**BAB 3**

**STUDI KASUS**

Bab ini mendeskripsikan *game* yang digunakan sebagai studi kasus untuk di-*refactor* dalam penelitian.

**3.1 Rangkuman studi kasus**

Dalam subbab ini akan dijelaskan mengenai detail dari *game* studi kasus yang akan digunakan dalam eksperimen. Penjelasan akan dilakukan untuk sisi permainan dan struktur *project*.

**3.1.1 *Game***

Subbab ini menjelaskan *game* studi kasus dilihat sebagai sebuah *game* atau aplikasi.

**3.1.1.1 Ringkasan Umum**

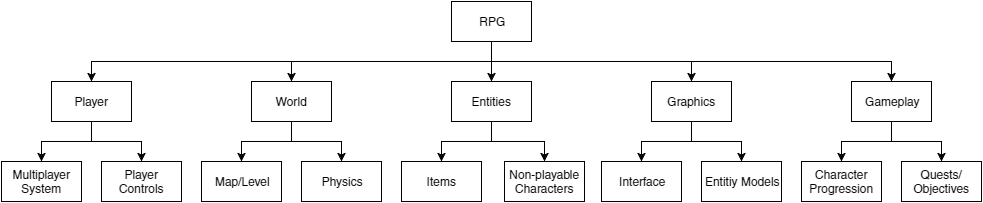
*Game* yang digunakan sebagai obyek studi kasus adalah Dungeon Crawl Stone Soup. Dungeon Crawl Stone Soup merupakan sebuah *game* yang pertama kali dikembangkan pada tahun 2006 sebagai upaya untuk menghidupkan kembali proyek *Linley’s Dungeon Crawl* yang dikembangkan oleh Linley Henzell pada tahun 1997.

Dungeon Crawl Stone Soup dapat dimenangkan dengan *win condition* tertentu, yaitu mendapatkan 3 dari 15 buah benda bernama *runes of zot*, mengambil sebuah benda bernama *orb of zot*, dan keluar dari *dungeon* dengan selamat.

**3.1.1.2 Genre dan Mekanika *Gameplay***

Pada umumnya, sebuah *game* dapat dikategorikan ke dalam sebuah genre. Genre dari sebuah *game* adalah jenis, tipe, atau kelompok dari *game* tersebut berdasarkan beberapa hal seperti cerita, mekanisme *gameplay*, model grafis, jumlah pemain, dan lain-lain. Dungeon Crawl Stone Soup dapat dikategorikan dalam genre RPG, atau *role-playing game*. RPG sendiri dapat didefinisikan sebagai sebuah *game* di mana pemain mengimitasi atau memerankan sebuah tokoh atau peran fiktif dalam sebuah cerita atau dunia, biasanya berlatar fantasi. Aslinya, RPG tidak terikat dalam *game* video. RPG berawal dari permainan papan (*board game*) atau permainan berbasis teks (*text-based*). Dalam studi kasus ini, RPG yang dimaksud adalah RPG dalam arti *game* video, di mana elemen-elemen *role-playing* yang terdapat dalam *game* tersebut berupa latar dan sistem yang terkomputerisasi.

*Game* yang termasuk dalam kategori genre RPG biasanya memiliki beberapa ciri khusus seperti pemain memainkan sebuah karakter, terdapat dunia—dinamis atau statis—yang dapat dijelajahi pemain, dan memiliki *character progression*. Dalam Dungeon Crawl Stone Soup, pemain memainkan sebuah tokoh yang memiliki peran tertentu, di mana tokoh tersebut dapat menjelajahi sebuah dunia yang dinamis. Setiap aksi signifikan yang dilakukan oleh pemain menghasilkan elemen-elemen tertentu yang secara langsung mengubah karakteristik tokoh yang dimainkan. Hal tersebut merupakan sebuah sistem *character progression*.

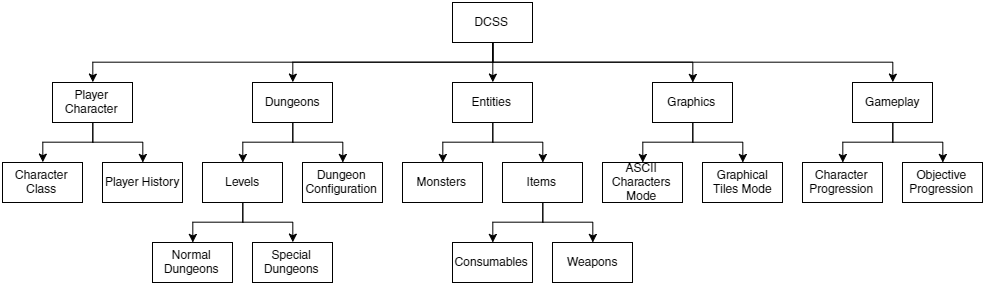


Gambar 3.1 Diagram fitur RPG secara umum

Dalam genre RPG sendiri terdapat beberapa sub-genre yang mengkategorikan sistem permainan dan elemen *role-playing* lebih lanjut. Pembagian sub-genre untuk RPG pun terbagi menjadi beberapa dasar pengelompokan, seperti berdasarkan unsur atau elemen *game* yang paling menonjol, model skenario permainan, jumlah pemain, dan lain-lain. Dungeon Crawl Stone Soup dapat dikategorikan ke dalam sub-genre *roguelike*.

