 0, 0, vbSrcPaint

  End If

End Select

frmMain.field1.Refresh

End Sub


Sub drawShip2()

'Gambar kapal komputer
```

```
Select Case length

Case 1

   BitBlt frmMain.field2.hDC, headX * 30, headY * 30, frmMain.tile.ScaleWidth,
frmMain.tile.ScaleHeight, frmMain.shipSingleHorMask.hDC, 0, 0, vbSrcAnd

   BitBlt frmMain.field2.hDC, headX * 30, headY * 30, frmMain.tile.ScaleWidth,
frmMain.tile.ScaleHeight, frmMain.shipSingleHor.hDC, 0, 0, vbSrcPaint

Case 2

   If direction = vbKeyRight Then

      BitBlt frmMain.field2.hDC, headX * 30, headY * 30, frmMain.tile.ScaleWidth,
frmMain.tile.ScaleHeight, frmMain.shipLeftEndMask.hDC, 0, 0, vbSrcAnd

      BitBlt frmMain.field2.hDC, headX * 30, headY * 30, frmMain.tile.ScaleWidth,
frmMain.tile.ScaleHeight, frmMain.shipLeftEnd.hDC, 0, 0, vbSrcPaint

      BitBlt frmMain.field2.hDC, (headX + 1) * 30, headY * 30, frmMain.tile.ScaleWidth,
frmMain.tile.ScaleHeight, frmMain.shipRightEndMask.hDC, 0, 0, vbSrcAnd

      BitBlt frmMain.field2.hDC, (headX + 1) * 30, headY * 30, frmMain.tile.ScaleWidth,
frmMain.tile.ScaleHeight, frmMain.shipRightEnd.hDC, 0, 0, vbSrcPaint

   ElseIf direction = vbKeyDown Then

      BitBlt frmMain.field2.hDC, headX * 30, headY * 30, frmMain.tile.ScaleWidth,
frmMain.tile.ScaleHeight, frmMain.shipUpEndMask.hDC, 0, 0, vbSrcAnd

      BitBlt frmMain.field2.hDC, headX * 30, headY * 30, frmMain.tile.ScaleWidth,
frmMain.tile.ScaleHeight, frmMain.shipUpEnd.hDC, 0, 0, vbSrcPaint

      BitBlt frmMain.field2.hDC, headX * 30, (headY + 1) * 30, frmMain.tile.ScaleWidth,
frmMain.tile.ScaleHeight, frmMain.shipDownEndMask.hDC, 0, 0, vbSrcAnd

      BitBlt frmMain.field2.hDC, headX * 30, (headY + 1) * 30, frmMain.tile.ScaleWidth,
frmMain.tile.ScaleHeight, frmMain.shipDownEnd.hDC, 0, 0, vbSrcPaint

   End If

Case 3

   If direction = vbKeyRight Then

      BitBlt frmMain.field2.hDC, headX * 30, headY * 30, frmMain.tile.ScaleWidth,
frmMain.tile.ScaleHeight, frmMain.shipLeftEndMask.hDC, 0, 0, vbSrcAnd

      BitBlt frmMain.field2.hDC, headX * 30, headY * 30, frmMain.tile.ScaleWidth,
frmMain.tile.ScaleHeight, frmMain.shipLeftEnd.hDC, 0, 0, vbSrcPaint

      BitBlt frmMain.field2.hDC, (headX + 1) * 30, headY * 30, frmMain.tile.ScaleWidth,
frmMain.tile.ScaleHeight, frmMain.shipMid1Mask.hDC, 0, 0, vbSrcAnd

      BitBlt frmMain.field2.hDC, (headX + 1) * 30, headY * 30, frmMain.tile.ScaleWidth,
frmMain.tile.ScaleHeight, frmMain.shipMid1.hDC, 0, 0, vbSrcPaint
```

```
    BitBlt frmMain.field2.hDC, (headX + 2) * 30, headY * 30, frmMain.tile.ScaleWidth,
frmMain.tile.ScaleHeight, frmMain.shipRightEndMask.hDC, 0, 0, vbSrcAnd

    BitBlt frmMain.field2.hDC, (headX + 2) * 30, headY * 30, frmMain.tile.ScaleWidth,
frmMain.tile.ScaleHeight, frmMain.shipRightEnd.hDC, 0, 0, vbSrcPaint

  ElseIf direction = vbKeyDown Then

    BitBlt frmMain.field2.hDC, headX * 30, headY * 30, frmMain.tile.ScaleWidth,
frmMain.tile.ScaleHeight, frmMain.shipUpEndMask.hDC, 0, 0, vbSrcAnd

    BitBlt frmMain.field2.hDC, headX * 30, headY * 30, frmMain.tile.ScaleWidth,
frmMain.tile.ScaleHeight, frmMain.shipUpEnd.hDC, 0, 0, vbSrcPaint

    BitBlt frmMain.field2.hDC, headX * 30, (headY + 1) * 30, frmMain.tile.ScaleWidth,
frmMain.tile.ScaleHeight, frmMain.shipMid1DownMask.hDC, 0, 0, vbSrcAnd

    BitBlt frmMain.field2.hDC, headX * 30, (headY + 1) * 30, frmMain.tile.ScaleWidth,
frmMain.tile.ScaleHeight, frmMain.shipMid1Down.hDC, 0, 0, vbSrcPaint

    BitBlt frmMain.field2.hDC, headX * 30, (headY + 2) * 30, frmMain.tile.ScaleWidth,
frmMain.tile.ScaleHeight, frmMain.shipDownEndMask.hDC, 0, 0, vbSrcAnd

    BitBlt frmMain.field2.hDC, headX * 30, (headY + 2) * 30, frmMain.tile.ScaleWidth,
frmMain.tile.ScaleHeight, frmMain.shipDownEnd.hDC, 0, 0, vbSrcPaint

  End If

Case 4

  If direction = vbKeyRight Then

    BitBlt frmMain.field2.hDC, headX * 30, headY * 30, frmMain.tile.ScaleWidth,
frmMain.tile.ScaleHeight, frmMain.shipLeftEndMask.hDC, 0, 0, vbSrcAnd

    BitBlt frmMain.field2.hDC, headX * 30, headY * 30, frmMain.tile.ScaleWidth,
frmMain.tile.ScaleHeight, frmMain.shipLeftEnd.hDC, 0, 0, vbSrcPaint

    BitBlt frmMain.field2.hDC, (headX + 1) * 30, headY * 30, frmMain.tile.ScaleWidth,
frmMain.tile.ScaleHeight, frmMain.shipMid1Mask.hDC, 0, 0, vbSrcAnd

    BitBlt frmMain.field2.hDC, (headX + 1) * 30, headY * 30, frmMain.tile.ScaleWidth,
frmMain.tile.ScaleHeight, frmMain.shipMid1.hDC, 0, 0, vbSrcPaint

    BitBlt frmMain.field2.hDC, (headX + 2) * 30, headY * 30, frmMain.tile.ScaleWidth,
frmMain.tile.ScaleHeight, frmMain.shipMid2Mask.hDC, 0, 0, vbSrcAnd

