

Penerapan Algoritma Divide and Conquer untuk Berburu Monster dalam Permainan Final Fantasy III

Juyus Muhammad Adinulhaq¹, Safuan²

^{1,2} Program Studi Informatika, Universitas Muhammadiyah Semarang, Semarang, Indonesia

Info Artikel

Histori Artikel:

Diterima, 10 September 2022

Revisi, 15 Desember 2022

Disetujui, 31 Januari 2023

Article History:

Received, 10 September 2022

Revised, 15 December 2022

Accepted, 31 January 2023

Kata kunci:

Final Fantasy III

Berburu

Monster

Penerapan

Permainan

ABSTRAK

Final Fantasy III adalah salah satu permainan RPG yang cukup populer pada masanya. Dalam permainan salah satu nilai jualnya adalah cerita dalam permainan tersebut. Salah satu hal yang penting dalam permainan ini adalah proses penguatan karakter untuk mempermudah dalam menyelesaikan permainan ini. Dalam artikel ini, akan dibahas tentang penerapan algoritma divide and conquer yaitu merge sort dalam permainan Final Fantasy III. Algoritma ini digunakan untuk mengurutkan data musuh yaitu Health Poin(HP). Penggunaan algoritma ini akan membuat proses pengurutan lebih cepat dan efisien. Hasil dari penerapan algoritma ini adalah memudahkan pemain untuk memperkuat karakter yang digunakan dengan cara mengalahkan berbagai monster sehingga membantu dalam menyelesaikan permainan Final Fantasy III.

ABSTRACT

Final Fantasy III is one of the most popular RPG games of its time. In the game one of the selling points is the story in the game. One of the important things in this game is the process of strengthening characters to make it easier to finish this game. In this article, we will discuss the application of the divide and conquer algorithm, namely merge sort, in Final Fantasy III. This algorithm is used to sort enemy data, namely Health Points (HP). Using this algorithm will make the sorting process faster and more efficient. The result of implementing this algorithm is that it makes it easier for players to strengthen the characters used by defeating various monsters so that it helps in completing the game Final Fantasy III.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Penulis Korespondensi:

Juyus Muhammad Adinulhaq

Program Studi Informatika, Fakultas Teknik, Universitas Muhammadiyah Semarang

Universitas Muhammadiyah Semarang

Jl. Kedungmundu No. 18, Semarang, Jawa Tengah, Indonesia

Email: lagirebahan2@gmail.com

1. PENDAHULUAN

Final Fantasy III adalah video game yang dikembangkan dan diterbitkan oleh Square untuk Family Computer yang akhirnya dikenal sebagai NES(Nintendo Entertainment System). Permainan ini awalnya dirilis di Jepang pada tanggal 27 April 1990, namun hanya rilis di Jepang. Pada 24 Agustus 2006. Final Fantasy III versi Remake dikembangkan oleh Matrix Software yang dirilis di Amerika Utara atau Eropa.

Final Fantasy III adalah role-playing game di mana pemain mengendalikan sekelompok empat karakter, yang dikenal sebagai "Light Warriors", mereka memulai perjalanan untuk mengembalikan cahaya ke dunia dan mengembalikan keseimbangan dunia. Salah satu nilai jual dalam permainan ini adalah

ceritanya, pemain bisa menyelesaikan bagian-bagian cerita, menjelajahi berbagai lokasi, mengalahkan berbagai monster untuk memperkuat karakter, berinteraksi dengan NPC, dan melawan musuh menggunakan pertarungan berbasis giliran. Pemain juga dapat meningkatkan karakter yang dimainkan untuk menjadi lebih kuat.