*Roguelike* merupakan sub-genre dari RPG yang mengandung unsur *dungeon crawler* dan biasanya terdiri dari sejumlah *level* yang dihasilkan secara prosedural. Beberapa karakteristik *roguelike* adalah *turn-based gameplay*, grafis *tile-based*, dan *permanent death* dari karakter yang dimainkan. Salah satu ciri khas *roguelike* adalah grafis yang ditampilkan dengan karakter ASCII dan hanya dapat dimainkan menggunakan *keyboard*, walaupun beberapa *game* bersub-genre *roguelike* modern telah menyediakan fitur *graphical tile* yang menggunakan gambar untuk merepresentasikan elemen-elemen dalam *game*. *Game* dengan sub-genre *roguelike* biasanya lebih mengedepankan kompleksitas mekanik *gameplay* dibandingkan dengan *interface* yang ramah dengan pemain. *Turn-based gameplay* pada *roguelike* biasanya merupakan sistem *real-time* yang dimodifikasi sehingga waktu dalam dunia *game* tersebut hanya akan berjalan jika pemain melakukan suatu aktivitas. Dalam Dungeon Crawl Stone Soup, setiap aktifitas dilakukan dengan satu ketikan tombol di *keyboard* atau *mouse*. Lama waktu yang berjalan dalam satu ketikan tersebut bergantung pada lama waktu aktivitas yang dilakukan. Selain *turn-based gameplay*, Dungeon Crawl Stone Soup juga memiliki sistem *permanent death*, di mana karakter yang terbunuh dalam *game* akan hilang selamanya, tanpa bisa di-*load* kembali seperti dalam *game* bergenre lain. Walaupun karakter yang telah terbunuh tidak dapat di-*load*, pemain dapat melakukan *save* dan *load* untuk karakter yang masih hidup, untuk menyimpan dan melanjutkan *progress* dalam *game*.

Unsur *dungeon crawl* yang terdapat dalam *roguelike* merupakan sebuah sistem atau skenario di mana tujuan atau aktivitas utama pemain adalah menjelajahi sebuah atau sejumlah *dungeon*. Dalam *dungeon* yang dijelajahi, pemain dapat bertarung dengan karakter-karakter musuh dan mengumpulkan benda-benda yang memiliki signifikansi dalam sistem *character progression*. *Game* dengan model skenario *dungeon crawl* biasanya memiliki sistem *loot*, di mana pemain dapat mengumpulkan benda atau barang-barang yang tersedia di dalam dunia *game* tersebut dengan berbagai cara, mulai dari mengambil barang yang tersedia di tempat-tempat tertentu seperti misalnya emas di dalam peti harta, atau benda yang didapat dari karakter musuh yang telah dikalahkan.



Gambar 3.2 Diagram fitur Dungeon Crawl Stone Soup

Seperti kebanyakan *game* dengan genre *roguelike*, Dungeon Crawl Stone Soup memiliki latar fantasi, di mana dunia yang dijelajahi merupakan sebuah dunia fiktif dengan karakter-karakter yang berasal dari berbagai mitologi. Dalam Dungeon Crawl Stone Soup, unsur cerita tidak terlalu ditonjolkan, hanya disebutkan mengenai tujuan permainan yang merupakan *win condition* dari *game* ini. Tingkah laku pemain tidak memiliki dampak terhadap cerita latar dan sebaliknya, cerita latar tidak memiliki dampak apapun terhadap permainan, kecuali sebagai penjelas *win condition*. Pemain tidak dapat mengubah cerita dan *win condition* dengan cara apapun.

Setiap sesi permainan Dungeon Crawl Stone Soup memiliki konfigurasi *dungeon* yang berbeda. Fitur ini dibuat agar *game* memiliki *replay value* yang tinggi, sehingga pemain tidak merasa bosan. Fitur *random/procedurally-generated levels* di *game-gameroguelike* juga dimaksudkan agar pemain tidak terdorong untuk menghafal pola *game* dan lebih menitikberatkan permainan pada strategi dan mempelajari kesalahan dari sesi-sesi sebelumnya.

Selain *dungeon level* yang *procedurally-generated*, fitur lain yang menaikkan *replay value* dan mengurangi unsur penghafalan pola adalah *scroll* dan *potion* yang memiliki efek yang di-*random* setiap sesinya. *Scroll* dan *potion* adalah salah satu elemen penting dalam *game* bertema fantasi dengan sistem *dungeon crawl*. *Scroll* merupakan gulungan perkamen yang memiliki efek sihir tertentu, sedangkan *potion* merupakan minuman obat atau pada kasus tertentu merupakan minuman racun. Dengan sistem seperti ini, pemain akan menjadi lebih berhati-hati ketika menggunakan *scroll* atau *potion*, karena harus memperhitungkan risiko yang terjadi jika benda yang digunakan memiliki efek samping yang buruk.

**3.1.1.3 Grafis**

Dungeon Crawl Stone Soup memiliki dua mode grafis, yaitu *console* dan *graphical tile*. Dalam mode *console*, *game* dijalankan di sebuah *console* dansemua elemen *game* direpresentasikan dalam karakter ASCII. Dalam mode *graphical tile*, *game* dijalankan dalam tampilan *user interface* interaktif, di mana semua elemen *game* memiliki representasi grafis berupa gambar 2D yang menyerupai deskripsi obyek yang direpresentasikan. Karena sistem yang digunakan adalah *tile-based*, maka elemen-elemen atau obyek yang berukuran lebih dari satu *tile* digambarkan melalui gabungan dari beberapa *tile*, bukan merupakan sebuah gambar tunggal.

