

# Tugas Besar 1 – IF 2210 Pemrograman Berorientasi Objek (STI)

## Final Fun-tasy

### Pengenalan dan Inti Permainan

Final Fun-tasy (FF) adalah sebuah RPG (Role Playing Game) yang dimodifikasi dari permainan “Final Fantasy V”. ([http://en.wikipedia.org/wiki/Final\\_Fantasy](http://en.wikipedia.org/wiki/Final_Fantasy))

Permainan ini berlatar di Tycoon Castle dan dimulai ketika seorang pengembara bernama Bartz menginvestigasi jatuhnya meteor di dekat Tycoon Castle. Dalam perjalanannya, ia menemukan bahwa jatuhnya Tycoon Meteorite berkaitan dengan bahaya hilangnya salah satu Crystal yang mengendalikan elemen Bumi. Crystal ini ternyata diincar oleh Exdeath, sebuah penyihir jahat, untuk mengumpulkan kekuatan. Bartz dan teman-temannya harus menjaga Crystal ini agar tidak dikuasai oleh Exdeath.

Permainan ini dimainkan dengan tiga orang karakter yang bertugas menjaga Crystal tersebut. Karakter tersebut antara lain bernama Bartz, Lenna, dan Faris. Setiap karakter memiliki kemampuan yang berbeda-beda (akan dijelaskan kemudian). Setiap karakter memiliki level, dengan batas level maksimal adalah level 40. Crystal dapat dikatakan bebas dari ancaman ketika ketiga karakter telah mencapai level maksimal.

Dalam permainan ini, pemain akan berjalan-jalan di arena Tycoon Castle. Dalam perjalanan ini, mereka mungkin bertemu dengan monster buatan Exdeath. Jika bertemu dengan monster tersebut, mereka harus melakukan battle. Pemain dapat melawan monster dengan melakukan attack, magic (menggunakan jurus sihir tertentu, tergantung pekerjaan karakter masing-masing), atau item (menggunakan item yang dimiliki). Hal ini dilakukan secara *turn-based* antara pemain dan monster (dimainkan komputer). Pemain akan menang battle jika monster yang ada berhasil dibunuh. Dengan memenangi battle, pemain mendapatkan kenaikan level. Sebaliknya, pemain akan kalah battle jika HP dari seluruh pemain habis dan monster belum mati.

Dalam perjalanan di Tycoon Castle pula, pemain dapat melakukan transaksi untuk membeli item-item yang dibutuhkan. Item ini dapat dibeli dengan uang yang dimiliki pemain. Selain itu, pemain juga dapat mengambil item-item yang ditemukan di arena menjadi miliknya.

### Deskripsi Persoalan

Berikut akan dijelaskan mengenai spesifikasi minimal permainan Final Fun-tasy yang harus dibuat. Selama spesifikasi minimal tercapai, **silahkan berkreasi** agar permainan yang dibuat menjadi lebih menarik. Anda dibebaskan dalam desain kelas, tampilan, dan sebagainya kecuali beberapa hal yang memang diberikan petunjuk tertentu.

## Batasan Program

Pada permainan Final Fantasy yang akan dibangun, komponen minimal yang perlu dibuat adalah sebagai berikut.

### Pemain

Pemain adalah kelas kontrol utama bagi yang memainkan permainan dengan spesifikasi minimal antara lain:

1. Money : jumlah uang yang dimiliki, bisa untuk membeli item
2. Item : daftar item yang dimiliki pemain, dapat digunakan dalam battle
3. Posisi : posisi pemain dalam arena Tycoon Castle
4. Characters : daftar karakter yang dimainkan (tiga karakter)

### Karakter serta Kelas Turunannya


Status minimal yang perlu dimiliki sebuah karakter adalah:



1. Nama : nama karakter, yaitu dengan default Bartz/Lenna/Faris
2. Status :
  - a. STR : status untuk menentukan besar attack damage pemain yaitu  $ATK=2*STR$
  - b. VIT : status untuk menentukan besar HP maksimum ( $10*VIT$ )
  - c. INT : status untuk menentukan besar MP maksimum ( $10*INT$ )
3. Level
4. HP (Health Point) : prinsipnya adalah “darah” pemain. Pemain akan dinyatakan mati jika nilai HP pemain adalah 0
5. MP (Magic Point) : kekuatan “sihir” pemain. Pemain tidak dapat melakukan langkah magic jika MP habis

Petunjuk pembuatan kelas:

Ketiga karakter memiliki jurus sihir tertentu tergantung pekerjaan dari karakter tersebut. Nilai STR, VIT, dan INT juga tergantung pekerjaan masing-masing. Untuk membedakan kemampuan pekerjaan karakter tersebut, buatlah dalam bentuk kelas pendetilan dari kelas *Karakter* tersebut secara **inheritance**.

