

# Penerapan Algoritma Decrease and Conquer dalam Hacking Minigame pada Permainan Fallout 4

Ryan Kurnia Hidayatullah - 13519212  
Program Studi Teknik Informatika  
Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung  
13519212@std.stei.itb.ac.id

**Abstract**—Fallout 4 merupakan sebuah permainan *action role-play* yang populer pada masanya. Game ini merupakan salah satu game *open world* yang mana player memiliki kebebasan untuk berkeliaran di dalam game ini. Dalam permainan ini terdapat *Hacking minigame* dimana player akan menebak password dalam sebuah komputer dan player memiliki 4 kali kesempatan untuk menebak. *Hacking minigame* ini dapat diselesaikan dengan mengaplikasikan algoritma *decrease and conquer*.

**Keywords**—Fallout 4, Hacking, Minigame, Decrease and Conquer.

## I. PENDAHULUAN

Pada zaman sekarang terutama pada masa pandemi COVID-19 ini, orang-orang sulit untuk keluar rumah dikarenakan adanya protokol kesehatan yang berlaku sehingga orang-orang yang lelah bekerja atau mahasiswa yang lelah mengerjakan tugas bingung cara menghabiskan waktu untuk *refreshing* dan bersenang-senang. Salah satu cara menghabiskan waktu dan bersenang-senang di rumah yang paling menyenangkan adalah bermain *video game*. Salah satu permainan digital yang menyenangkan untuk dimainkan sendiri (*single player*) adalah Fallout 4. Fallout 4 merupakan sebuah permainan *action role-play* yang juga merupakan *open-world game* yang dikembangkan oleh Bethesda Game Studio dan dirilis pada tanggal 10 November 2015.



**Gambar 1.1** Cover Permainan Fallout 4

Sumber : <https://gamestation.co.id/mod-fallout-4-kini-sudah-bisa-diakses-di-xbox-one/>



**Gambar 1.2** Gameplay Fallout 4

Sumber : <https://www.nextgenbase.com/fallout-4-what-makes-you-s-p-e-c-i-a-l-part-2-perception/>

Seperti game *open-world* pada umumnya, player dapat bereksplorasi di dalam dunia pada permainan ini. dan juga memiliki *minigame*. Salah satu *minigame* pada permainan ini yaitu *hacking minigame* yang mana player akan diminta untuk menebak password dari banyaknya kata yang ditampilkan. Player memiliki 4 kali kesempatan untuk menebak *password*. List kandidat password yang ditampilkan dapat dieliminasi dengan menghapus kandidat *password* yang tidak sesuai setelah menebak pertama dan kembali melakukan langkah tersebut hingga mendapatkan password yang diinginkan sehingga permainan ini dapat diselesaikan menggunakan algoritma *decrease and conquer* untuk mengeliminasi kandidat password yang tidak sesuai.

## II. DASAR TEORI

### A. Decrease and Conquer

Algoritma *Decrease and Conquer* merupakan metode perancangan algoritma dengan mereduksi persoalan menjadi dua upa-persoalan (*sub-problem*) yang lebih kecil, tetapi selanjutnya hanya memproses satu sub-persoalan saja. Berbeda dengan *Divide and Conquer* yang memproses semua upa-persoalan dan menggabung semua solusi setiap sub-persoalan.

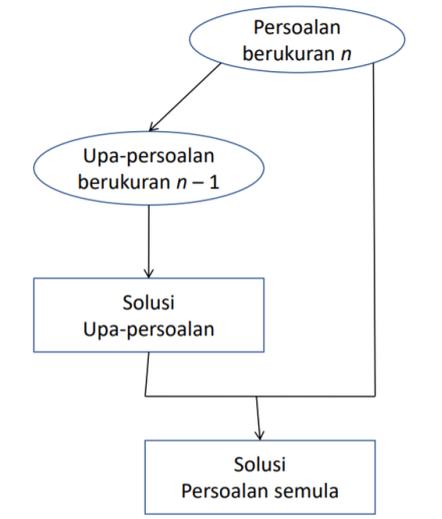
Di dalam literatur lama, semua algoritma yang membagi persoalan menjadi dua upa-persoalan yang lebih kecil dimasukkan ke kategori divide and conquer. Meskipun demikian, tidak kedua upa-persoalan hasil pembagian diselesaikan. Jika hanya satu upa-persoalan yang diselesaikan, maka tidak tepat dimasukkan sebagai algoritma divide and conquer. Mereka dikategorikan sebagai decrease and conquer.