    BitBlt frmMain.field2.hDC, (headX + 2) * 30, headY * 30, frmMain.tile.ScaleWidth,
frmMain.tile.ScaleHeight, frmMain.shipMid2.hDC, 0, 0, vbSrcPaint

    BitBlt frmMain.field2.hDC, (headX + 3) * 30, headY * 30, frmMain.tile.ScaleWidth,
frmMain.tile.ScaleHeight, frmMain.shipRightEndMask.hDC, 0, 0, vbSrcAnd

    BitBlt frmMain.field2.hDC, (headX + 3) * 30, headY * 30, frmMain.tile.ScaleWidth,
frmMain.tile.ScaleHeight, frmMain.shipRightEnd.hDC, 0, 0, vbSrcPaint

  ElseIf direction = vbKeyDown Then
```

```
    BitBlt frmMain.field2.hDC, headX * 30, headY * 30, frmMain.tile.ScaleWidth,
frmMain.tile.ScaleHeight, frmMain.shipUpEndMask.hDC, 0, 0, vbSrcAnd

    BitBlt frmMain.field2.hDC, headX * 30, headY * 30, frmMain.tile.ScaleWidth,
frmMain.tile.ScaleHeight, frmMain.shipUpEnd.hDC, 0, 0, vbSrcPaint

    BitBlt frmMain.field2.hDC, headX * 30, (headY + 1) * 30, frmMain.tile.ScaleWidth,
frmMain.tile.ScaleHeight, frmMain.shipMid2DownMask.hDC, 0, 0, vbSrcAnd

    BitBlt frmMain.field2.hDC, headX * 30, (headY + 1) * 30, frmMain.tile.ScaleWidth,
frmMain.tile.ScaleHeight, frmMain.shipMid2Down.hDC, 0, 0, vbSrcPaint

    BitBlt frmMain.field2.hDC, headX * 30, (headY + 2) * 30, frmMain.tile.ScaleWidth,
frmMain.tile.ScaleHeight, frmMain.shipMid1DownMask.hDC, 0, 0, vbSrcAnd

    BitBlt frmMain.field2.hDC, headX * 30, (headY + 2) * 30, frmMain.tile.ScaleWidth,
frmMain.tile.ScaleHeight, frmMain.shipMid1Down.hDC, 0, 0, vbSrcPaint

    BitBlt frmMain.field2.hDC, headX * 30, (headY + 3) * 30, frmMain.tile.ScaleWidth,
frmMain.tile.ScaleHeight, frmMain.shipDownEndMask.hDC, 0, 0, vbSrcAnd

    BitBlt frmMain.field2.hDC, headX * 30, (headY + 3) * 30, frmMain.tile.ScaleWidth,
frmMain.tile.ScaleHeight, frmMain.shipDownEnd.hDC, 0, 0, vbSrcPaint

  End If

End Select

frmMain.field2.Refresh

End Sub

Function sectionNumber(ByVal x, ByVal y) As Integer

Dim tempX As Integer 'temp X coord

Dim tempY As Integer 'temp Y coord

tempX = headX     'temp_X

tempY = headY     'temp_Y

sectionNumber = 1

'Arah kapal

If direction = vbKeyRight Then

  Do Until tempX = x

    tempX = tempX + 1

    sectionNumber = sectionNumber + 1

  Loop

Else
```

```
  Do Until tempY = y
    tempY = tempY + 1
    sectionNumber = sectionNumber + 1
  Loop
End If
End Function
Sub setValues(ByVal x As Integer, ByVal y As Integer, ByVal s As Integer, ByVal dir As Integer)
Dim i As Integer   'counter
headX = x   'head_X kapal
headY = y   'head_Y kapal
length = s  'ukuran kapal
direction = dir 'arah kapal (vbKeyRight or vbKeyDown)
ReDim status(1 To s) 'Set status array untuk ukuran kapal
For i = 1 To s
  status(i) = 0
Next i
End Sub
Function cripple() As Boolean
'False jika kapal tertembak and
'true jika kapal tenggelam.
Dim i As Integer 'counter
'Loop through array and set the next available space
'to 1
For i = 1 To length
  If status(i) = 0 Then
    status(i) = 1
    cripple = isDead()
    Exit Function
  End If
Next i
```

```
cripple = isDead()

End Function

'Is Dead

Function isDead() As Boolean

Dim i As Integer 'counter

For i = 1 To length

  If status(i) = 0 Then

    isDead = False

    Exit Function

  End If

Next i

isDead = True

End Function
```

Deklarasi Fungsi

```
Option Explicit

'User log-in terakhir

Declare Function GetUserName Lib "advapi32.dll" Alias "GetUserNameA" (ByVal lpBuffer As String,
nSize As Long) As Long

'BitBlt untuk gambar

Public Declare Function BitBlt Lib "gdi32" (ByVal hDestDC As Long, ByVal x As _

    Long, ByVal y As Long, ByVal nWidth As Long, ByVal nHeight As Long, ByVal _

    hSrcDC As Long, ByVal xSrc As Long, ByVal ySrc As Long, ByVal dwRop As Long) As Long

Declare Function mciSendString Lib "winmm.dll" Alias "mciSendStringA" (ByVal lpstrCommand As
String, ByVal lpstrReturnString As String, ByVal uReturnLength As Long, ByVal hwndCallback As Long)
As Long

Public IsMusicOn As Boolean

Public RetValue As Long

'Untuk main musik

Declare Function sndPlaySound Lib "winmm.dll" Alias "sndPlaySoundA" (ByVal lpszSoundName As
String, ByVal uFlags As Long) As Long

Public Const SND_ASYNC = &H1
```

```
Public Const SND_LOOP = &H8
Public Const SND_NODEFAULT = &H2
Public Const SND_SYNC = &H0
Public Const SND_NOSTOP = &H10
Public Const SND_MEMORY = &H4
Public Const max = 10   'max field size
Public curFlame        'next available flame index
Public frameNum        'used with hitAnim and missAnim
Public isDead          'true = ship is dead
Public nextShotX As Integer 'computer tembakan berikutnya X coord
Public nextShotY As Integer 'computer tembakan berikutnya Y coord
Public Type flame
   typ As Integer
   frame As Integer
   x As Integer
   y As Integer
   backImage As PictureBox
   backImageMask As PictureBox
End Type
Sub Main()
   frmSplash.Show
  ' frmMain.Show 1
  ' End
End Sub
```

```
Private Sub Command1_Click()
End Sub
Private Sub Form_Activate()
   MMControl1.FileName = App.Path + "\back.mp3"
```

```
  MMControl1.Command = "Open"
  MMControl1.Command = "Play"
End Sub
Private Sub Form_KeyPress(KeyAscii As Integer)
  Unload Me
End Sub
Private Sub Frame1_Click()
  Unload Me
End Sub
Private Sub Form_QueryUnload(Cancel As Integer, UnloadMode As Integer)
MMControl1.Command = "Close"
End Sub
Private Sub Form_Unload(Cancel As Integer)
  Load frmMain
  frmMain.Show
End Sub
Private Sub Timer1_Timer()
  If Shape1.Left < Width Then
    Shape1.Left = Shape1.Left + 20
  Else
    Timer1.Enabled = False