Contoh karakter pada permainan:

Karakter	Keterangan	Peningkatan nilai status
 Bartz Klauser (Kelas: Hero)	Status : <ul style="list-style-type: none"><li>• STR : 6</li><li>• VIT : 10</li><li>• INT : 4</li></ul>	Level 1-10 : Status 100% Level 11-20 : Status 120% Level 21-30 : Status 150% Level 31-40 : Status 200%

 Lenna Charlotte Tycoon (Kelas: Mage)	Status : <ul style="list-style-type: none"> <li>● STR : 2</li> <li>● VIT : 6</li> <li>● INT : 12</li> </ul>	Level 1-10 : Status 100% Level 11-20 : Status 110% Level 21-30 : Status 120% Level 31-40 : Status 130%
 Faris Scherwiz (Kelas: Pirate)	Status : <ul style="list-style-type: none"> <li>● STR : 8</li> <li>● VIT : 8</li> <li>● INT : 4</li> </ul>	Level 1-10 : Status 100% Level 11-20 : Status 120% Level 21-30 : Status 140% Level 31-40 : Status 160%

\*Catatan: Jenis-jenis karakter diatas hanyalah contoh, silahkan berkreasi dengan nama kelas dan status yang berbeda.

### *Monster dan Kelas Turunannya*


Status minimal yang perlu dimiliki oleh monster adalah




1. Nama : nama monster
2. ATK : kekuatan serangan monster dalam mengurangi HP pemain
3. HP : prinsipnya adalah “darah” dari monster tersebut. Monster akan mati apabila nilai HP monster mencapai 0.
4. Posisi : posisi monster di arena Tycoon Castle
5. Money : Nilai uang yang ditinggalkan apabila monster musnah
6. IncLevel : peningkatan level yang diperoleh jika monster dikalahkan

Petunjuk pembuatan kelas:

Setiap monster memiliki Nama, ATK, HP, Money, dan IncLevel berbeda-beda sesuai jenisnya. Untuk membedakan jenis tersebut, buatlah dalam bentuk kelas pendetilan dari kelas *Monser* secara **inheritance**.

Contoh jenis monster pada permainan

Jenis	Keterangan
 Alchymia	<ul style="list-style-type: none"> <li>● ATK : 5</li> <li>● HP : 200</li> <li>● Money : 100</li> <li>● IncLevel : 1</li> </ul>




 Blue Dragon	<ul style="list-style-type: none"> <li>• ATK : 12</li> <li>• HP : 300</li> <li>• Money : 200</li> <li>• IncLevel : 2</li> </ul>
 Farfarello	<ul style="list-style-type: none"> <li>• ATK : 3</li> <li>• HP : 100</li> <li>• Money : 70</li> <li>• IncLevel : 0</li> </ul>
 Hydra	<ul style="list-style-type: none"> <li>• ATK : 10</li> <li>• HP : 150</li> <li>• Money : 120</li> <li>• IncLevel : 1</li> </ul>

\*Catatan: Jenis-jenis monster diatas hanyalah contoh, silahkan berkreasi dengan nama kelas dan status yang berbeda.

#### Item

Sebuah item akan memiliki kegunaan serta harga yang berbeda-beda. Jumlah jenis item minimal adalah 3 jenis.

Contoh jenis item pada permainan

Jenis	Keterangan
 Potion	<ul style="list-style-type: none"> <li>• Kegunaan: menambahkan HP saat ini dari salah satu karakter dalam pertarungan</li> <li>• Harga: 50 gold</li> </ul>
 Ether	<ul style="list-style-type: none"> <li>• Kegunaan: menambahkan MP saat ini dari salah satu karakter dalam pertarungan</li> <li>• Harga: 150 gold</li> </ul>
 Bomb	<ul style="list-style-type: none"> <li>• Kegunaan: mengurangi HP monster sebesar 200 dalam pertarungan</li> <li>• Harga: 500 gold</li> </ul>

\*Catatan: Jenis-jenis item diatas hanyalah contoh, silahkan berkreasi dengan nama, harga, serta kegunaan yang berbeda..