Algoritma *Decrease and Conquer* terdiri dari dua tahapan :

1. *Decrease* yaitu mereduksi persoalan menjadi beberapa persoalan yang lebih kecil (biasanya dua upa-persoalan).
  2. *Conquer* yaitu memproses satu upa-persoalan secara rekursif
- Tidak ada tahap combine dalam decrease and conquer, karena hanya satu upa-persoalan yang diselesaikan.

Tiga varian decrease and conquer :

1. *Decrease by a constant* yaitu ukuran instans persoalan direduksi sebesar konstanta yang sama setiap iterasi algoritma. Biasanya konstanta = 1. Contoh persoalan : *perpangkatan, selection sort, insertion sort*.

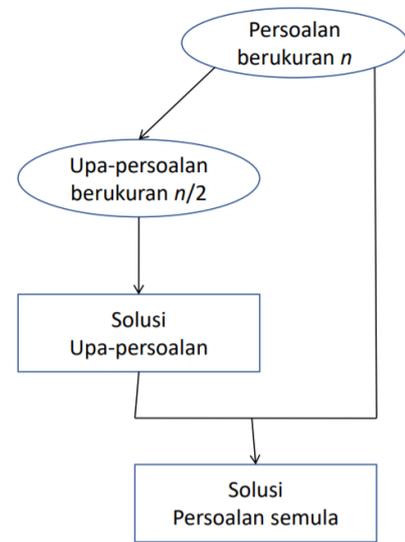


**Gambar 2.1 Ilustrasi Decrease by a Constant**

Sumber :

<https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Decrease-and-Conquer-2021-Bagian1.pdf>

2. *Decrease by a constant factor* yaitu ukuran instans persoalan direduksi sebesar faktor konstanta yang sama setiap iterasi algoritma. Biasanya faktor konstanta = 2. Contoh persoalan : *binary search, mencari koin palsu*.

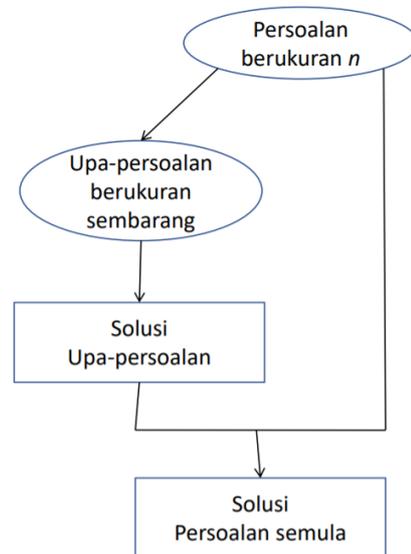


**Gambar 2.2 Ilustrasi Decrease by a Constant Factor**

Sumber :

<https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Decrease-and-Conquer-2021-Bagian1.pdf>

3. *Decrease by a variable size* yaitu ukuran instans persoalan direduksi bervariasi pada setiap iterasi algoritma. Contoh persoalan : *interpolation search, mencari nilai median*.



**Gambar 2.3 Ilustrasi Decrease by a Variable Size**

Sumber :

<https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Decrease-and-Conquer-2021-Bagian2.pdf>

Karena pada makalah ini kita akan menyelesaikan persoalan dengan menggunakan *Decrease by a variable size*. Maka akan

diberikan contoh persoalan yang diselesaikan dengan varian *Decrease and Conquer* tersebut. Berikut merupakan satu contoh persoalan yang diselesaikan dengan Algoritma *Decrease and Conquer* dengan varian *Decrease by a variable size*.