Dalam proses menyelesaikan cerita permainan, terkadang pemain kesulitan dalam mengalahkan monster atau musuh, di saat kondisi tersebut pemain dituntut untuk mengalahkan monster-monster dan berbagai musuh yang tersedia dalam dunia game tersebut untuk memperkuat karakter yang digunakan pemain. Namun, apabila pemain secara acak melawan monster, pemain bisa berhadapan dengan monster yang sangat kuat dan belum siap dihadapi oleh karakter yang digunakan pemain, untuk mempermudah pemain dalam memilih dan mengurutkan monster mana yang lebih baik dilawan terlebih dahulu dibuatlah artikel ini dengan menggunakan algoritma divide and conquer, adanya artikel ini diharapkan pemain bisa terbantu dalam menyelesaikan permasalahan tersebut.

2. METODE

2.1. Algoritma Divide dan Conquer

Divide and conquer merupakan algoritma yang mempunyai konsep memecah-mecah suatu masalah menjadi bagian-bagian kecil atau upa-masalah, sehingga lebih mudah untuk diselesaikan. Pemecahan masalah tersebut dilakukan hingga masalah tersebut tidak dapat pecah lagi, kemudian solusi dari setiap bagian masalah tersebut digabungkan untuk menjadi solusi yang utuh, menyelesaikan setiap submasalah secara mandiri, dan menggabungkan solusi dari submasalah untuk menemukan solusi akhir dari masalah utama.

Algoritma Divide and Conquer dalam penerapannya memiliki beberapa tahapan, yaitu:

- Divide adalah membagi permasalahan menjadi upa-masalah.
- Conquer menyelesaikan masing-masing upa-masalah (secara langsung atau secara rekursif).
- Combine adalah menggabungkan solusi masing-masing upa-masalah untuk menemukan solusi akhir dari masalah utama.

2.2. Algoritma Merge Sort

Merge Sort merupakan algoritma pengurutan yang menggunakan metode divide and conquer. Prinsip dari algoritma ini adalah membagi array atau list data menjadi 2 bagian yang lebih kecil, lalu masing-masing bagian dari array tersebut diurutkan secara terpisah, dan akhirnya dua bagian tersebut digabungkan kembali menjadi satu array yang terurut.

Pseudocode dari Algoritma Merge Sort

```
function mergeSort(arr)
  if (length of arr <= 1) return arr
  var middle = floor(length of arr / 2)
  var left = arr[0...middle]
  var right = arr[middle...length of arr]
  left = mergeSort(left)
  right = mergeSort(right)
  return merge(left, right)

function merge(left, right)
  var result = []
  var leftIndex = 0
  var rightIndex = 0
  while (leftIndex < length of left and rightIndex < length of right)
    if (left[leftIndex] <= right[rightIndex])
      append left[leftIndex] to result
      leftIndex++
    else
      append right[rightIndex] to result
      rightIndex++
  while (leftIndex < length of left)
    append left[leftIndex] to result
    leftIndex++
  while (rightIndex < length of right)
    append right[rightIndex] to result
    rightIndex++
  return result
```

Keterangan:

- **arr** adalah array yang akan diurutkan dengan menggunakan algoritma merge sort.
- **middle** adalah nilai tengah dari panjang array **arr**, yang diperoleh dengan membagi panjang array dengan 2 dan dibulatkan ke bawah menggunakan fungsi **floor**.
- **left** adalah array yang berisi elemen dari index 0 hingga middle dari array **arr**.
- **right** adalah array yang berisi elemen dari index middle hingga panjang array **arr**.
- **mergeSort(left)** dan **mergeSort(right)** mengurutkan array **left** dan **right** secara rekursif menggunakan algoritma merge sort.
- **merge(left, right)** digunakan untuk menggabungkan array **left** dan **right** yang telah diurutkan menjadi satu array yang terurut. Proses ini dilakukan dengan mengecek elemen pada index ke-*i* pada array **left** dan **right**, lalu memasukkan elemen yang lebih kecil ke dalam array **result**.
- **leftIndex** dan **rightIndex** digunakan untuk mengetahui indeks pada array **left** dan **right** yang sedang dibandingkan.
- Proses pengurutan dan penggabungan dilakukan sampai semua elemen pada array **left** dan **right** sudah digabungkan ke dalam array **result**.