**3.1.1.4 *Interface***

Mode *graphical tile* menyediakan tampilan *interface* untuk *inventory* pemain yang dapat diakses dan dilihat dengan mudah, langsung saat bermain. *Interface* mode *graphical tiles* secara umum memiliki lebih banyak tampilan, di mana tampilan yang terdapat di layar *game* tidak hanya karakter pemain dan dunia di sekitarnya, tetapi juga peta dari *level* dungeon yang sedang dijelajahi, visualisasi daftar *inventory* atau barang-barang yang dimiliki oleh pemain, daftar *skill* atau jurus yang dapat dilakukan oleh pemain, dan daftar *command* untuk membuat karakter pemain melakukan kegiatan tertentu secara otomatis.

**3.1.1.5 *Controls***

Masing-masing mode grafis yang disediakan Dungeon Crawl Stone Soup memiliki mekanisme *controls* yang berbeda. Mode *console* hanya dapat dikendalikan menggunakan *keyboard*, sedangkan mode *graphical tile* dapat dikendalikan dengan menggunakan *keyboard* dan *mouse*.

**3.1.1.6 Audio**

Dungeon Crawl Stone Soup menyediakan fitur suara yang dapat diaktifkan dengan mengubah *file* konfigurasi secara manual. Secara *default*, fitur suara tidak diaktifkan.

**3.1.1.7 Platform**

Untuk saat ini, Dungeon Crawl Stone Soup tersedia untuk *operating system* Windows, Mac OS X, dan Linux. Pengembangan *game* untuk platform Android saat ini masih berlangsung, namun *unstable build* terbaru dapat diunduh dari situs resmi Dungeon Crawl Stone Soup. Dungeon Crawl Stone Soup dapat dimainkan secara *local*, melalui *executable file* di platform Windows dan melalui *terminal* di platform Linux. Selain *local*, tersedia juga versi *web browser* yang menyediakan Dungeon Crawl Stone Soup. Versi yang disediakan di *web browser* adalah *graphical tile version*.

Untuk eksperimen, pengerjaan dilakukan pada platform Linux karena kemudahan dalam proses *installation* dan *compile*.

**3.1.1.8 *Availability***

Dungeon Crawl Stone Soup dapat diunduh secara gratis melalui situs resmi. Situs Dungeon Crawl Stone Soup juga menyediakan tautan untuk mengakses dan mengunduh repositori *source-code* yang tersedia secara ­*open-source*. Dungeon Crawl Stone Soup dapat diunduh melalui tautan <https://crawl.develz.org/download.htm>.

**3.1.1.9 Lisensi**

Dungeon Crawl Stone Soup dirilis di bawah lisensi GNU General Public License v2.0 atau setelahnya, yang berarti semua orang berhak mengakses, mengunduh, menyebarkan, dan mengubah *game* ini dengan bebas.

**3.1.2 *Source Code***

Subbab ini menjelaskan *source-code* dari *game* studi kasus.

**3.1.2.1 *Overview***

*Source-code* Dungeon Crawl Stone Soup di-*host* di situs repositori GitHub, di mana *code* dapat diakses dan diubah dengan bebas oleh kontributor-kontributor yang telah terdaftar.

**3.1.2.2 *Repository***

Repositori Dungeon Crawl Stone Soup dapat diakses melalui tautan <https://github.com/crawl/crawl>.

**3.1.2.3 Bahasa**

Bahasa pemrograman yang digunakan dalam pengembangan Dungeon Crawl Stone Soup adalah C/C++. Meskipun format *code* yang digunakan adalah C++, tetapi paradigma *code* yang digunakan belum menerapkan konsep *object-oriented*.

**3.1.2.4 *Packages***

Dungeon Crawl Stone Soup memanfaatkan beberapa *package* yang tersedia secara *open-source*, yaitu:

* Lua scripting language, yang digunakan untuk fungsionalitas *in-game* dan *user macros*.
* PCRE library, yang digunakan untuk *regular expressions*.
* SQLite library, yang digunakan sebagai *engine* dari *database*.
* SDL and SDL\_image libraries, yang digunakan untuk tampilan *tile*.
* libpng library, yang digunakan untuk memuat gambar *tile*.

**3.1.2.5 *Project Hierarchy***

Seluruh *source-code* utama yang membentuk *game* Dungeon Crawl Stone Soup terdapat dalam satu buah folder dengan direktori crawl/crawl-ref/source/. Setiap *file* yang mengandung *source-code* ditempatkan dalam satu *folder* tersebut, tanpa dipisahkan berdasarkan pengelompokan tertentu.