### 1. Interpolation Search

Algoritma pencarian ini mirip dengan pencarian kata di dalam kamus atau di dalam ensiklopedi dengan cara memperkirakan letak kata tersebut di di dalam kamus yang mana semua entri di dalam kamus sudah terurut menaik (dari A sampai Z). Algoritma ini yaitu memperkirakan letak kata di dalam larik dilakukan dengan teknik interpolasi dengan kondisi awal yaitu larik A sudah terurut menaik dan K adalah nilai yang dicari.

```

procedure Interpolationsearch(input A : LarikInteger, i, j : integer; K : integer; output idx : integer)
{ Mencari elemen bernilai K di dalam larik A[i..j] dengan interpolation search.
Masukan: larik A sudah terurut menaik, K sudah terdefinisi nilainya
Luaran: indeks lariks sedemikian sehingga A[idx] = K
}
Deklarasi
mid : integer
Algoritma:
if i > j then { ukuran larik sudah 0 }
idx ← -1 { K tidak ditemukan }
else
mid ← i + (j - i) * (K - A(i)) / (A(j) - A(i))
if A(mid) = K then { K ditemukan }
idx ← mid { indeks elemen larik yang bernilai = K }
else
if A(mid) > K then
Interpolationsearch(A, i, mid - 1, K, idx) { cari di upalarik kiri, di dalam larik A[i..mid] }
else
Interpolationsearch(A, mid + 1, j, K, idx) { cari di upalarik kanan, di dalam larik A[mid+1..j] }
endif
endif
endif

```

Gambar 2.4 Pseudocode persoalan Interpolation Search

Sumber :

<https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Decrease-and-Conquer-2021-Bagian2.pdf>

Kompleksitas algoritma interpolation search :

- Kasus terburuk: O(n), untuk sembarang distribusi data
- Kasus terbaik: O(log log n), jika data di dalam larik terdistribusi uniform

### B. Hacking Minigame Fallout 4

Hacking merupakan cara mengeksploitasi kelemahan dalam sebuah sistem komputer atau jaringan komputer. Hal ini juga berlaku dalam permainan Fallout 4. Hacking dalam permainan Fallout 4 mirip dengan sebuah board game yang bernama Mastermind. Pemain akan disuguhkan dengan daftar kumpulan kata-kata yang berbeda dengan panjang yang sama yang ditentukan dari tingkat kesulitan Hacking tersebut.

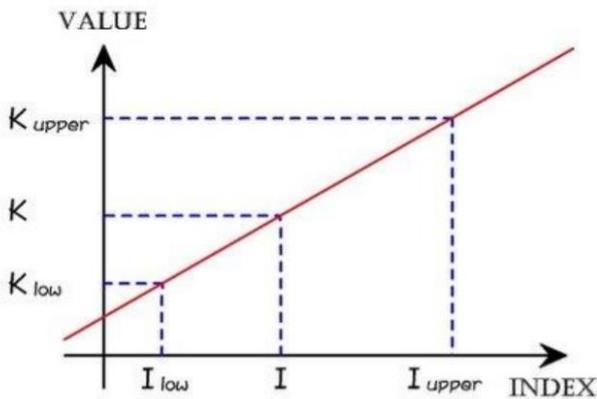


Gambar 2.7 Ilustrasi Hacking Minigame dalam permainan Fallout 4

Sumber : <https://www.tentonhammer.com/guides/fallout-4-hacking-mini-game-guide>

Hacking dalam game Fallout 4 merupakan sebuah cara untuk mendapatkan akses lebih di dunia *open world* ini. Hacking dapat digunakan untuk membuka pintu yang dikunci menggunakan sebuah sistem, membuka brankas, melemahkan pertahanan musuh, dan lain-lain.

Mekanisme permainan ini yaitu pemain akan ditampilkan daftar kumpulan kata, salah satu kata tersebut merupakan



Gambar 2.4 Grafik perbandingan K dengan I

Sumber :

<https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Decrease-and-Conquer-2021-Bagian2.pdf>

Dengan rumus perbandingan Gradien maka kita akan mendapatkan perkiraan posisi K dalam larik

$$I = I_{low} + (I_{upper} - I_{low}) \times \frac{K - K_{low}}{K_{upper} - K_{low}}$$

Gambar 2.5 Rumus perkiraan posisi K dalam larik

Sumber :

<https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Decrease-and-Conquer-2021-Bagian2.pdf>

I low : indeks ujung kiri larik

I upper : indeks ujung kanan larik

K low : elemen minimum di dalam larik

K upper : elemen maksimum di dalam larik

password yang sebenarnya. Pemain akan diminta untuk menebak password yang benar dari kumpulan kata tersebut. Pemain akan mengorbankan 1 kesempatan untuk menguji *likeliness* dari sebuah kata yang dipilih. *Likeliness* merupakan banyaknya huruf dengan urutan yang sama dengan password yang aslinya, ini dapat menjadi petunjuk untuk menyelesaikan permainan ini dan menebak *password* yang sebenarnya. Semakin besar *likeliness* maka akan semakin mirip kata tersebut dengan password yang aslinya. Pemain akan tetap menebak hingga mendapatkan kata yang ditebak sama dengan *password* yang sebenarnya.