2.3. Final Fantasy III

Final Fantasy III adalah video game yang dikembangkan dan diterbitkan oleh Square pada tanggal 27 April 1990, namun hanya rilis di Jepang. Lalu pada tanggal 24 Agustus 2006, Final Fantasy III versi Remake yang dikembangkan oleh Matrix Software dirilis. Permainan ini berjenis role-playing game (RPG) dengan tipe pertarungan turn based strategy (TBS), di mana pemain mengendalikan sekelompok empat karakter, yang dikenal sebagai "Light Warriors", mereka memulai perjalanan untuk mengembalikan cahaya ke dunia dan mengembalikan keseimbangan dunia.

Role-playing game (RPG) adalah jenis permainan di mana pemain menjalankan peran sebagai karakter fiktif yang ditentukan dalam cerita atau dunia yang diciptakan oleh pengembang permainan. Dalam RPG, pemain biasanya akan mengambil keputusan yang mempengaruhi alur cerita dan hasil akhir permainan, serta pemain mengembangkan karakter mereka melalui interaksi dengan lingkungan dan NPC (Non-Player Character) yang ada dalam permainan.

Turn-based adalah sistem permainan yang digunakan dalam beberapa jenis permainan, terutama dalam RPG, di mana pemain dan musuh bergantian melakukan aksi dalam suatu periode waktu yang ditentukan. Dalam sistem ini, pemain akan menentukan aksi yang akan dilakukan oleh karakternya dalam satu putaran, kemudian sistem akan mengeksekusi aksi tersebut dan memberikan giliran ke musuh untuk melakukan aksi.

Permainan ini berfokus pada cerita, pemain juga bisa menjelajahi berbagai lokasi di dalam permainan, mengalahkan berbagai monster untuk memperkuat karakter, berinteraksi dengan NPC, dan melawan musuh menggunakan pertarungan berbasis giliran. Pemain juga dapat meningkatkan karakter yang dimainkan untuk menjadi lebih kuat. Dalam proses menyelesaikan cerita permainan, terkadang pemain kesulitan dalam mengalahkan monster atau musuh, di saat kondisi tersebut pemain dituntut untuk mengalahkan monster monster dan berbagai musuh yang tersedia dalam dunia game tersebut untuk memperkuat karakter yang digunakan pemain.

Proses untuk memperkuat karakter yang dilakukan berulang kali biasa disebut dengan grinding, grinding bisa dilakukan dengan berbagai cara, salah satunya dengan berburu monster, dengan mengalahkan monster-monster karakter pemain mendapatkan suatu poin yang dinamakan EXP(Experience) dengan terkumpulnya poin EXP tersebut karakter dari pemain bisa naik level lalu nilai status dari karakter pemain bertambah.

3. HASIL DAN PEMBAHASAN

Penerapan algoritma divide and conquer pada pemilihan monster di permainan Final Fantasy III berfokus pada pengurutan HP(Health Poin) dari monster yang ingin dilawan oleh pemain. Pemain dapat menganalisis situasi dan strategi permainan untuk menentukan monster mana yang paling mudah dan paling cocok untuk dilawan terlebih dahulu sesuai dengan kondisi dari karakter yang digunakan pemain.

3.1 Final Fantasy III

Untuk melakukan penerapan algoritma ini diperlukan data dari monster dalam permainan tersebut, dataset yang akan digunakan tidak begitu besar, hanya terdapat 16 data monster pada permainan Final Fantasy III, data monster-monster yang saya ambil bisa dilihat pada tabel dibawah ini:

Tabel 1. Data 16 Monster Final Fantasy III
Sumber : [https://finalfantasy.fandom.com/wiki/Bestiary_\(Final_Fantasy_III\)](https://finalfantasy.fandom.com/wiki/Bestiary_(Final_Fantasy_III))