**3.1.2.6 Instalasi dan*Building***

Karena bahasa yang digunakan dalam *source code* adalah C/C++, maka untuk kemudahan eksperimen, *game* akan di-*install* pada platform Linux. Untuk platform Linux, instalasi dilakukan dengan memasukkan beberapa *command* di Terminal sebagai berikut:

# Install the source repository

echo 'deb https://crawl.develz.org/debian crawl 0.18' | sudo tee -a /etc/apt/sources.list

# Install the DCSS signing key

wget https://crawl.develz.org/debian/pubkey -O - | sudo apt-key add -

# update your package list

sudo apt-get update

# install console version

sudo apt-get install crawl

*Command* di atas digunakan untuk melakukan instalasi Dungeon Crawl Stone Soup versi *console*. Instalasi untuk versi *graphical tile* dapat menggunakan *command* berikut:

# install tiles version

sudo apt-get install crawl-tiles

Setelah instalasi selesai dilakukan, Dungeon Crawl Stone Soup dapat di-*compile* dengan *command* berikut:

# go to file directory

cd crawl/crawl-ref/source/

# compile console version

make

*Command* di atas digunakan untuk *compilegame* versi *console*. Untuk *compile* versi *graphical tile* dapat menggunakan *command* berikut:

# compile tiles version

make TILES=y

Setelah selesai *compile*, Dungeon Crawl Stone Soup dapat dijalankan dengan *command*:

# launch the game

./crawl

**3.1.2.7 *Dependencies***

Untuk sistem Debian, *dependencies* dapat diunduh dengan *command* berikut:

apt-get install build-essential libncursesw5-dev bison flex liblua5.1-0-dev libsqlite3-dev libz-dev pkg-config libsdl2-image-dev libsdl2-mixer-devlibsdl2-dev libfreetype6-dev libpng-dev ttf-dejavu-core

**3.2 Dream sheep commit (Entity Addition)**

Dream Sheep merupakan sebuah *entity* baru yang ditambahkan pada DCSS oleh seorang kontributor bernama isloat pada Agustus\* 2016. Menurut deskripsi yang ditulis oleh isloat di halaman *commit*,ide mengenai *entity* ini dicetuskan oleh kontributor bernama PleasingFungus beberapa tahun lalu.

Dream Sheep merupakan sebuah karakter musuh yang muncul secara bergerombol pada level-level tingkat lanjut. Dream Sheep juga dapat muncul sebagai bagian dari gerombolan *entity* lain bernama Torpor Snail.

*Entity* ini menyerang pemain dengan cara memberikan *status effect* negatif pada pemain seperti *sleep*. Menurut isloat, *code* yang mengimplementasi kemampuan memberikan *sleep* tersebut terinspirasi dari *code* \_starcursed\_scream dan \_will\_starcursed\_scream milik kontributor lain bernama DracoOmega.

Kemampuan atau *ability* memberikan *sleep* yang dimiliki oleh Dream Sheep bernama Sleep Dust. Sleep Dusthanya dapat digunakan oleh lebih dari satu ekor Dream Sheep. Jika terdapat dua sampai empat ekor Dream Sheep di lingkup pandangan pemain, maka mereka akan mencoba untuk menidurkan pemain untuk waktu yang singkat. Sedangkan jika dalam lingkup pandangan pemain terdapat lima ekor atau lebih Dream Sheep, maka mereka akan mencoba untuk menidurkan pemain untuk waktu yang lama.

Sleep Dust memiliki *cooldown* yang relatif lama, yaitu 8 hingga 12 *turn*.*Ability* ini hanya dapat dilakukan jika semua Dream Sheep yang ada telah siap untuk melakukan *ability* tersebut. Tidak ada *damage* terhadap pemain yang dihasilkan dari *ability* ini.

Dream Sheep merupakan sebuah *entity* yang bersifat *recursive summoner*, yaitu dapat memunculkan atau memanggil *entity* lain dengan jenis yang sama dengan dirinya. Seekor Dream Sheep dapat memunculkan Dream Sheep lain ketika *health*-nya rendah. Tidak seperti *recursive summoner* lain, Dream Sheep hanya dapat memunculkan satu ekor Dream Sheep.

Secara kekuatan, Dream Sheep merupakan sebuah versi yang lebih kuat dari *entity* Sheep biasa. Dibandingkan dengan Sheep biasa, Dream Sheep sedikit lebih tangguh, lebih sering menghindar dari serangan pemain, dan memiliki *magic resistance* yang lebih tinggi.

*Code changes:*

* enum.h

Pada berkas ini, dilakukan penambahan *if-condition* untuk menangani kompatibilitas *entity* baru dengan *save profile* yang menggunakan versi lama yang tidak terdapat *entity* baru tersebut.

*Code* yang baru mengecek *tag* versi *game*. Jika *game* memiliki *tag* versi 34, maka monster Sheep dimunculkan. Jika *game* memiliki *tag* versi yang lebih tinggi dari versi 34, maka monster Dream Sheep yang dimunculkan. Penambahan juga dilakukan untuk menempatkan Dream Sheep ke dalam kelompok monster baru.

* godwrath.cc

Berkas ini berisi *function-function* terkait *god* atau dewa dalam *game*. Dungeon Crawl Stone Soup memiliki sistem penyembahan dewa di mana pemain dapat menjadi penyembah seorang dewa. *Function-function* tersebut sebagian besar menangani efek dari pemujaan dewa tertentu, serta konfigurasi monster-monster pengikut-pengikut setiap dewa. Pada berkas ini, tidak terdapat pengubahan yang mengubah baris *code*, pengubahan hanya terjadi pada sebuah baris *comment* terkait *entity* Sheep di mana kata ‘Sheep’ menjadi ‘Dream Sheep’.