Berikut merupakan contoh kode untuk menghitung *likeliness*

```
def likeliness(word1, word2):
    match = 0
    if len(word1) == len(word2):
        for i in range(len(word1)):
            if word1[i] == word2[i]:
                match += 1
    return match
```

Gambar 2.8 Source Code untuk Menghitung Likeliness dalam Bahasa Pemrograman Python

### III. PEMBAHASAN

Penjelasan cara berpikir Algoritma *Decrease and Conquer* pada *hacking minigame* akan dijelaskan dengan melakukan simulasi *minigame* dengan menggunakan list kata yang ada di gambar 2.7.

Pada gambar 2.7 kita mendapatkan daftar kumpulan kata yaitu

**AGES**  
**WOOD**  
**SPOT**  
**GETS**  
**WHEN**  
**POTS**  
**FLAT**  
**DRAG**  
**WEAK**  
**HUTS**  
**EGOS**

Gambar 3.1 List Kandidat Password

Karena kita akan melakukan simulasi *mini game* maka kita akan mengasumsikan kata unik yang merupakan *password* yang sebenarnya adalah DRAG.

Langkah pertama yang akan dilakukan pemain adalah mengorbankan satu langkah untuk menguji kata pertama dan kecocokannya dengan password yang sebenarnya. Pada langkah ini kita akan memilih kata AGES, maka kita mendapatkan *likeliness* AGES dengan DRAG yaitu 0 yang berarti tidak ada kemiripan kata sama sekali. Maka kita akan menghapus AGES kata beserta semua kata yang memiliki *likeliness* > 0 dari *list* kandidat password.

Langkah selanjutnya kita mencari kata-kata yang memiliki *likeliness* > 0 dengan menghitung *likeliness* dari kata AGES dengan kata lainnya.

Kata pertama	Kata kedua	Likeliness
AGES	WOOD	0
AGES	SPOT	0
AGES	GETS	1
AGES	WHEN	1
AGES	POTS	1
AGES	FLAT	0
AGES	DRAG	0
AGES	WEAK	0
AGES	HUTS	1
AGES	EGOS	2

Tabel 3.1 Likeliness kata AGES dengan kata lainnya

Dari tabel 3.1 kita mendapatkan kata yang memiliki kemiripan minimal 1 dengan AGES yaitu GETS, WHEN, POTS, HUTS, dan EGOS. Maka kita akan menghapus kata tersebut dari *list* kata sehingga didapatkan *list* kata yang sudah di *decrease* yaitu

**WOOD**  
**SPOT**  
**FLAT**  
**DRAG**  
**WEAK**

Gambar 3.2 List Kandidat Password Setelah di Decrease

Kemudian kita akan mengulangi dua langkah sebelumnya. Misal kita akan memilih WOOD sebagai pilihan kita maka kita akan mendapatkan *likeliness* WOOD dengan AGES yaitu 0 yang berarti tidak ada kemiripan kata sama sekali. Maka kita akan menghapus WOOD kata beserta semua kata yang memiliki *likeliness* > 0 dari *list* kandidat password.

Langkah selanjutnya kita mencari kata-kata yang memiliki *likeliness* > 0 dengan menghitung *likeliness* dari kata WOOD dengan kata lainnya.

Kata pertama	Kata kedua	<i>Likeliness</i>
WOOD	SPOT	1
WOOD	FLAT	0
WOOD	DRAG	0
WOOD	WEAK	1

Tabel 3.2 *Likeliness* kata WOOD dengan kata lainnya

Dari tabel 3.2 kita mendapatkan kata yang memiliki kemiripan minimal 1 dengan WOOD yaitu SPOT dan WEAK. Maka kita akan menghapus kata tersebut dari *list* kata sehingga didapatkan *list* kata yang sudah di *decrease* yaitu

**FLAT**  
**DRAG**

Gambar 3.3 *List Kandidat Password Setelah di Decrease*

Kita mendapatkan 2 kandidat kata yaitu FLAT dan DRAG. Maka kita akan kembali menebak kata karena kata yang sesuai dengan *password* yang sebenarnya belum ditemukan. Misal kata yang kita pilih adalah FLAT maka kita akan mendapatkan *likeliness* FLAT dengan DRAG yaitu 1. Maka kita akan menghapus FLAT dari daftar kumpulan kandidat password sehingga daftar kumpulan kata yang kita punya tinggal bersisa 1 kata yaitu DRAG. Maka kita simpulkan DRAG merupakan *password* sebenarnya yang kita cari.