#### Arena Permainan

Terdapa dua jenis arena permainan yang wajib ditampilkan:

Arena pertama adalah **arena Tycoon Castle**, berupa *map* di mana pemain akan berjalan-jalan. Luas arena ini adalah 10x10 grid. Dalam arena ini terdapat :

- 5 petak yang berupa toko : jika pemain berada dalam petak tersebut, pemain dapat melakukan transaksi untuk membeli item yang ada
- 10 petak berisi item : jika pemain berada dalam petak tersebut, pemain dapat mengambil item yang berada dalam petak tersebut
- 20 petak berupa lokasi monster : jika pemain berada dalam petak tersebut, pemain akan bertemu monster dan melakukan battle (pindah ke arena Battle)

Arena kedua adalah arena **Battle**. Dalam arena ini, pemain akan melakukan battle dengan monster. Luas arena battle bebas, ditentukan oleh masing-masing kelompok.

### Program Utama

Ketika pengguna baru pertama kali membuka aplikasi, maka menu yang ditampilkan adalah:

- **New Game** ( Untuk memulai Game )
- **Load Game** ( Untuk membuka file Game yang sudah ada )
- **Exit** ( Untuk keluar dari aplikasi )

Ketika pengguna memilih menu *New Game*, maka diberikan tampilan untuk memberikan nama bagi seluruh karakter, dengan nama pada spesifikasi menjadi nama *default* karakter.

Ketika pengguna memilih menu *Load Game*, maka ditampilkan daftar simpanan permainan yang tersimpan pada komputer. Pengguna dapat memilih simpanan yang akan dimainkan dan data-data simpanan tersebut akan dipanggil.

Setelah memulai permainan pengguna akan dapat memberikan perintah kepada karakter, antara lain:

- **Move** ( Untuk memindahkan karakter )
- **Shop** ( Untuk membeli item ). Bila opsi ini dipilih, maka ditampilkan daftar barang yang dapat dibeli serta harganya. Bila pengguna telah memilih barang yang akan dibeli, pengguna dapat menentukan berapa jumlah barang yang ingin dibeli.
- **Save** ( Untuk menyimpan data permainan yang sedang digunakan ). Pengguna dapat memilih apakah akan melakukan *overwrite* pada data permainan yang sudah ada ataupun menyimpan pada slot baru bila masih tersedia.

Interaksi ketika melakukan pertarungan telah dijelaskan pada spesifikasi.

### Interaksi dengan Pengguna

Interaksi antara pengguna dan aplikasi dilakukan dengan memberikan perintah melalui keyboard pada konsol. Masukan yang diberikan dapat berupa:

- **Angka**. Beberapa pilihan aksi yang dapat dilakukan ditampilkan di layar dengan nomor-nomor tertentu dan pengguna memasukkan nomor aksi yang ingin dilakukan.

Contoh:

Tampilan	<ol style="list-style-type: none"> <li>1. Move</li> <li>2. Shop</li> <li>3. Save</li> <li>4. Exit</li> </ol>
Masukan	> 4

- **String.** Pengguna mengetikkan nama aksi yang akan dilakukan dan aplikasi akan mendeteksi aksi yang ingin dilakukan oleh pengguna.

Contoh:

Masukan	> <b>Move</b>
---------	---------------

- **Tombol Pada Keyboard.** Pengguna menekan tombol pada *keyboard* yang telah di-assign untuk melakukan aksi tertentu.

Contoh:

Masukan	a = pindah ke kiri satu grid s = pindah ke bawah satu grid f = shop
---------	---

Sebagai bentuk interaksi, aplikasi harus memberikan respon setiap kali pengguna melakukan suatu aksi. **Contoh:** Setiap pengguna melakukan serangan. Ditampilkan pesan, HP milik monster A berkurang sebanyak 32.

## Spesifikasi Program

Untuk implementasi Tugas Besar ini peserta diwajibkan untuk menggunakan bahasa pemrograman Java. Terkait tampilan aplikasi, peserta cukup membuat tampilan pada *console ( terminal )*, tidak perlu membuat GUI.