* mgen-enum.h

Pada berkas ini, dilakukan pengubahan pada pendefinisian *band* di mana *band* Sheep diubah menjadi *band* Dream Sheep. *Band* merupakan sejumlah *entity* yang sama yang beraktifitas secara bergerombol.

* mon-abil.cc

Pada berkas ini, dilakukan implementasi *ability* Dream Dust yang dimiliki oleh Dream Sheep. Ada beberapa *function* dan *method* yang ditambahkan.

Pertama adalah *method*\_sheep\_message() yang menghasilkan output berupa teks yang mendeskripsikan aksi-aksi yang dilakukan oleh Dream Sheep, atau efek-efek yang dialami oleh pemain ketika berinteraksi dengan Dream Sheep. Terdapat beberapa tipe teks yang ditambahkan. Setiap tipe teks memiliki variasi masing-masing sesuai dengan ketepatan *event* yang terjadi.

* + Teks untuk mendeskripsikan tingkat kekuatan*ability* yang dilancarkan kumpulan Dream Sheep.
  + Teks untuk mendeskripsikan efek yang dialami karakter non-pemain ketika terkena Dream Dust.
  + Teks untuk mendeskripsikan efek yang dialami pemain ketika terkena Dream Dust.

Kedua adalah *method*\_dream\_sheep\_sleep() yang memiliki beberapa fungsi:

* + Mengecek sasaran yang akan diserang oleh seekor Dream Sheep. Jika tidak ada sasaran atau sasaran yang diincar oleh Dream Sheep telah mati, maka *function* akan melakukan *return*.
  + Mengatur posisi dan gerakan gerombolan Dream Sheep, serta menghitung jumlah Dream Sheep dalam sebuah gerombolan
  + Men-*generate* kemungkinan berhasilnya sebuah gerombolan Dream Sheep untuk melakukan *ability*.
  + Memanggil *method*\_sheep\_message() untuk mengeluarkan output sesuai dengan hasil perhitungan dalam *method* ini.
  + Menentukan durasi Sleep yang dialami karakter sasaran ketika Dream Dust berhasil.

Ketiga adalah *case*MONS\_DREAM\_SHEEP yang memiliki fungsi:

* + Mengecek apakah Dream Sheep tidak dalam kondisi bingung (*confused*) dan tidak sedang melarikan diri (*fleeing*), lalu mengecek apakah karakter sasaran tidak berada dalam sebuah *sanctuary*, sedang tidak tidur, dan tidak memiliki kekebalan terhadap Sleep.
  + Jika semua kondisi di atas terpenuhi, maka Dream Sheep memiliki peluang 20% untuk melakukan *ability*.
* mon-data.h

Pada berkas ini, Sheep ‘didaftarkan’ sebagai monster lama yang sudah tidak dipakai untuk versi baru, hanya disimpan untuk kompatibilitas versi lama. Selain itu, pengubahan juga banyak dilakukan untuk mengganti setiap *instance* dan konfigurasi yang terkait dengan Sheep agar menjadi Dream Sheep.

* mon-ench.cc

Secara umum, berkas ini berisi *function-function* yang mengatur karakteristik setiap *enchantment* yang terdapat dalam *game*, serta efeknya terhadap monster-monster yang ada. *Enchantment* merupakan jurus, atau *status effect* dari jurus yang dilancarkan oleh sebuah karakter atau *entity* dalam *game*, yang dapat memberikan efek tertentu pada sasaran yang menerima serangan jurus tersebut. Pada berkas ini, dilakukan pengubahan untuk *case*ENCH\_STICKY\_FLAME yang mengatur implementasi *spell* Sticky Flame. *Spell* tersebut memiliki keunikan berupa efektifitas yang sangattinggi terhadap monster dengan *genus*MONS\_SHEEP. *Genus* merupakan kelompok monster yang masih memiliki relasi biologis atau anatomis yang dekat, fiktif atau non-fiktif. Perubahan yang dilakukan adalah mengganti efek *spell* tersebut yang tadinya ditujukan secara spesifik untuk monster Sheep, menjadi untuk kelompok *genus* dari Sheep, yang mengandung Sheep dan Dream Sheep.

* mon-pick-data.h

Berkas ini berisi data mengenai kemunculan monster seperti lokasi dan peluang muncul. Data tersebut disimpan dalam beberapa *const* yang masing-masing mewakilkan kelompok monster berdasarkan lokasi *spawn* atau muncul. Berkas *header* ini dimanfaatkan oleh berkas mon-pick.cc. Pada berkas ini, dilakukan penambahan data untuk MONS\_DREAM\_SHEEP, dan penghapusan data untuk MONS\_SHEEP.Dream Sheep dan Sheep diprogram untuk *spawn* di kategori level *dungeon* yang bernama Lair.