Kita telah melakukan pencarian *password* dengan menggunakan cara berpikir algoritma *Decrease and Conquer*. Pertama kita menebak kata AGES untuk pengujian kata pertama dan untuk membuang kandidat kata yang tidak sesuai dengan *password* sebenarnya. Kemudian kita melakukan hal yang sama dengan kata WOOD dan FLAT. Maka jika ditotal kita menebak sebanyak 4 kali termasuk tebakan terakhir yaitu DRAG yang merupakan *password* yang sebenarnya.

#### IV. KESIMPULAN

*Hacking Minigame* pada permainan *Fallout 4* dapat dipermudah penyelesaiannya dengan menggunakan algoritma

*Decrease and Conquer*. Algoritma ini dapat mempermudah player dalam mendecrase kandidat *password* yang mungkin sehingga proses penebakan *password* menjadi lebih mudah dan memakan langkah yang lebih sedikit dibanding jika *player* harus menebak secara manual. Menebak secara manual kemungkinan besar dapat menyebabkan *player* kehabisan langkah dan *player* jadi merasa tidak mendapatkan kesenangan akibat kekalahan di dalam sebuah permainan. Setelah pengujian yang dilakukan *player* berkemungkinan besar menyelesaikan persoalan ini dalam kurang dari 4 langkah sehingga dapat memenangkan permainan.

Penulis disini tidak menekankan pada implementasi berupa kode program yang ditulis dengan bahasa pemrograman tertentu. Akan tetapi disini penulis lebih menekankan kepada cara berpikir algoritma *Decrease and Conquer* untuk menyelesaikan permainan *hacking minigame* karena dalam permainan kita tidak akan mengetahui *password* yang sebenarnya dan tidak dapat kita simulasikan dalam program.

#### VIDEO LINK AT YOUTUBE

Video penjelasan makalah di *upload* di channel youtube penulis yaitu

[https://www.youtube.com/channel/UCGhbRcr\\_QzZQJ8ksU9qohPw](https://www.youtube.com/channel/UCGhbRcr_QzZQJ8ksU9qohPw)

#### UCAPAN TERIMA KASIH

Puji dan syukur penulis panjatkan kepada Allah SWT karena berkat rahmat dan hidayahnya penulis dapat menyelesaikan makalah ini. Penulis juga tidak lupa berterima kasih kepada semua pihak yang membantu membuat referensi agar penulis dapat lebih mudah menyelesaikan makalah ini. Karena tugas-tugas yang diberikan selama semester ini, penulis mendapatkan manfaat algoritma yang dipelajari selama ini tidak hanya dalam tugas kuliah, tetapi juga dalam permainan yang sering dimainkan dalam kehidupan sehari-hari.

Penulis berharap penjelasan mengenai cara berpikir algoritma *Decrease and Conquer* dalam permainan *Fallout 4* dapat membantu pemain-pemain yang ingin menamatkan game tersebut agar lebih mengerti cara mengeliminasi kata-kata yang tidak perlu ditebak.

Penulis mengucapkan terima kasih kepada Bapak Rinaldi Munir selaku dosen mata kuliah Strategi Algoritma IF2211 kelas 04 dan juga kepada seluruh tim pengajar IF2211 yang telah menemani penulis selama satu semester di Teknik Informatika ITB.

#### REFERENSI

- [1] <https://informatika.stei.itb.ac.id/~rinaldi.munir/Smik/2020-2021/Algoritma-Decrease-and-Conquer-2021-Bagian1.pdf>
- [2] <https://informatika.stei.itb.ac.id/~rinaldi.munir/Smik/2020-2021/Algoritma-Decrease-and-Conquer-2021-Bagian2.pdf>
- [3] <https://fallout.fandom.com/wiki/Hacking>
- [4] <https://eip.gg/fallout4/guides/hacking/>

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 26 April 2021

A handwritten signature in black ink, consisting of a large, stylized initial 'R' followed by a series of connected loops and a long horizontal stroke at the end.

Ryan Kurnia Hidayatullah 13519212