

BAB I PENDAHULUAN

1.1. LATAR BELAKANG

Aplikasi *game* komputer dibuat untuk mengisi waktu luang, hiburan atau hobi. Ada berbagai macam jenis *game* komputer, diantaranya terdapat *board game*, *card game*, dan *turn-based strategy*. Masing-masing jenis *game* ini, sejak awal mula diciptakannya *game* komputer, sudah banyak dibuat dengan berbagai macam variasi. Memainkan jenis *game* yang sama secara terus-menerus akan membuat pemain merasa cepat jenuh dan beralih ke *game* lainnya dalam waktu singkat. Diperlukan suatu inovasi baru setiap saat untuk membuat konsep *game* yang benar-benar berbeda, agar *game* tersebut dapat membuat para pemainnya ingin memainkan *game* tersebut untuk waktu yang lama, dan membuat *game* tersebut terkenal di kalangan *gamer*.

Berawal dari pemikiran tersebut, diciptakan *game* ini, yang merupakan gabungan konsep ketiga jenis *game* tersebut di atas. Adanya konsep baru ini diharapkan dapat menarik perhatian para *gamer* dan menjadi *game* yang cukup adiktif untuk dimainkan.

Aplikasi *game* akan dibuat dengan bahasa pemrograman *Visual C# .NET 2005*. Teknik komunikasi antar komputer yang akan dipakai adalah *.NET Remoting* dengan arsitektur *network pure peer-to-peer*, yang memiliki kelebihan dibandingkan arsitektur *client-server*, yaitu apabila *server* terputus, maka *game* masih dapat dilanjutkan. Selain itu, setelah mencari sumber di *internet*, penulis tidak menemukan referensi arsitektur *pure peer-to-peer* menggunakan *.NET Remoting*, sehingga penulis tertarik untuk merancanginya. Tampilan permainan akan memanfaatkan teknik *GDI+* yang mutlak diperlukan dalam pembuatan *game*. Rincian data *game* akan memanfaatkan pembacaan *file* teks, melihat kedinamisannya dibandingkan dengan *database*.

1.2. RUMUSAN MASALAH

Permasalahan dalam aplikasi *game* ini adalah:

1. Bagaimana cara membuat koneksi *pure peer-to-peer* menggunakan teknologi *.NET remoting* ?
2. Bagaimana cara membuat tampilan dengan memanfaatkan *OwnerDraw* dan *GDI+* ?

3. Bagaimana cara membaca data dari *file* teks ?

1.3. TUJUAN

Tujuan dari pembuatan aplikasi ini adalah:

1. Membuat koneksi *pure peer-to-peer* menggunakan teknologi *.NET Remoting*
2. Membuat tampilan *game* dengan memanfaatkan *OwnerDraw* dan *GDI+*
3. Memakai metode pembacaan data dari *file* teks

1.4. BATASAN MASALAH

1. Bahasa pemrograman yang digunakan adalah *Microsoft Visual C#.NET 2005*
2. Sistem operasi yang didukung adalah *Microsoft Windows Vista* atau *Microsoft Windows XP Professional Service Pack 2*, yang telah dilengkapi dengan *.NET Framework 2.0*.
3. Teknik pemrograman *network* yang digunakan adalah *.NET Remoting*.
4. Arsitektur *network* pada aplikasi ini adalah *pure peer-to-peer*.
5. *Game* ini berjudul *BloodRush*.
6. *Game* ini dapat dimainkan oleh 1 sampai 4 orang.
7. Resolusi monitor minimal adalah 1024 x 768 *pixel*.
8. Tampilan pemain bersifat 2 dimensi.
9. Spesifikasi minimum perangkat keras untuk menjalankan *game* ini:
 - a. Processor : setingkat *Intel Pentium 3 1.5 GHz*
 - b. Memori : 512 MB
 - c. Sisa ruang *hard disk* : 150 MB
 - d. *Local Area Network* tersedia
10. Terdapat 19 *avatar* yang bisa dipilih pemain
11. Terdapat 1 peta permainan
12. Tidak menggunakan intelegensi buatan
13. Tidak terdapat fitur menyimpan permainan
14. *Testing* dilakukan dengan *freeware* perangkat lunak NUnit versi 2.4.3
15. Menggunakan gambar dari game *Warcraft III* dengan seijin *Blizzard Entertainment*

16. Memakai kelas *AutoDiscovery* untuk mendeteksi adanya *peer* yang terkoneksi dengan komputer. Kelas ini dibuat dan dipublikasikan oleh *John R. Lewis* di websitenya <http://aspzone.com> dan dapat diperoleh atau dipakai secara umum/gratis. Kelas ini tidak dimodifikasi dan tidak dibahas lebih lanjut.
17. Pemakaian kursor beranimasi di dalam aplikasi memanfaatkan *library* yang telah disediakan oleh *Windows*, yaitu *user32.dll*. Pemakaian kelas ini tidak dibahas lebih lanjut.
18. *Testing* pada metode *SortCards()* langsung terhenti apabila nilai *null* terdeteksi, sehingga nilai kartu terakhir hasil *test* belum berubah. Hal ini bertentangan dengan program yang memanfaatkan nilai *null*. Hal ini tidak berpengaruh pada praktek permainan, sehingga dibiarkan dan tidak dibahas lebih lanjut.

1.5. SISTEMATIKA PEMBAHASAN

Organisasi penulisan laporan pembuatan aplikasi ini per bab adalah sebagai berikut:

1. Pendahuluan

Bab ini berisi latar belakang, rumusan masalah, tujuan, batasan masalah, sistematika pembahasan, serta *time schedule*. Bab ini harus menyediakan gambaran mengenai keseluruhan dokumen.

2. Dasar Teori

Bab ini menjabarkan teori keseluruhan dari aplikasi, seperti aturan permainan, teori penunjang, metode algoritmik yang dipakai dan protokol komunikasi.

3. Analisa dan Pemodelan

Bab ini berisi latar belakang cerita, rincian *game*, arsitektur *game*, *storyboard*, dan *creative strategy*. Di dalam bab ini UML yang akan digunakan adalah :

- a. *Use Case*
- b. *Activity Diagram*
- c. *Class Diagram*
- d. *Sequence Diagram*

4. Perancangan

Bab ini menjabarkan tiap metode atau fungsi utama yang terdapat di dalam aplikasi melalui kata-kata dan *flowchart*. Cuplikan *source code* ditampilkan pada bab IV.

5. Pengujian

Bab ini menjabarkan hasil pengujian secara *whitebox* terhadap tiap metode atau fungsi utama yang ada dan memiliki nilai *output* yang patut diuji. *Whitebox testing* menggunakan aplikasi *NUnit* versi 2.4.3.

6. Kesimpulan dan Saran

Bab ini berisi kesimpulan dan saran. Kesimpulan berisi penegasan/pembuktian dan pengetahuan baru yang didapat setelah pembuatan aplikasi ini. Saran berisi hal baru yang dapat digunakan untuk mengembangkan karya ilmiah ini.

1.6. TIME SCHEDULE

Tabel I.1 *Time Schedule*

	Jan	Feb	Mar	Apr	Mei	Jun	Jul	Agt	Sep	Okt	Nov	Des
Perencanaan	■											
Pencarian bahan		■										
Penyuntingan gambar			■									
Desain		■										
Implementasi			■	■	■	■	■	■	■	■	■	■
Testing			■	■	■	■	■	■	■	■	■	■
Laporan			■	■	■	■	■	■	■	■	■	■