    Unload Me
  End If
End Sub
```

Form Main Class

```
Option Explicit
Dim mess As String     'Pesan Status Game
Dim curX As Integer    'Last shot X coord
Dim curY As Integer    'Last shot Y coord
```

```
Dim user As String     'Nama Lawan
Dim user1 As Boolean    'Giliran (true = user1 turn)
Dim user2 As Boolean    'Giliran (true = user2 turn)
Dim zone1(0 To max - 1, 0 To max - 1)   'Info Lokasi Kapal Anda
Dim zone2(0 To max - 1, 0 To max - 1)   'Info Lokasi Tembakan Anda
Dim zone1flame(1 To 20) As flame    'Lokasi Nyala Api
Dim ships(1 To 10) As shipClass     'Kapal Anda
Dim durasi As Integer
Dim Waktu As Integer
Dim Score As Integer
'Apakah semua kapal sudah di-set pada tempatnya
Function allSet() As Boolean
Dim i As Integer    'counter
allSet = True
'Looping
For i = 1 To 10
  If ships(i) Is Nothing Then
    allSet = False
    Exit Function
  End If
Next i
End Function
'Prosedur giliran
Sub changeTurn()
user1 = Not user1
user2 = Not user2
field2.Enabled = Not field2.Enabled
Waktu = 0
PBar.Value = 0
'user1 = true?
```

```
If user1 Then
   'State caption
   lblState.Caption = " Giliran Anda !"
  'Hidupkan timer the turnTaggle untuk berkedip
   turnTaggle.Enabled = True
  'Hidupkan timer giliran
   tmrTurn.Enabled = True
Else
   'State caption
   lblState.Caption = "Giliran " & user & " !"
   'Matikan blinking
   turnTaggle.Enabled = False
   lblState.ForeColor = &HFFFFFF
   'Matikan timer giliran
   tmrTurn.Enabled = False
End If
End Sub
'Prosedur menggambar petak di lapangan
Sub drawField()
Dim i As Integer   'counter
Dim j As Integer   'counter
For i = 0 To max - 1
   For j = 0 To max - 1
      BitBlt field1.hDC, i * 30, j * 30, tile.ScaleWidth, tile.ScaleHeight, tile.hDC, 0, 0, vbSrcCopy
   Next j
Next i
field1.Refresh
field1.Visible = True
End Sub
'Prosedur menggambar semua kapal di lapangan
```

```
Sub drawShips()
Dim i As Integer    'counter
For i = 1 To 10
   ships(i).drawShip
Next
End Sub
'Prosedur gambar petak biru di lap1 (kiri)
'dan petak kosong di lap2 (kanan)
Sub drawWater()
Dim i As Integer    'counter
Dim j As Integer    'counter
For i = 0 To max - 1
   For j = 0 To max - 1
      BitBlt field1.hDC, i * 30, j * 30, tile.ScaleWidth, tile.ScaleHeight, water.hDC, 0, 0, vbSrcCopy
      BitBlt field2.hDC, i * 30, j * 30, tile.ScaleWidth, tile.ScaleHeight, tile.hDC, 0, 0, vbSrcCopy
   Next j
Next i
field1.Refresh
field2.Refresh
End Sub
'Prosedur ambil gambar dari pos x dan y
Sub getBackImage(ByVal x, ByVal y, ByRef backImage As PictureBox, ByRef backImageMask As
PictureBox)
Dim tempX As Integer    'temp X coord
Dim tempY As Integer    'temp Y coord
Dim headX As Integer    'head_X kapal
Dim headY As Integer    'head_Y kapal
Dim size As Integer     'ukuran kapal
Dim direc As Integer    'arah kapal
Dim i As Integer        'counter
```

```
Dim section As Integer  'section
tempX = x
tempY = y
'Cek koordinat apakah sedang menunjuk pada kapal
If zone1(tempX, tempY) <> 0 Then
  'Cari head_x dari kapal dengan cek mundur dari point yang diberikan
  Do
    tempX = tempX - 1
    If tempX < 0 Then
      Exit Do
    End If
  Loop Until zone1(tempX, y) = 0
  headX = tempX + 1
  'Cari head_y dari kapal dengan cek mundur dari point yang diberikan
  Do
    tempY = tempY - 1
    If tempY < 0 Then
      Exit Do
    End If
  Loop Until zone1(x, tempY) = 0
  headY = tempY + 1
End If
'Looping semua kapal untuk mencari apakah ada yang berada di x dan y
For i = 1 To 10
  If ships(i) Is Nothing Then
    'Do nothing
  ElseIf ships(i).headX = headX And ships(i).headY = headY Then
    direc = ships(i).direction
    size = ships(i).length
    section = ships(i).sectionNumber(x, y)
```

```
        Exit For
    End If
Next i
'Arah
If direc = vbKeyRight Then
    'ukuran -> back image
    If size = 1 Then
        Set backImage = shipSingleHor
        Set backImageMask = shipSingleHorMask
    ElseIf size = 2 Then
        If section = 1 Then
            Set backImage = shipLeftEnd
            Set backImageMask = shipLeftEndMask
        ElseIf section = 2 Then
            Set backImage = shipRightEnd
            Set backImageMask = shipRightEndMask
        End If
    ElseIf size = 3 Then
        If section = 1 Then
            Set backImage = shipLeftEnd
            Set backImageMask = shipLeftEndMask
        ElseIf section = 2 Then
            Set backImage = shipMid1
            Set backImageMask = shipMid1Mask
        ElseIf section = 3 Then
            Set backImage = shipRightEnd
            Set backImageMask = shipRightEndMask
        End If
    ElseIf size = 4 Then
        If section = 1 Then
```

```
        Set backImage = shipLeftEnd
        Set backImageMask = shipLeftEndMask
      ElseIf section = 2 Then
        Set backImage = shipMid1
        Set backImageMask = shipMid1Mask
      ElseIf section = 3 Then
        Set backImage = shipMid2
        Set backImageMask = shipMid2Mask
      ElseIf section = 4 Then
        Set backImage = shipRightEnd
        Set backImageMask = shipRightEndMask
      End If
    End If
  Else
    'ukuran
    If size = 1 Then
      Set backImage = shipSingleHor
      Set backImageMask = shipSingleHorMask
    ElseIf size = 2 Then
      If section = 1 Then
        Set backImage = shipUpEnd
        Set backImageMask = shipUpEndMask
      ElseIf section = 2 Then
        Set backImage = shipDownEnd
        Set backImageMask = shipDownEndMask
      End If
    ElseIf size = 3 Then
      If section = 1 Then
        Set backImage = shipUpEnd
        Set backImageMask = shipUpEndMask
```

```
      ElseIf section = 2 Then
        Set backImage = shipMid2Down
        Set backImageMask = shipMid2DownMask
      ElseIf section = 3 Then
        Set backImage = shipDownEnd
        Set backImageMask = shipDownEndMask
      End If
    ElseIf size = 4 Then
      If section = 1 Then
        Set backImage = shipUpEnd
        Set backImageMask = shipUpEndMask
      ElseIf section = 2 Then
        Set backImage = shipMid2Down
        Set backImageMask = shipMid2DownMask
      ElseIf section = 3 Then
        Set backImage = shipMid1Down
        Set backImageMask = shipMid1DownMask
      ElseIf section = 4 Then
        Set backImage = shipDownEnd
        Set backImageMask = shipDownEndMask
      End If
    End If
  End If
End Sub
'Apakah sudah game_over
Function isOver() As Boolean
Dim i As Integer
isOver = True
'Looping test semua apakah sudah tenggelam
For i = 1 To 10
```