NO.	GAMBAR	NAMA	HEALTH POIN (HP)
1.		Helldiver	120
2.		Rust Bird	135
3.		Rukh	155
4.		Basilisk	100
5.		Bugbear	110
6.		Mandrake	120
7.		Leprechaun	142
8.		Darkface	168
9.		Petit	103
10.		Poison Bat	98
11.		Lilliputian	118
12.		Wererat	130
13.		Blood Worm	165

3.2 Hasil penerapan algoritma

Berdasarkan tabel data di atas, kita mendapatkan nilai sebagai berikut:

120	135	155	100	110	120	142	168	103	98	118	130	165
135	173	155										

Selanjutnya kita coba urutkan angka-angka diatas menggunakan Algoritma *merge sort*. Langkah 1 (Pembagian/Pemecahan):

120	135	155	100	110	120	14
135	173	155				

Langkah 2:

103	98	118	130	165
-----	----	-----	-----	-----

120	135	155	100
165	135	173	155

Langkah 3:

110	120	142	168
-----	-----	-----	-----

103	98	118	130
-----	----	-----	-----

120	135
165	135

Langkah 4:

155	100
173	155

110	120
-----	-----

142	168
-----	-----

103	98
-----	----

118	130
-----	-----

120

135

155

100

110

120

142

168

103

98

118

130

165

135

173

155

Langkah 5 (Penggabungan dan Pengurutan)

120	135
135	165

Langkah 6:

100	155
155	173

110	120
-----	-----

142	168
-----	-----

118	130
-----	-----

100	120	135	155
135	165	155	173

Langkah 7:

110	120	142	168
-----	-----	-----	-----

98	103	118	130
----	-----	-----	-----

100	110	120	120	135	142	15
155	165	173				

Langkah 8:

98	103	118	130	135
----	-----	-----	-----	-----

98	100	103	110	118	120	120	130	135	135	142	155	155
165	168	173										

Selanjutnya kita coba untuk menerapkan algoritma tersebut dalam program python dengan source code dibawah ini:

Prosedur Merge:

```

procedure Merge(input/output A : TabelInt, input kiri,tengah,kanan :
integer)
{ Menggabung tabel A[kiri..tengah] dan tabel A[tengah+1..kanan]
menjadi tabel A[kiri..kanan] yang terurut menaik.
Masukan: A[kiri..tengah] dan tabel A[tengah+1..kanan] yang sudah
terurut menaik.
Keluaran: A[kiri..kanan] yang terurut menaik.
}
Deklarasi
B : TabelInt
i, kidal1, kidal2 : integer
Algoritma:
kidall←kiri      { A[kiri .. tengah] }
kidal2←tengah + 1 { A[tengah+1 .. kanan] }
i←kiri
while (kidall ≤ tengah) and (kidal2 ≤ kanan) do
  if Akidall ≤ Akidal2 then
    Bi←Akidall
    kidall←kidall + 1
  else
    Bi←Akidal2
    kidal2←kidal2 + 1
  endif
  i←i + 1
endwhile
{ kidall > tengah or kidal2 > kanan }

{ salin sisa A bagian kiri ke B, jika ada }
while (kidall ≤ tengah) do
  Bi←Akidall
  kidall←kidall + 1
  i←i + 1
endwhile
{ kidall > tengah }

{ salin sisa A bagian kanan ke B, jika ada }
while (kidal2 ≤ kanan) do
  Bi←Akidal2
  kidal2←kidal2 + 1
  i←i + 1
endwhile
{ kidal2 > kanan }

{ salin kembali elemen-elemen tabel B ke A }
for i←kiri to kanan do
  Ai←Bi
endfor
{ diperoleh tabel A yang terurut membesar }

```

Gambar 1. Source code algoritma merge sort

Dengan menggunakan program diatas kita mendapatkan hasil pengurutan, sebagai berikut: [98, 100, 103, 110, 118, 120, 120, 130, 135, 135, 142, 155, 155, 165, 168, 173]. Hasil tersebut sesuai atau sama dengan tanpa menggunakan coding, jadi sudah bisa dianggap benar, dan berdasarkan hasil diatas, kita mendapatkan pengurutan, sebagai berikut:

Tabel 1. Data 16 Monster Final Fantasy III
 Sumber : [https://finalfantasy.fandom.com/wiki/Bestiary_\(Final_Fantasy_III\)](https://finalfantasy.fandom.com/wiki/Bestiary_(Final_Fantasy_III))

NO.	GAMBAR	NAMA	HEALTH POIN (HP)
1.		Poison Bat	120
2.		Basilisk	135
3.		Petit	155
4.		Bugbear	100
5.		Lilliputian	110
6.		Helldiver	120
7.		Mandrake	142
8.		Wererat	168
9.		Rust Bird	103
10.		Killer Fish	98
11.		Leprechaun	118
12.		Rukh	130
13.		Sea Elemental	165

Data yang terurut sudah didapatkan, sehingga dengan adanya data dan metode ini diharapkan pemain dapat melawan monster mulai dari yang termudah (yang mempunyai nilai HP terkecil) hingga yang tersulit, karena jika pemain melawan musuh secara acak maka bisa saja terjadi kondisi dimana status karakter belum mencukupi untuk melawan suatu monster, jika dimulai dari yang terlemah maka karakter pemain akan bertambah kuat seiringnya melawan para monster, saat bertambah kuat pemain bisa melawan monster yang lebih kuat begitu seterusnya.

4. KESIMPULAN DAN SARAN

Final Fantasy III adalah Video Game yang mempunyai gameplay Role Playing-Game dengan salah satu nilai jualnya yaitu cerita dalam permainan tersebut, namun terkadang ada kondisi dimana pemain kesulitan menyelesaikan suatu bagian cerita yang diharuskan mengalahkan suatu monster atau musuh, sehingga pemain diharuskan untuk melakukan penguatan karakter dengan cara grinding, dengan melakukan grinding ini pemain bisa memperkuat karakter yang dimainkan, dalam melakukan proses grinding ini jika pemain secara acak melawan monster, pemain bisa saja menemukan kondisi dimana karakter yang dimainkan belum siap untuk melawan monster, sehingga akan membuang banyak waktu pemain.

Metode Divide and Conquer dengan Algoritma Merge Sort menjadi salah satu pilihan yang tepat untuk membantu pemain untuk memilih musuh dalam proses grinding, karena pemain dapat mengetahui musuh mana yang sekiranya lebih mudah dihadapi terlebih dahulu.

REFERENSI

- [1] Munir, R. 2018. Algoritma Divide and Conquer (Bagian 1). [https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2017-2018/Algoritma-Divide-and-Conquer-\(2018\).pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2017-2018/Algoritma-Divide-and-Conquer-(2018).pdf)
- [2] Munir, R. 2018. Algoritma Divide and Conquer (Bagian 2). [https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2006-2007/Algoritma%20Divide%20and%20Conquer%20\(Bagian%202\).pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2006-2007/Algoritma%20Divide%20and%20Conquer%20(Bagian%202).pdf)
- [3] Final Fantasy III – Wikipedia. https://en.wikipedia.org/wiki/Final_Fantasy_III
- [4] Bestiary (Final Fantasy III) | Final Fantasy Wiki | Fandom . [https://finalfantasy.fandom.com/wiki/Bestiary_\(Final_Fantasy_III\)](https://finalfantasy.fandom.com/wiki/Bestiary_(Final_Fantasy_III))
- [5] Role-playing game – wikipedia. https://en.wikipedia.org/wiki/Role-playing_game
- [6] Strategi berbasis giliran - Wikipedia bahasa Indonesia, ensiklopedia bebas https://id.wikipedia.org/wiki/Strategi_berbasis_giliran
- [7] Arti dan Penjelasan Grinding di Game RPG dan MOBA (leskompi.com). <https://www.leskompi.com/arti-grinding-di-game/>
- [8] Algoritma Merge Sort – Arfian Hidayat. <https://pesonainformatika.com/programming/apa-itu-merge-sort-dan-cara-penyelesaiannya/>