* mon-place.cc

Berkas ini berisi *function-function* yang digunakan ketika menempatkan monster di dalam *dungeon*. Bagian yang diubah pada berkas ini merupakan:

* + *const*map<monster\_type, band\_set> bands\_by\_leader yang berisi daftar monster khusus yang memiliki peluang untuk *spawn* diikuti dengan sebuah *band* monster tersebut atau *band* monster lain. Pengubahan yang dilakukan adalah:
    - Mengganti konfigurasi kemunculan *band* Sheep menjadi untuk *band* Dream Sheep.
    - Mengganti perhitungan peluang 20% untuk kemunculan *band* Sheep yang mengikuti monster Cyclops menjadi peluang 100% untuk kemunculan *band* Dream Sheep.
  + *const*map<branch\_type, band\_weights> bands\_pickyang berisi penanganan *band-band* yang muncul mengikuti monster-monster dengan karakteristik kepemimpinan *band* khusus. Pengubahan yang dilakukan adalah:
    - Mengganti konfigurasi untuk kemunculan *band* Sheep yang mengikuti monster Torpor Snail menjadi untuk *band* Dream Sheep. Konfigurasi ini terdapat pada sebuah *case* tersendiri untuk MONS\_TORPOR\_SNAIL di instansiasi band\_type \_choose\_band().
  + *const*map<band\_type, vector<member\_possibilites>> band\_membership yang berisi daftar *band* monster beserta konfigurasi kemungkinan monster-monster yang terdapat dalam *band* tersebut. Pengubahan yang dilakukan adalah:
    - Mengganti konfigurasi komposisi *band* Sheep menjadi untuk *band* Dream Sheep.
* mon-spell.h

Berkas ini berisi daftar *spell* yang dimiliki oleh semua monster yang terdapat dalam *game*.*Spell* disusun berdasarkan monster yang memiliki *spell* tersebut.Pada berkas ini, terdapat penambahan berupa sebuah kategori *spell* yang dimiliki oleh Dream Sheep. Kategori tersebut berisi sebuah *spell* yang memiliki nama obyek SPELL\_SLEEP.

* player-reacts.cc

Berkas ini berisi *function-function* yang berhubungan dengan karakter pemain. Sebagian besar *function* yang terdapat dalam berkas ini digunakan untuk mengatur *enchantment* pada karakter pemain. Pada berkas ini, terdapat pengubahan *typo* untuk *command*you.awake() menjadi you.awaken(). *Command* ini berfungsi untuk membangunkan pemain setelah Sleep yang dialami pemain kadaluarsa.

* player.cc

Berkas ini berisi *function-function* yang berhubungan dengan pemain. Perbedaan dengan player-reacts.cc adalah player.cc berisi *function* untuk penggunaan yang lebih umum. Terdapat beberapa penambahan dan pengubahan pada berkas ini:

* + Penambahan *bool*player::can\_sleep yang memeriksa apakah pemain dapat dibuat tidur oleh *enchantment* Sleep. Fungsi ini me-return true jika pemain tidak memiliki kekebalan terhadap Sleep.
  + Pengubahan *typo* untuk *method*player::awake() menjadi player::awaken()
  + Pengubahan instansiasi nilai kekebalan pemain terhadap Sleep di player::awaken(). Sebelum diubah, pemain di-set untuk memiliki kekebalan selama 1 turn. Setelah pengubahan, pemain hanya di-set untuk memiliki kekebalan selama 1 turn jika pada saat fungsi dipanggil pemain tidak memiliki kekebalan sama sekali.
  + Pengubahan *typo* untuk pemanggilan *method*awake() menjadi awaken() di *method*player::check\_awaken() yang memeriksa apakah pemain sudah terbangun.

Note: fungsi ini memiliki ~8000 baris code. Apa bisa dilakukan improvement agar code sebanyak itu tidak menumpuk ? Atau sudah best approach ?

* player.h

Berkas ini merupakan *header* untuk player.cc. Isi dari berkas ini merupakan inisialisasi variabel-variabel yang digunakan di player.cc. Pengubahan yang dilakukan pada berkas ini adalah pengubahan *typo* untuk inisialisasi awake() menjadi awaken().

Pull Requests related to Dream Sheep

* Sheep no spell-sleep <https://github.com/crawl/crawl/pull/335>

Pull Request ini memiliki beberapa *commit*:

* + Remove emergency sleep spell from dream sheep

Dikutip dari deskripsi Pull Request Dream Sheep: “As an emergency-flagged spell, dream sheep will directly cast Sleep instead of relying on dream dust”.

Dream Sheep dalam jumlah kecil sudah memiliki peluang untuk membuat pemain tertidur dengan menggunakan Dream Dust, sehingga *emergency* Sleep dirasa tidak perlu dan membingungkan pemain, terutama karena Sleep mengecek Magic Resistance sedangkan Dream Dust tidak.

Berkas *source-code* yang diubah:

* + - mon-spell.h

Fungsi MST\_DREAM\_SHEEP yang berisi Sleep dihapus dari *source code*.

* + Remove SPFLAG\_BATTLE

Menghilangkan fungsi SPFLAG\_BATTLE dari *source code* karena tidak terpakai (sepertinya tidak berhubungan dengan Dream Sheep).

Berkas *source code* yang diubah:

* + - spl-cast.h

Inisialisasi *enum* SPFLAG\_BATTLE dihapus dari *source code*.

* + - spl-data.h

SPFLAG\_BATTLE dihapus dari semua fungsi di *source code*.

* + Make dream dust a proper monspell

Memindahkan *code* implementasi Dream Dust dari mon-abil.cc ke mon-spell.h dan mon-cast.cc. “Dream Dust” dapat dilihat sebagai *natural ability* di layar xv (belum dibaca lebih lanjut apa itu “xv”, ada yang menuliskan sebagai “x-v”. Dari apa yang sudah dibaca (terdapat di crawl-ref/docs/develop/levels/advanced.txt) ‘xv’ merupakan command (sepertinya untuk Lua) untuk menampilkan deskripsi lengkap dari sebuah monster).