```
    If Not ships(i).isDead Then
        isOver = False
        Exit Function
    End If
Next i
End Function
'Tombol keluar
Private Sub Command1_Click()
    Unload Me
End Sub
'Tombol start
Private Sub Command4_Click()
    Option2.Enabled = False
    Option3.Enabled = False
    Command4.Enabled = False
If Option2.Value Then
    '2 orang - Sebagai server
    Winsock.Close
    Winsock.LocalPort = 20202
    Winsock.Listen
     Caption = "Bettle-Net [Server]"
    mess = " Sedang menunggu computer client ..."
    lblStatus.Caption = mess & " ."
    lblStatus.Visible = True
    timer1.Enabled = True
ElseIf Option3.Value Then
    '2 orang - Sebagai klien
    Winsock.Close
    If txtServer.Text <> "" Then
        Winsock.Connect txtServer.Text, 20202
```

```
        Caption = "Bettle-Net [Client]"
        mess = " Menunggu balasan dari server ..."
        lblStatus.Caption = mess & " ."
        lblStatus.Visible = True
        timer1.Enabled = True
    Else
        MsgBox "Masukkan IP Server !", vbCritical
        Exit Sub
    End If
End If
End Sub
'Tombol Mulai Permainan
Private Sub Command5_Click()
'Lapangan u/ bermain
field2.Enabled = False
radarAnim.Enabled = True
txtReply.Visible = True
lblReply.Visible = True
Picture1.Visible = True
QProGIF1.Visible = True
Shape4.Visible = True
Frame1.Visible = False
frameField2.Visible = True
Shape3.Visible = True
field1.Enabled = False
drawWater
box.Visible = False
field1.Refresh
'Kirim info : mulai
Winsock.SendData "Info:Ready"
```

```
user1 = True
'Jika dua-dua sudah siap, mulai permainan
If user1 And user2 Then
  timer1.Enabled = False
  user2 = True
  user1 = False
  field2.Enabled = False
  changeTurn
Else
  'PESAN
  mess = " " & user & " masih belum siap, harap bersabar."
  lblState.Caption = mess & " ."
  timer1.Enabled = True
End If
drawShips
lblScore.Visible = True
Label9.Visible = True
Command5.Visible = False
'Waktu Tunggu
Label10.Visible = True
PBar.Visible = True
End Sub
Private Sub field1_MouseDown(Button As Integer, Shift As Integer, x As Single, y As Single)
'XXXXXX  On Error GoTo handler
Dim tempOption As OptionButton  'option select
Dim tempLbl As Label        'corresponding label
Dim curShip As Integer      'ship index sekarang
Dim tempX As Integer        'temp x
Dim tempY As Integer        'temp y
Dim headX As Integer        'head x
```

```
Dim headY As Integer          'head y
Dim direc As Integer          'arah kapal
Dim size As Integer           'ukuran ship
Dim i As Integer              'counter
Dim mark As Integer           'current mark (1-4 ship type)
Dim tempWidth As Integer      'temp width of box
Dim clickedX As Integer       'clicked x
Dim clickedY As Integer       'clicked y
Dim r                         'return value
'klik-kiri
If Button = 1 Then
  'Cari tempat yang masih bisa ditempati kapal di ship array
  For curShip = 1 To 9
    If ships(curShip) Is Nothing Then
      Exit For
    End If
  Next
  'Tandai kapal apa yang anda tempatkan
  If Option7.Value Then
    mark = 1
    Set tempOption = Option7
    Set tempLbl = lbl7
  ElseIf Option6.Value Then
    mark = 2
    Set tempOption = Option6
    Set tempLbl = lbl6
  ElseIf Option5.Value Then
    mark = 3
    Set tempOption = Option5
    Set tempLbl = lbl5
```

```
ElseIf Option4.Value Then
    mark = 4
    Set tempOption = Option4
    Set tempLbl = lbl4
Else
    'hapus tanda
    mark = 0
End If
'Hapus tanda
If mark = 0 Then
    clickedX = Int((box.Left + 10) / 30)
    clickedY = Int(y / 30)
    tempX = clickedX
    tempY = clickedY
    'Cek apakah kapal
    If zone1(tempX, tempY) <> 0 Then
        'Cari head_x dari kapal
        Do
            tempX = tempX - 1
            If tempX < 0 Then
                Exit Do
            End If
        Loop Until zone1(tempX, clickedY) = 0
        headX = tempX + 1
        'Cari head_y dari kapal
        Do
            tempY = tempY - 1
            If tempY < 0 Then
                Exit Do
            End If
```

```
        Loop Until zone1(clickedX, tempY) = 0
        headY = tempY + 1
    'Cari kapal dengan head_x dan head_y
    For i = 1 To 10
        If ships(i) Is Nothing Then
            'Nothing
        ElseIf ships(i).headX = headX And ships(i).headY = headY Then
            direc = ships(i).direction
            size = ships(i).length
            Set ships(i) = Nothing 'hapus kapal
            Exit For
        End If
    Next i
    'Gambar petak dimana kapal berada
    If direc = vbKeyRight Then
        For i = 0 To size - 1
            BitBlt field1.hDC, (headX + i) * 30, headY * 30, tile.ScaleWidth, tile.ScaleHeight, tile.hDC, 0, 0,
vbSrcCopy
            zone1((headX + i), headY) = 0
        Next i
    Else
        For i = 0 To size - 1
            BitBlt field1.hDC, headX * 30, (headY + i) * 30, tile.ScaleWidth, tile.ScaleHeight, tile.hDC, 0, 0,
vbSrcCopy
            zone1(headX, headY + i) = 0
        Next i
    End If
    'Tulis Jumlah Kapal
    If size = 4 Then
        Option4.Enabled = True
        lbl4.Caption = Val(lbl4.Caption) + 1 & " buah"
```

```
    ElseIf size = 3 Then
      Option5.Enabled = True
      lbl5.Caption = Val(lbl5.Caption) + 1 & " buah"
    ElseIf size = 2 Then
      Option6.Enabled = True
      lbl6.Caption = Val(lbl6.Caption) + 1 & " buah"
    ElseIf size = 1 Then
      Option7.Enabled = True
      lbl7.Caption = Val(lbl7.Caption) + 1 & " buah"
    End If
    field1.Refresh
    End If
  Else   'Set kapal
    'Jika horizontal
    If box.Width >= box.Height Then
      'Valid?
      If checkValidity(Int((box.Left + 10) / 30), Int(y / 30), vbKeyRight, mark) Then
        'Sound
        r = sndPlaySound(App.Path & "\taruh.wav", SND_ASYNC)
        'Tandai petak di sekeliling kapal
        For i = Int((box.Left + 10) / 30) To Int((box.Left + 10) / 30) + mark - 1
          zone1(i, Int(y / 30)) = mark
        Next i
      'Set kapal baru
        Set ships(curShip) = New shipClass
        ships(curShip).setValues Int((box.Left + 10) / 30), Int(y / 30), mark, vbKeyRight
      'Gambar kapal
        ships(curShip).drawShip
      Else
        MsgBox "Posisi Tidak Diperbolehkan !", vbCritical
```