Dalam melakukan perancangan kelas, harap menerapkan konsep-konsep Pemrograman Berorientasi Objek yang telah diajarkan di kuliah ( *Inheritance, Interface, Abstract Class*, dll ). Penamaan *File* disesuaikan dengan nama kelas yang terdapat pada *file* tersebut.

## Laporan

Hal-hal yang harus tercakup dalam laporan Tugas Besar ini antara lain:

- **Halaman Cover**, berisi Kode dan Nama Kuliah, Nama Tugas, Nomor Kelas, Nomor Kelompok, NIM dan Nama Anggota Kelompok, Logo ITB, jurusan, Nama Sekolah (STEI), dan Perguruan Tinggi (ITB), dan Tahun (2015).
- **Daftar Isi.**
- **Daftar Tabel** (jika ada).
- **Daftar Gambar** (jika ada).
- **Daftar pembagian kerja anggota kelompok**, mulai dari perancangan kelas, implementasi, hingga pengujian.

Contoh:

Kelas	Perancang kelas (Atribut, fungsi, dsb)	Pelaksana Implementasi	Penguji
Character	FNA	SA	FNA
Area	TW/BY	TW	DPS
...	BY/DPS	YW	NUM
....	...	...	...

**Keterangan:** FNA: Fazat Nur Azizah, SA: Saiful Akbar, TW: Tricya Widagdo, DPS: Dicky Prima Satya, BY: Bayu Hendradjaya, YW: Yani Widyani, NUM: Nur Ulfa Maulidevi

### ● Checklist hasil Pekerjaan.

Contoh Checklist:

Kelas	Desain	Implementasi	Testing
Character	V	V	-
Area	V	V	V
p...	V	-	X
....	...	...	...

**Keterangan:** V: sudah selesai dikerjakan, -: dikerjakan, tapi belum selesai, X: tidak dikerjakan sama sekali.

- **Deskripsi persoalan:** berisi penjelasan kembali (dengan “bahasa sendiri”) tentang persoalan yang akan diselesaikan.
- **Rancangan untuk setiap kelas.** Bagian ini berisi penjelasan mengenai atribut-atribut yang terdapat dalam setiap kelas serta deskripsi metode-metode pada kelas yang dianggap perlu dijelaskan ( metode selain Constructor, Getter, dan Setter ) dalam notasi algoritmik.
- **Penjelasan Konsep OOP.** Bagian ini berfungsi untuk menjelaskan konsep-konsep OOP yang diterapkan dalam Tugas Besar ini. Setiap konsep OOP ( Inheritance, Interface, dsb ) memiliki bagian masing-masing dan dijelaskan bagaimana konsep tersebut diterapkan pada pengerjaan Tugas Besar ini. Contoh penjelasan: Pada Tugas besar ini diterapkan interface Battle untuk karakter dan monster karena aksi yang dilakukan saat *battle* sama dan harus seragam. Penjelasan yang diberikan harus lebih detail dan komprehensif dengan *reasoning* yang jelas.
- **Implementasi Tugas Besar.** Berisi Driver dari tugas besar serta implementasi dari kelas-kelas yang telah dirancang sebelumnya.
- **Pengujian (testing).** Berisi pengujian dari modul-modul Tugas Besar yang telah dijelaskan dalam spesifikasi. Pengujian harus meliputi skenario dimana *input*/masukan yang diberikan memberikan kondisi salah ( Misal, mengetikkan perintah yang tidak didefinisikan, menekan angka yang tidak ada daftar pilihan, dsb ). Hasil pengujian berupa apakah aplikasi yang dibuat memberikan hasil sesuai dengan yang diharapkan atau tidak.

### Keterangan Tambahan

Deadline pengerjaan Tugas Besar ini adalah sekitar 3 minggu sejak waktu rilis ( 26 April 2015 ). Dan selama pengerjaannya peserta kuliah diwajibkan untuk melakukan asistensi sebanyak 3 kali. Asistensi dilakukan minimal sekali setiap minggunya. Peserta diperbolehkan melakukan asistensi melebihi jumlah yang diwajibkan. Alokasi Asisten serta mekanisme menghubungi asisten akan dijelaskan pada dokumen terpisah.

Apabila ada pertanyaan lebih lanjut mengenai Tugas Besar ini, dapat menghubungi asisten melalui email [13511030@std.stei.itb.ac.id](mailto:13511030@std.stei.itb.ac.id).

- Selamat Mengerjakan! -