Berkas *source code* yang diubah:

* + - enum.h

SPELL\_DREAM\_DUST ditambahkan ke dalam daftar *spell*.

* + - mon-abil.cc

Implementasi Dream Dust dihapus dari berkas ini.

* + - mon-cast.cc

Dream Dust diimplementasi di berkas ini. Penjelasan terdapat di beberapa Materi Tambahan sebelumnya.

* + - mon-spell.h

MST\_DREAM\_SHEEP berisi SPELL\_DREAM\_DUST beserta konfigurasinya ditambahkan dalam berkas ini.

* + - spl-data.h

Penambahan Dream Dust beserta data dan konfigurasinya.

* Make Clarity resist sleep <https://github.com/crawl/crawl/pull/331>

Untuk memberikan kegunaan lain pada Clarity yang cukup langka dan sulit untuk didapatkan, ditambahkan fungsi agar Clarity dapat memberikan kekebalan kepada pemain terhadap Sleep. Fungsi ini telah dibahas di Materi Tambahan sebelumnya.

Berkas *source code* yang diubah:

* + actor.cc

Penambahan fungsi yang mengecek Clarity pada pemain yang menjadi sasaran Sleep.

* Remove chain sleeps <https://github.com/crawl/crawl/pull/334>

Untuk mencegah Sleep yang berkelanjutan atau berkepanjangan, ditambahkan sebuah fungsi yang membuat pemain mendapatkan kekebalan terhadap Sleep selama 2 hingga 4 *turn*. Fungsi ini telah dibahas di Materi Tambahan sebelumnya.

Berkas *source code* yang diubah:

* + mon-abil.cc

Menghilangkan implementasi kekebalan yang lama di mana kekebalan diberikan dalam *turn* yang sama dengan saat pemain terkena Sleep. Kekebalan yang lama memiliki durasi 4 hingga 8 *turn* tanpa melihat durasi Sleep yang dialami pemain.

* + player.cc

Mengganti fungsi kekebalan Sleep yang lama menjadi fungsi baru dengan durasi kekebalan 2 hingga 4 *turn*.

* Reduce sleep damage bonus <https://github.com/crawl/crawl/pull/336>

Mengurangi *damage multiplier* terhadap karakter yang sedang terkena Sleep dari 2.5x menjadi 2.0x.

Note: Tidak jadi dipakai. Pull Request ini dibuat sebagai antisipasi jika Dream Sheep dirasa terlalu kuat atau mengganggu. Menurut **isloat** Pull Request ini tidak jadi dibutuhkan karena implementasi Pull Request #334 dan #335 telah berfungsi dengan baik untuk menanggulangi masalah *balancing* Dream Sheep.

Berkas *source code* yang diubah:

* + melee\_attack.cc

Pengubahan *damage* yang diterima oleh karakter yang sedang tertidur dari dikalikan dengan 5/2 menjadi dikalikan dengan 2.

Issues with Dream Sheep:

Isu-isu berikut merupakan komplain dari komunitas pemain Dungeon Crawl Stone Soup terkait penambahan karakter Dream Sheep dalam *game* yang diambil dari laman *web*<https://www.reddit.com/r/dcss/comments/502w5p/please_nerf_or_remove_dream_sheep_preferably_by/>

* Dream Dust tidak mengecek Magic Resistance

Menurut pemain-pemain yang telah menghadapi Dream Sheep dalam *game*, Dream Sheep terlalu mudah membuat pemain tertidur dengan *ability* Dream Dust-nya. Hal ini disebabkan oleh mekanika Dream Dust yang hanya mengecek Clarity, yaitu sebuah *intrinsic* (sebuah kekuatan khusus yang dimiliki pemain melalui cara tertentu) yang melindungi karakter pemain dari gangguan psikis. Banyak pemain yang beranggapan bahwa sebaiknya Dream Dust juga mengecek Magic Resistance pemain, yaitu ketahanan pemain terhadap jurus atau *ability* sihir musuh. Dengan demikian, pemain yang tidak memiliki Clarity memiliki cara atau kesempatan untuk sepenuhnya menangkal Dream Dust.

* Kekebalan terhadap Sleep belum terimplementasi dengan baik

Menurut salah satu *developer* Dungeon Crawl Stone Soup dengan *username***amalloy** di *thread* Reddit di atas, seharusnya pemain menerima kekebalan terhadap Sleep selama beberapa *turn* setelah pemain mengalami Sleep, tetapi untuk sekarang fungsi tersebut belum bekerja.

* Mekanika yang belum diimplementasi

Perihal pengecekan Magic Resistance dalam implementasi Dream Dust sebenarnya sudah dibahas oleh **isloat** sebagai *contributor* Dream Sheep dalam deskripsi *commit­-*nya. **isloat** mengatakan bahwa jika pengecekan Clarity saja tidak cukup untuk mencegah Dream Sheep menjadi terlalu kuat atau mengganggu, maka pengecekan Magic Resistance akan ditambahkan dalam implementasi Dream Dust. Opini: Kemungkinan besar pengecekan Magic Resistance tidak dilakukan sejak awal untuk menghindari perubahan *code* dalam jumlah besar, sehingga menyulitkan *contributor* dan memperpanjang masa *development*.