```
        Exit Sub
      End If
    Else 'Vertikal
      If checkValidity(Int(x / 30), Int((box.Top + 10) / 30), vbKeyDown, mark) Then
        'Sound
        r = sndPlaySound(App.Path & "\taruh.wav", SND_ASYNC)
        'Tandai petak di sekeliling kapal
        For i = Int((box.Top + 10) / 30) To Int((box.Top + 10) / 30) + mark - 1
          zone1(Int(x / 30), i) = mark
        Next i

        'Set kapal baru
        Set ships(curShip) = New shipClass
        ships(curShip).setValues Int(x / 30), Int((box.Top + 10) / 30), mark, vbKeyDown
        'Gambar kapal
        ships(curShip).drawShip
      Else
        MsgBox "Posisi Tidak Diperbolehkan !", vbCritical
        Exit Sub
      End If
    End If
    'Enable/Disable dan set Focus di kapal berikutnya yang enable
    tempLbl.Caption = Val(Left(tempLbl.Caption, 1)) - 1 & " buah"
    If Left(tempLbl.Caption, 1) = "0" Then
      tempOption.Enabled = False
      If Option4.Enabled Then
        Option4.SetFocus
      ElseIf Option5.Enabled Then
        Option5.SetFocus
      ElseIf Option6.Enabled Then
```

```
            Option6.SetFocus
          ElseIf Option7.Enabled Then
            Option7.SetFocus
          Else
            Option8.SetFocus
          End If
        End If
      End If
    Else 'klik kanan
      'Ubah arah kotak
      tempWidth = box.Width
      box.Width = box.Height
      box.Height = tempWidth
    End If
    'Cek apakah sudah semua posisi kapal di-set
    If allSet() Then
      Command5.Enabled = True
    Else
      Command5.Enabled = False
    End If
    Exit Sub
    'Error handler
    handler:
      MsgBox ("Out of bounds error")
    End Sub
    'Cek validitas letak kapal
    Function checkValidity(ByVal x As Integer, ByVal y As Integer, ByVal dir As Integer, ByVal mark As
    Integer) As Boolean
    Dim xStart As Integer   'kapal start_X coord
    Dim yStart As Integer   'kapal start_Y coord
```

```
Dim xEnd As Integer    'kapal end_X coord
Dim yEnd As Integer    'kapal end_Y coord
Dim i As Integer       'counter 1
Dim k As Integer       'counter 2
'Arah menentukan bagaimana letak bagian kepala dan akhir kapal
If dir = vbKeyRight Then
  If x > 0 Then
    xStart = x - 1
  Else
    xStart = 0
  End If
  If x + mark < max Then
    xEnd = x + mark
  Else
    xEnd = max - 1
  End If
  If y > 0 Then
    yStart = y - 1
  Else
    yStart = 0
  End If
  If y < max - 1 Then
    yEnd = y + 1
  Else
    yEnd = max - 1
  End If
  For i = xStart To xEnd
    For k = yStart To yEnd
      If zone1(i, k) <> 0 Then
        checkValidity = False
```

```
            Exit Function
        End If
      Next k
    Next i
  Else
    'Arah lain
    If x > 0 Then
      xStart = x - 1
    Else
      xStart = 0
    End If
    If x < max - 1 Then
      xEnd = x + 1
    Else
      xEnd = max - 1
    End If
    If y > 0 Then
      yStart = y - 1
    Else
      yStart = 0
    End If
    If y + mark < max Then
      yEnd = y + mark
    Else
      yEnd = max - 1
    End If
    For i = xStart To xEnd
      For k = yStart To yEnd
        If zone1(i, k) <> 0 Then
          checkValidity = False
```

```
            Exit Function
        End If
      Next k
    Next i
  End If
  checkValidity = True
  End Function
  'Pergerakan box dengan mouse
  Private Sub field1_MouseMove(Button As Integer, Shift As Integer, x As Single, y As Single)
  'Set box visible
  If Not box.Visible Then
    box.Visible = True
  End If
  'Cursor di tengah kotak
  box.Left = x - box.Width / 2
  box.Top = y - box.Height / 2
  'Kotak tetap di dalam box
  If box.Left < 0 Then
    box.Left = 0
  End If
  If box.Top < 0 Then
    box.Top = 0
  End If
  If box.Left + box.Width > 300 Then
    box.Left = 300 - box.Width
  End If
  If box.Top + box.Height > 300 Then
    box.Top = 300 - box.Height
  End If
  End Sub
```

```
'Mouse_click untuk Lapangan / Daerah Lawan
Private Sub field2_MouseDown(Button As Integer, Shift As Integer, x As Single, y As Single)
Dim r As Integer   'return value
Dim i As Integer
Dim tempX As Integer
Dim tempY As Integer
Dim headX As Integer
Dim headY As Integer
  curX = Int(x / 30)
  curY = Int(y / 30)
 'Jika petak kosong
  If zone2(curX, curY) = 0 Then
    changeTurn
    zone2(curX, curY) = 1
   'Kirim info shot
    Winsock.SendData "Shot:" & curX & "," & curY
  End If
End Sub
Private Sub field2_MouseMove(Button As Integer, Shift As Integer, x As Single, y As Single)
  field2.MousePointer = 99
End Sub
'Animasi Api pada Kapal
Private Sub fireAnim_Timer()
Dim i As Integer   'counter
i = 1
Do Until zone1flame(i).typ = 0
  'Mask api
  BitBlt field1.hDC, zone1flame(i).x * 30, zone1flame(i).y * 30, 30, 30, fireMask.hDC,
zone1flame(i).frame * 30, 0, vbSrcAnd
  'Gambar api
```

```
  BitBlt field1.hDC, zone1flame(i).x * 30, zone1flame(i).y * 30, 30, 30, fire.hDC, zone1flame(i).frame *
30, 0, vbSrcPaint

 'Inc. frame

  zone1flame(i).frame = zone1flame(i).frame + 1

 'Reset frame

  If zone1flame(i).frame = 19 Then

     zone1flame(i).frame = 0

  End If

 'Increment counter

  i = i + 1

  'Tidak bisa lebih besar dari 20

  If i > 20 Then

     fireAnim.Enabled = False

     Exit Sub

  End If

Loop

field1.Refresh

'Reset counter

i = 1

Do Until zone1flame(i).typ = 0

  BitBlt field1.hDC, zone1flame(i).x * 30, zone1flame(i).y * 30, tile.ScaleWidth, tile.ScaleHeight,
zone1flame(i).backImageMask.hDC, 0, 0, vbSrcAnd

  BitBlt field1.hDC, zone1flame(i).x * 30, zone1flame(i).y * 30, tile.ScaleWidth, tile.ScaleHeight,
zone1flame(i).backImage.hDC, 0, 0, vbSrcPaint

  i = i + 1

Loop

End Sub

Private Sub Form_Activate()