* Terlalu menyusahkan pemain

Karena poin-poin di atas, Dream Sheep terasa terlalu kuat atau terlalu menyusahkan pemain. Contoh kasus atau situasi di mana Dream Sheep terasa terlalu menyusahkan pemain adalah ketika *band* Dream Sheep muncul berdekatan dengan karakter musuh lainnya yang jauh lebih kuat. *Ability* Dream Dust dari Dream Sheep dapat dengan mudah membuat karakter pemain tertidur, sehingga karakter musuh lain dapat menyerang karakter pemain dengan leluasa tanpa perlawanan. Penempatan Dream Sheep di level yang cukup lanjut, di mana sebagian besar monster yang menghuni level tersebut memiliki kekuatan dan kemampuan kelas tinggi, sehingga ketika pemain dibuat tertidur, musuh-musuh tersebut dapat membunuh pemain hanya dengan sedikit serangan. Dalam beberapa kasus yang dialami pemain, musuh dapat membunuh pemain dengan sekali serang. Waktu bermain ketika berhadapan dengan Dream Sheep pun menjadi sangat sedikit karena ketika pemain terkena Sleep, pemain langsung dibunuh oleh karakter musuh lain.

* Penghapusan karakter Sheep dari *game*.

Penambahan Dream Sheep dalam *game* membuat *entity* Sheep biasa dihapus dari *game*. Hal ini menerima respon yang sangat bercampur di komunitas pemain. Beberapa pemain menyebutkan bahwa Sheep memang sepantasnya dihilangkan dari *game* karena merupakan karakter yang tidak berguna dan tidak memiliki efek atau pengaruh terhadap permainan. Di sisi lain, banyak pemain yang menyatakan kecewa terhadap penghapusan karakter Sheep dari *game*. Pemain-pemain tersebut menjelaskan bahwa Sheep memang sangat lemah dan tidak berguna, namun memiliki unsur hiburan tersendiri bagi pemain. Sheep seringkali menjadi bahan lelucon atau menjadi bulan-bulanan pemain, sehingga memberikan kesenangan atau kepuasan tersendiri bagi beberapa pemain yang berhadapan dengan Sheep.

Development issues:

* Kompleksitas *source code*

Pada umumnya, penambahan *entity* terutama karakter monster baru dalam *game* Dungeon Crawl Stone Soup mengharuskan *contributor* atau *developer* untuk mengubah lebih dari 20 buah berkas *source code* dan teks deskripsi. Kerumitan ini dapat membuat pengembang *game* ragu-ragu untuk mengimplementasi perubahan yang signifikan, atau hal-hal kecil yang implementasi *code*-nya tersebar dalam banyak berkas*.* Pada contoh kasus Dream Sheep, masalah ini dapat dilihat (atau diperkirakan/asumsikan terjadi) pada masalah pengecekan Magic Resistance yang masih berupa *To-Do* yang baru akan dikerjakan jika mekanika yang sudah diimplementasi dirasa kurang baik, di mana sebaiknya (atau seharusnya) pengecekan Magic Resistance diimplementasi sejak awal, hanya saja implementasi pengecekan Magic Resistance harus dilakukan dengan mengubah banyak berkas *code*.

* *Entity Balancing*

Berdasarkan poin-poin masalah yang dibahas sebelumnya, dapat ditarik kesimpulan bahwa salah satu isu yang signifikan dalam penambahan *entity* ke dalam *game* adalah *balancing*atau penyeimbangan karakteristik *entity* tersebut dengan lingkungan (dalam konteks ini, mekanika dan karakteristik *entity* lain dalam *game*). Keseimbangan yang dimaksud adalah ketika *entity* yang ditambahkan tidak memiliki perbedaan yang signifikan dengan *entity* lain dalam *game*, tidak mengubah pengalaman bermain yang dirasakan pemain secara signifikan, tidak terlalu kuat dan menyulitkan bagi pemain, serta tidak terlalu lemah dan tidak berefek terhadap permainan.

* *Gameplay Balancing*

Selain penyeimbangan *entity*, perlu dilakukan juga penyeimbangan mekanika *gameplay* terhadap *entity*. Untuk kasus penambahan *entity* secara masif atau signifikan (perlu contoh kasus, tapi belum ditemukan, tapi yakin ada), kurang efektif dan efisien jika penyeimbangan diberatkan pada *entity*, di mana setiap *entity* perlu menyesuaikan diri dengan mekanika yang ada. Untuk kasus tersebut, ada baiknya bila mekanika *gameplay* yang berubah mengikuti *entity-entity* baru yang ditambahkan. Pengubahan atau penyesuaian *gameplay* juga dapat dilakukan untuk men-*supportentity* yang ada demi pengalaman bermain yang lebih imersif. Contohnya adalah penggunaan *scripted difficulty* pada level terakhir Ace Combat: Assault Horizon, di mana pada umumnya semua karakter musuh memiliki *artificial intelligence* sendiri untuk bertempur, tetapi khusus untuk karakter musuh *boss* pada level terakhir, semua gerakan*boss*dan *event* yang berhubungan dengan *boss* tersebut sudah ditentukan dalam *script* untuk memberikan kesan *thrill* yang lebih kuat pada pemain.