  MMControl1.FileName = App.Path + "\back.mp3"

  MMControl1.Command = "Open"

  MMControl1.Command = "Play"
```

```
  durasi = 0
End Sub
Private Sub Form_Load()
'Set default user status
user1 = False
user2 = False
'Set curFlame
curFlame = 1
'Set radar frame number
'Center frameType and status label
frameType.Left = Me.Width / 2 - frameType.Width / 2
frameType.Top = Me.Height / 2 - frameType.Height / 2 - 1000
'lblStatus.Left = frameType.Left - 70
'lblStatus.Top = frameType.Top + frameType.Height + 300
'Server default
Option2.Value = True
'Initialize compShot
nextShotX = -1
durasi = 0
End Sub
Private Sub Form_QueryUnload(Cancel As Integer, UnloadMode As Integer)
  MMControl1.Command = "Close"
  Cancel = 1
  If MsgBox("Anda Yakin Untuk Menutup Program?", vbQuestion + vbYesNo, "Konfirmasi") = vbYes
Then
    Winsock.Close
    Cancel = 0
  Else
    Load frmSplash
    frmSplash.Show
```

```
    Cancel = 0
  End If
End Sub
'Animasi ledakan - kena tembak
Private Sub hitAnim_Timer()

Dim r As Integer   'return value
'Gambar frame ledakan
BitBlt field2.hDC, curX * 30, curY * 30, 30, 30, hitMask.hDC, frameNum * 32, 0, vbSrcAnd
BitBlt field2.hDC, curX * 30, curY * 30, 30, 30, hit.hDC, frameNum * 32, 0, vbSrcPaint
'Inc. frame
frameNum = frameNum + 1
field2.Refresh
'Stop animasi
If frameNum = 19 Then
  frameNum = 0
  hitAnim.Enabled = False
  BitBlt field2.hDC, curX * 30, curY * 30, tile.ScaleWidth, tile.ScaleHeight, shipDead.hDC, 0, 0,
vbSrcCopy
  field2.Refresh
  If isDead Then
    isDead = False
  End If
End If
End Sub
Private Sub missAnim_Timer()
'Animasi miss -> tidak kena
'Gambar frame
BitBlt field2.hDC, curX * 30, curY * 30, 30, 30, waterAnim.hDC, frameNum * 30, 0, vbSrcCopy
'Inc Frame
```

```
frameNum = frameNum + 1

field2.Refresh

'Stop animasi

If frameNum = 4 Then

   frameNum = 0

   missAnim.Enabled = False

   BitBlt field2.hDC, curX * 30, curY * 30, tile.ScaleWidth, tile.ScaleHeight, waterMiss.hDC, 0, 0,
vbSrcCopy

   field2.Refresh

End If

End Sub

Private Sub MMControl1_StatusUpdate()

If MMControl1.Mode = 525 Then

   MMControl1.From = 0

   MMControl1.Command = "Play"

End If

End Sub

'Multi player server

Private Sub Option2_Click()

   frameGame.Visible = False

End Sub

'Multi player client

Private Sub Option3_Click()

   frameGame.Visible = True

   txtServer.Text = Winsock.LocalIP

End Sub

'Kapal Battleship

Private Sub Option4_Click()

box.BorderColor = &HF0C4A6

'Gambar kapal
```

```
picCurrent.Cls
BitBlt picCurrent.hDC, 0, 0, tile.ScaleWidth, tile.ScaleHeight, shipLeftEndMask.hDC, 0, 0, vbSrcAnd
BitBlt picCurrent.hDC, 0, 0, tile.ScaleWidth, tile.ScaleHeight, shipLeftEnd.hDC, 0, 0, vbSrcPaint
BitBlt picCurrent.hDC, 30, 0, tile.ScaleWidth, tile.ScaleHeight, shipMid1Mask.hDC, 0, 0, vbSrcAnd
BitBlt picCurrent.hDC, 30, 0, tile.ScaleWidth, tile.ScaleHeight, shipMid1.hDC, 0, 0, vbSrcPaint
BitBlt picCurrent.hDC, 60, 0, tile.ScaleWidth, tile.ScaleHeight, shipMid2Mask.hDC, 0, 0, vbSrcAnd
BitBlt picCurrent.hDC, 60, 0, tile.ScaleWidth, tile.ScaleHeight, shipMid2.hDC, 0, 0, vbSrcPaint
BitBlt picCurrent.hDC, 90, 0, tile.ScaleWidth, tile.ScaleHeight, shipRightEndMask.hDC, 0, 0, vbSrcAnd
BitBlt picCurrent.hDC, 90, 0, tile.ScaleWidth, tile.ScaleHeight, shipRightEnd.hDC, 0, 0, vbSrcPaint
picCurrent.Refresh
'Ubah ukuran kotak
box.Width = 30 * 4
box.Height = 30
End Sub
'Kapal Cruiser
Private Sub Option5_Click()
box.BorderColor = &HF0C4A6
'Gambar kapal
picCurrent.Cls
BitBlt picCurrent.hDC, 0, 0, tile.ScaleWidth, tile.ScaleHeight, shipLeftEndMask.hDC, 0, 0, vbSrcAnd
BitBlt picCurrent.hDC, 0, 0, tile.ScaleWidth, tile.ScaleHeight, shipLeftEnd.hDC, 0, 0, vbSrcPaint
BitBlt picCurrent.hDC, 30, 0, tile.ScaleWidth, tile.ScaleHeight, shipMid1Mask.hDC, 0, 0, vbSrcAnd
BitBlt picCurrent.hDC, 30, 0, tile.ScaleWidth, tile.ScaleHeight, shipMid1.hDC, 0, 0, vbSrcPaint
BitBlt picCurrent.hDC, 60, 0, tile.ScaleWidth, tile.ScaleHeight, shipRightEndMask.hDC, 0, 0, vbSrcAnd
BitBlt picCurrent.hDC, 60, 0, tile.ScaleWidth, tile.ScaleHeight, shipRightEnd.hDC, 0, 0, vbSrcPaint
picCurrent.Refresh
'Ubah ukuran kotak
box.Width = 30 * 3
box.Height = 30
End Sub
```

```
'Kapal sub-marine
Private Sub Option6_Click()
box.BorderColor = &HF0C4A6

'Gambar kapal
picCurrent.Cls
BitBlt picCurrent.hDC, 0, 0, tile.ScaleWidth, tile.ScaleHeight, shipLeftEndMask.hDC, 0, 0, vbSrcAnd
BitBlt picCurrent.hDC, 0, 0, tile.ScaleWidth, tile.ScaleHeight, shipLeftEnd.hDC, 0, 0, vbSrcPaint
BitBlt picCurrent.hDC, 30, 0, tile.ScaleWidth, tile.ScaleHeight, shipRightEndMask.hDC, 0, 0, vbSrcAnd
BitBlt picCurrent.hDC, 30, 0, tile.ScaleWidth, tile.ScaleHeight, shipRightEnd.hDC, 0, 0, vbSrcPaint
picCurrent.Refresh
'Ubah ukuran kotak
box.Width = 30 * 2
box.Height = 30
End Sub
'Kapal destroyer
Private Sub Option7_Click()
box.BorderColor = &HF0C4A6
'Gambar kapal
picCurrent.Cls
BitBlt picCurrent.hDC, 0, 0, tile.ScaleWidth, tile.ScaleHeight, shipSingleHorMask.hDC, 0, 0, vbSrcAnd
BitBlt picCurrent.hDC, 0, 0, tile.ScaleWidth, tile.ScaleHeight, shipSingleHor.hDC, 0, 0, vbSrcPaint
picCurrent.Refresh
'Ubah ukuran kotak
box.Width = 30 * 1
box.Height = 30
End Sub
'Hapus Kapal
Private Sub Option8_Click()
box.BorderColor = vbRed
```

```
'Gambar kapal
picCurrent.Cls
BitBlt picCurrent.hDC, 0, 0, tile.ScaleWidth, tile.ScaleHeight, tile.hDC, 0, 0, vbSrcCopy
'Ubah ukuran kotak
box.Width = 30 * 1
box.Height = 30
End Sub
'Animasi radar
Private Sub radarAnim_Timer()
End Sub
'Animasi pesan
Private Sub Timer1_Timer()
AeroProgressBar1.Value = durasi
If (durasi < AeroProgressBar1.max) Then
   durasi = durasi + 1
If lblStatus.Caption = mess & " ." Then
   lblStatus.Caption = mess & " .."
ElseIf lblStatus.Caption = mess & " .." Then
   lblStatus.Caption = mess & " ..."
ElseIf lblStatus.Caption = mess & " ..." Then
   lblStatus.Caption = mess & " ."
End If
If lblState.Caption = mess & " ." Then
   lblState.Caption = mess & " .."
ElseIf lblState.Caption = mess & " .." Then
   lblState.Caption = mess & " ..."
ElseIf lblState.Caption = mess & " ..." Then
   lblState.Caption = mess & " ."
End If
Else
```

```
    durasi = 0
    timer1.Enabled = False
    'Winsock.Close
    lblStatus.Caption = "Close Connectios.."
    Option2.Enabled = True
    Option3.Enabled = True
    Command4.Enabled = True
End If
End Sub
Private Sub tmrTurn_Timer()
    Waktu = Waktu + 1
    PBar.Value = Waktu
    If Waktu >= 20 Then
        Winsock.SendData "Status:TimeOver"
        DoEvents
        Call changeTurn
        MsgBox "Anda Kehabisan Waktu !", vbCritical
    End If
End Sub
'Timer untuk blink
Private Sub turnTaggle_Timer()
If lblState.ForeColor = &HFF00& Then
    lblState.ForeColor = &HFFFFFF
Else
    lblState.ForeColor = &HFF00&
End If
End Sub
Private Sub txtReceive_Change()
txtReceive.SelStart = Len(txtReceive.Text)
End Sub
```

```
Private Sub txtReply_KeyPress(KeyAscii As Integer)

'XXXXXX  On Error GoTo handler

'Kirim

If KeyAscii = vbKeyReturn And txtReply.Text <> "" Then

   Winsock.SendData "Message:" & txtReply.Text

   'Tampilkan

   If txtReceive.Text <> "" Then

      txtReceive.Text = txtReceive.Text & vbNewLine & "Anda -> " & txtReply.Text

   Else

      txtReceive.Text = txtReceive.Text & "Anda -> " & txtReply.Text

   End If

   txtReply.Text = ""

End If

Exit Sub

handler:

   txtReceive.Text = txtReply.Text

End Sub

Private Sub Winsock_Close()

  If MsgBox(user & " telah keluar !" & Chr(13) & Chr(10) & "Apakah Anda Mau Main Lagi?",
vbQuestion + vbYesNo, "Konfirmasi") = vbYes Then

      Load frmSplash

      frmMain.Hide

      Unload frmMain

      frmSplash.Show

   Else

       End

   End If

End Sub

Private Sub Winsock_Connect()

'On connection
```

```
'hide status label
timer1.Enabled = False
lblStatus.Visible = False
'Kirim - connect
Winsock.SendData "Status:Connected"
frameType.Visible = False
frameField1.Visible = True
Shape1.Visible = True
drawField   'gambar daerah kosong
'Set game field
Frame1.Visible = True
lblState.Visible = True
Label8.Visible = True
Command5.Visible = True
lblStatus.Visible = False
End Sub
Private Sub Winsock_ConnectionRequest(ByVal requestID As Long)
'Terima koneksi
Winsock.Close
Winsock.Accept requestID
End Sub
'Terima dATA
Private Sub Winsock_DataArrival(ByVal bytesTotal As Long)
Dim info As String
Dim r
Winsock.GetData info
If info = "Status:Connected" Then
   timer1.Enabled = False
   Frame1.Visible = True
```

```
    lblState.Visible = True
    Label8.Visible = True
    Command5.Visible = True
    lblStatus.Visible = False
    lblStatus.Visible = False
    Winsock.SendData "Status:Gotit"
    Do
        r = InputBox("Masukkan Nama Anda :", "Nama", GetUser)
    Loop While r = ""
    'Kirim nama ke lawan
    Winsock.SendData "User:" & r
   drawField
    frameType.Visible = False
ElseIf info = "Status:Gotit" Then
   Do
        r = InputBox("Masukkan Nama Anda :", "Nama", GetUser)
    Loop While r = ""
    'Kirim nama ke lawan
    Winsock.SendData "User:" & r
    field2.Enabled = True
    drawField
    frameType.Visible = False
ElseIf info = "Status:TimeOver" Then
    'Ubah giliran
    Call changeTurn
Else
    'Parse pesan
    parse (info)
End If
End Sub
```

```
Function GetUser() As String
Dim lpUserID As String
Dim nBuffer As Long
Dim ret As Long
lpUserID = String(25, 0)
nBuffer = 25
ret = GetUserName(lpUserID, nBuffer)
If ret Then
   GetUser$ = lpUserID$
End If
End Function
'Parse msg
Sub parse(info As String)
'XXXXXX  On Error GoTo handler
Dim msgtype As String   'tipe message
Dim msg As String      'simpan msg
Dim cX As Integer      'X coord
Dim cY As Integer      'Y coord
Dim tempX As Integer    'temp X coord
Dim tempY As Integer    'temp Y coord
Dim headX As Integer    'head X coord
Dim headY As Integer    'head Y coord
Dim r                  'return value
Dim i As Integer       'counter
Dim sendStr As String   'Pesan Kirim
Dim tempship As shipClass   'untuk gambar kapal berikutnya
Dim tempstr As String      'temp

Dim tempdir As Integer
Dim tempsize As Integer
```

```
Dim tempheadx As Integer
Dim tempheady As Integer
Dim cNilai As String
Dim P_Score As Integer
msgtype = Left(info, InStr(1, info, ":") - 1)
P_Score = InStr(1, info, ";")
If P_Score > 0 Then
   msg = Mid(info, InStr(1, info, ":") + 1, P_Score - (InStr(1, info, ":") + 1))
   cNilai = Mid(info, InStr(1, info, ";") + 1)
   lblScore = cNilai
Else
   msg = Mid(info, InStr(1, info, ":") + 1)
End If
'Proses pesan
If msgtype = "User" Then
   MsgBox (msg & " telah bergabung !")
   user = msg
   frameType.Visible = False
   frameField1.Visible = True
   Shape1.Visible = True
   drawField
ElseIf msgtype = "Shot" Then
   'Tembak di posisi x, y
   cX = Left(msg, InStr(1, msg, ",") - 1)
   cY = Mid(msg, InStr(1, msg, ",") + 1)
  'Koordinat_Temp
   tempX = cX
   tempY = cY
   'zone1(x,y) < 5 - kena tembak
   If zone1(cX, cY) > 0 And zone1(cX, cY) < 5 Then
```

```
zone1(cX, cY) = zone1(cX, cY) + 4
'Cari head_x kapal
Do
    tempX = tempX - 1
    If tempX < 0 Then
        Exit Do
    End If
Loop Until zone1(tempX, cY) = 0
headX = tempX + 1
'Cari head_y kapal
Do
    tempY = tempY - 1
    If tempY < 0 Then
        Exit Do
    End If
Loop Until zone1(cX, tempY) = 0
headY = tempY + 1
'Cari kapal dengan head_x dan head_y
For i = 1 To 10
    If ships(i).headX = headX And ships(i).headY = headY Then
        'Apakah kapal sudah tenggelam
        r = ships(i).cripple()
        'r = true - kapal tenggelam
        If r Then
            Dim nA As Integer
            nA = ships(i).length * 10
            Score = Score + nA
            'lblScore = Score
            'Game_over?
            If isOver() Then
```

```
                'Menang
                Winsock.SendData "Result:Win"
            Else
                'Kapal Tenggelam
                Winsock.SendData "Result:Dead;" & Score
            End If
        Else
            'Kapal Tertembak
            Winsock.SendData "Result:Wound"
        End If
        zone1flame(curFlame).frame = 0
        Randomize
        zone1flame(curFlame).typ = Int(Rnd() * 3) + 1
        zone1flame(curFlame).x = cX
        zone1flame(curFlame).y = cY
        getBackImage cX, cY, zone1flame(curFlame).backImage,
zone1flame(curFlame).backImageMask
        curFlame = curFlame + 1
        Exit For
      End If
    Next i
  Else
    'Gambar 'air' di daerah tidak kena tembak
    BitBlt field1.hDC, cX * 30, cY * 30, tile.ScaleWidth, tile.ScaleHeight, waterMiss.hDC, 0, 0,
vbSrcCopy
    'Kirim pesan
    Winsock.SendData "Result:Miss"
  End If
  changeTurn
  field1.Refresh
'Hasi pertandingan
```

```
ElseIf msgtype = "Result" Then
  If msg = "Dead" Then
    'animasi ledakan
    hitAnim.Enabled = True
    isDead = True
   'sound
    r = sndPlaySound(App.Path & "\kena.wav", SND_ASYNC)
    MsgBox "Anda telah menenggelamkan kapal " & user & " !", vbExclamation
  ElseIf msg = "Win" Then
    'animasi ledakan
    hitAnim.Enabled = True
    isDead = True
   'kapal masih belum tenggelam
    sendStr = "Remain:"
    For i = 1 To 10
      If Not ships(i).isDead Then
        If sendStr <> "Remain:" Then
          sendStr = sendStr & ":" & ships(i).headX & "," & ships(i).headY & "," & ships(i).direction &
"," & ships(i).length
        Else
          sendStr = sendStr & ships(i).headX & "," & ships(i).headY & "," & ships(i).direction & "," &
ships(i).length
        End If
      End If
    Next i
    Winsock.SendData sendStr

    'sound
    r = sndPlaySound(App.Path & "\kena.wav", SND_ASYNC)
    MsgBox "Anda telah menenggelamkan kapal " & user & " !", vbExclamation
    MsgBox "Selamat, Anda telah mengalahkan " & user & " !", vbInformation, "Menang"
```

```
  ElseIf msg = "Wound" Then
    'animasi ledakan
    hitAnim.Enabled = True
    'Sound
    r = sndPlaySound(App.Path & "\kena.wav", SND_ASYNC)
  ElseIf msg = "Miss" Then
    'animate tidak kena
    missAnim.Enabled = True
    'sound
    r = sndPlaySound(App.Path & "\miss.wav", SND_ASYNC)
  End If
'Info = status message
ElseIf msgtype = "Info" Then
  'Lawan sudah siap
  If msg = "Ready" Then
    user2 = True    'set true
    'jika dua-dua sudah siap
    If user1 And user2 Then
      timer1.Enabled = False
      user1 = True
      user2 = False
      field2.Enabled = True
      changeTurn
    Else
      'tunggu
      If user1 Then
        mess = " " & user & " masih belum siap, harap bersabar !"
      Else
        mess = " " & user & " telah siap untuk bermain !"
      End If
```

```
        lblState.Caption = mess & " ."
        timer1.Enabled = True
      End If
    End If
'Pesan - chatting
ElseIf msgtype = "Message" Then
  If txtReceive.Text <> "" Then
      txtReceive.Text = txtReceive.Text & vbNewLine & user & " -> " & msg
    Else
      txtReceive.Text = txtReceive.Text & user & " -> " & msg
    End If
  txtReceive.SelStart = Len(txtReceive.Text)
ElseIf msgtype = "Remain" Then
  Set tempship = New shipClass
  Do
    On Error GoTo handler
    tempstr = Left(msg, InStr(1, msg, ":") - 1)
    msg = Mid(msg, InStr(1, msg, ":") + 1)
    tempheadx = Left(tempstr, InStr(1, tempstr, ",") - 1)
    tempstr = Mid(tempstr, InStr(1, tempstr, ",") + 1)
    tempheady = Left(tempstr, InStr(1, tempstr, ",") - 1)
    tempstr = Mid(tempstr, InStr(1, tempstr, ",") + 1)
    tempdir = Left(tempstr, InStr(1, tempstr, ",") - 1)
    tempstr = Mid(tempstr, InStr(1, tempstr, ",") + 1)
    tempsize = tempstr
    tempship.setValues tempheadx, tempheady, tempsize, tempdir
    tempship.drawShip2
  Loop While msg <> ""
  'MsgBox "Anda telah dikalahkan oleh " & user & "!", vbCritical, "Kalah"
```

```
  If MsgBox("Anda telah dikalahkan oleh " & user & "!" & Chr(13) & Chr(10) & "Apakah Anda Mau
Main Lagi?", vbQuestion + vbYesNo, "Konfirmasi") = vbYes Then

      Load frmSplash

      frmSplash.Show

      Unload frmMain

   Else

      Unload frmMain

      End

   End If

End If

Exit Sub

handler:

    If MsgBox("Anda telah dikalahkan oleh " & user & "!" & Chr(13) & Chr(10) & "Apakah Anda Mau
Main Lagi?", vbQuestion + vbYesNo, "Konfirmasi") = vbYes Then

      Load frmSplash

      frmSplash.Show

      Unload frmMain

   Else

      Unload frmMain

      End

   End If

End Sub
```